

利用网格化处理建模并基于遗传算法优化定日镜场模型

摘要

在优化提升定日镜场整体对太阳光能的利用效率的过程中，定日镜场的布置方式是其中重要的影响因素之一。定日镜场的布置方式涉及到很多参数的设置，在本文中，我们探究了定日镜的排布方式（包括定日镜之间的间隔、定日镜的排布数量和坐标、定日镜的安装高度）、定日镜的尺寸、吸收塔的位置坐标等相关因素对太阳光能的利用效率，以此进行优化布置，使得单位镜面面积年平均输出热功率尽量大，为实际测试和生产提供理论参考。本文结合实际情况对于定日镜的排布中的相关要素进行合理的假设，建立三维空间坐标系。在满足约束条件下，在问题一中利用网格处理建模，在问题二和三中利用遗传算法进行优化，计算有效面积，同时对于定日镜的排布方式做出合理假设，计算求解使得单位镜面面积年平均输出热功率尽量大。

针对问题一 建立三维坐标系，计算出太阳光入射的方向向量，利用光的反射定律，可求解出定日镜法向量和平面方程。利用题干约束条件，将其转化为数学形式，列出镜面方程。然后对定日镜镜面网格化处理，分为 $n * n$ 奇数个网格，用每个网格的中心代替这个网格进行处理。下面计算有效面积，分为三部分计算：1、吸收塔的阴影遮挡 2、相邻定日镜对反射光线的遮挡 3、反射光线是否可以打在集热器上。第一部分，将吸收塔和集热器抽象成圆柱体模型，考虑它们在太阳光下的三维立体阴影部分，将阴影部分划分为一个三棱柱和一个倾斜的圆柱，两部分会有重叠，判断光线是否会落在这两块阴影部分；第二部分，首先筛选可能对选定为研究对象的定日镜产生遮挡的定日镜，然后利用网格化处理的方法，判断每个网格的反射光线和其他镜面的约束方程联立之后是否有解，以此判断反射光线是否会被遮挡。第三部分，根据光线的位置关系建立数学方程进行求解，如果结果解落在集热器的高度区间内，说明可以打到集热器上。

针对问题二 在查阅了大量的文献资料并考虑现实条件之后，我们确定了两种较为合理的定日镜排放方案。方案一：均匀圆形排列，径向交错布置；方案二：由内向外呈等差数列排布，内疏外密。其中方案一可提高对于太阳光的利用率；经计算方案二对光的利用率会略低。所以本文中采用方案一完成对于定日镜场的优化布置，并对于方案一的求解和计算进行详细的解释推导。定日镜场优化时，可将前提约束条件转化为数学形式，然后列出定日镜在三维坐标下的坐标表示。利用附录公式计算推导单位镜面面积年平均输出热功率，并利用遗传算法进行优化。

针对问题三 由问题二的计算结果可看出，在其他条件相同时，问题二中方案一定日镜均匀排布、径向交错的方法求解出来的单位镜面面积年平均输出热功率较大。所以在本题中，我们建立模型时，选择方案一中定日镜场的排布方式，结合实际情况和计算的需求做出合理的假设，规定每 5 圈定日镜的尺寸、高度都一致，共设 55 圈，每圈镜子的数量为 50。其他模型的细节和问题二一致。

关键字： 网格处理 径向交错的辐射网格状圆形布局 遗传算法

一、问题重述

1.1 问题背景

近现代以来,世界科技高速发展、工业化持续推进、人口大幅增长,人类对于能源的需求和消耗越来越大。伴随着化石能源日益枯竭,对于可再生资源的研究和应用成文人类社会重要议题。太阳能是可再生能源的重要类型之一,随着技术的发展,太阳能发电的成本已大大减少许多。太阳能热发电技术是大规模开发利用太阳能的一个重要技术途径。太阳能热发电系统按照聚光形式的不同,主要分为槽式、塔式、线性菲涅尔式和碟式 4 种太阳能热发电系统 [2]。其中,塔式太阳能热发电系统具有高聚光比、高太阳能转化率以及更大的发电效率提升空间等优点,因此,该发电系统成为最具有开发潜力的太阳能热发电系统,也是大规模太阳能热发电系统的最理想方式。塔式太阳能光热发电站是通过大量的定日镜聚集太阳光能,使低密度的太阳光能量反射到位于高塔顶部的吸热器上聚集成为高盛度的太阳光能量,并由吸热器将光能转换成工质热能,再经汽轮发电机组将热能转换成电能的过程。太阳能转换为热能,其原理实际上是反射镜成像的过程(太阳为光源、定日镜为反射镜、吸热器是光斑的成像点)[1]。

定日镜场是塔式光热电站的重要且关键的组成部分之一,定日镜场在整个系统工作的环节中实现能量的输入,并且其综合效率直接决定了发电系统的最高性能。整个镜场的建设成本占据电站总投资的一半,而大约有 3 到 4 成的能量损失发生在定日镜场,因此塔式太阳能光热发电站在建设前期需要解决的关键问题在于如何实现定日镜场的优化布置。

一个镜场通常需要通过大量定日镜才能实现能量聚集,在镜场设计过程中,最重要的是需要考虑如何提高镜场整体对太阳光能的利用效率即镜场光学效率,镜场光学效率是指吸热器接收到的光通量与镜场反射镜能接收的最大的光通量之比 [1]。镜场设计原则是:以最小的反射镜接收面积,为吸热器提供最大的太阳光能量,与此同时还需保证吸热器上能流密度的分布满足吸热器要求。镜场的布置方式是决定镜场光学效率的主要因素。镜场布置时,太阳所处的位置、定日镜之间的距离,镜场的排布方式,定日镜与吸热器之间的距离,吸热器布置高度,都直接影响着镜场的光学效率。由于太阳在全年每一天的不同时刻,都处于天空中的不同位置,即光源相对于镜场中各定日镜的位置是以年为周期随时间不断变化的,但各定日镜反射光线的目标点衡量一个镜场布置是否合理,最主要的指标就是镜场光学效率的高低,因此对于镜场光学效率的计算方法研究是光热电站一项重点内容 [1]。

1.2 题目重述

由于定日镜场的综合效率会直接影响甚至决定发电系统的最高性能，所以需要考虑优化提升定日镜场整体对太阳光能的利用效率。在提升整体的综合效率之中，最为重要的部分是定日镜场的布置方式。同时定日镜场的设计、布置方式会收到多方面因素的影响。如太阳所处位置、定日镜场中定日镜间的距离、定日镜和吸收塔集热器的距离、定日镜的排布方位、定日镜的尺寸、吸收塔和集热器的高度等。所以，在题目一中需要先对指定镜场条件下，计算出一些有关镜场整体的综合效率和参数。然后在问题二、三中，需要在能够达到设定额定年平均输出热功率的条件下，根据不同的限制条件，建立合理的模型和假设，求解参数、进行优化设计。

(1) 问题一

给定定日镜场的排布方式、定日镜场的参数、公式和条件，计算反映定日镜场综合效率的物理量。利用附录提供的相关公式计算出相关数据，构建模型，结合实际情况和计算的需求做出合理的假设，根据物理关系和数学方程等将其转化为可求解的公式，计算出结果。

(2) 问题二

给定额定年平均输出热功率，调节参数值并在满足所有定日镜尺寸及安装高度相同的条件，使得单位镜面面积年平均输出热功率尽量大，利用已知的约束条件，将其转化为数学形式进行表述，对定日镜场的布局进行合理的假设，结合计算的需求建立合适模型并做出假设条件，计算出两种方案的结果最后进行比较选择最有效的方案。

(3) 问题三

给定额定年平均输出热功率，要在能够达到额定功率的情况下，调节参数值，实现单位镜面面积年平均输出热功率最大。建立模型与合适的假设，列出数学关系进行求解，和问题二过程类似。

二、问题分析

2.1 问题一分析

(1) 求解定日镜镜面约束方程：已知定日镜的尺寸和具体位置，建立空间三维坐标系，利用附录中提供的公式计算出表格 1 中所要求的每月 21 日 5 个计算时点的太阳赤纬角 δ 、太阳高度角 α_s 、太阳方位角 γ_s ，通过这些参数的求解，求得太阳光入射的方向向量。利用已知的集热器中心和定日镜中心坐标，以及求解出来的太阳光入射的方向向量经过定日镜中心的条件，可以确定入射光线和反射光线，利用光的反射定律，可以求解出定日镜平面方程。

(2) 计算有效面积：分别考虑入射光线和反射光线。入射光线的影响主要来自吸收塔的阴影，反射光线的影响主要来自相邻定日镜的遮挡。最后，还要考虑反射光线是否

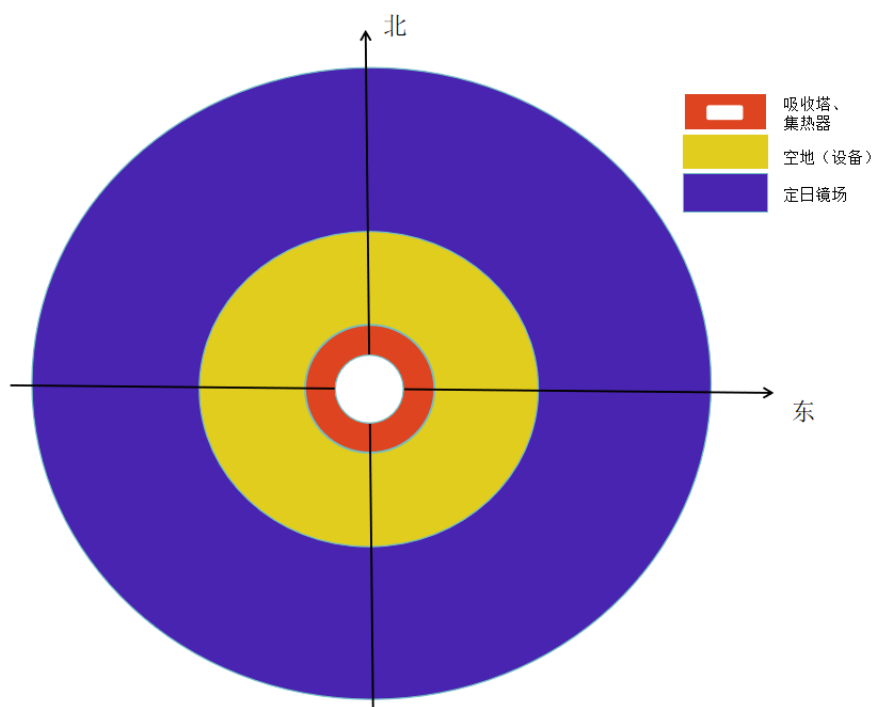


图 1 初始镜场区域分布

可以打在集热器上。

（3）关键方法：对定日镜进行网格化处理，其关键思想是微分法，当网格划分的数量足够大时，可以用每个网格的中心替代这个网格进行处理。

2.2 问题二分析

（1）确定模型、设定参数：首先确定三维坐标系，和问题一建立的三维坐标系类似，这一题将阴影遮挡效率和截断效率设为常数进行计算。设定定日镜的参数，包括镜面高度、镜面宽度、安装高度、排放方式、数量等。

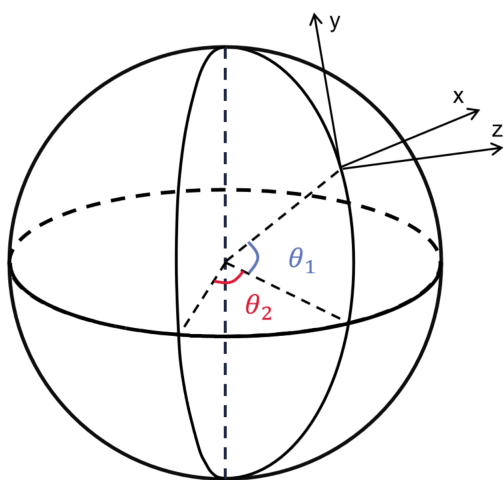
（2）约束条件的转化：将题目所设定的硬性条件转化为数学表达形式，同时将有关定日镜的相关参数的关系转化为数学表达形式。最后基于遗传算法优化求解。最终求解出满足题意的结果

2.3 问题三分析

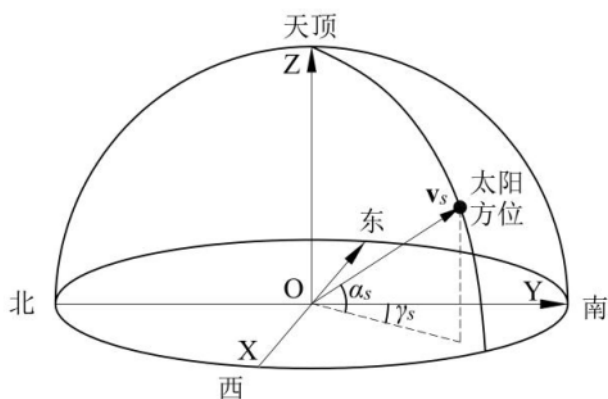
在问题二的基础上进行优化，为简化计算，建立模型时，选择方案一中定日镜场的排布方式，结合实际情况和计算的需求做出合理的假设，规定每 5 圈定日镜的尺寸、高度都一致，共设 55 圈，每圈镜子的数量为 50。同样基于遗传算法优化求解。其他模型的细节和问题二一致。

三、模型假设

3.1 坐标系



(a) 时角坐标系



(b) 地平坐标系

* 时角坐标系（赤道坐标系）中 θ_1 是纬度， θ_2 是经度。

* 地平坐标系中 α_s 是太阳高度角， γ_s 是太阳方位角。

3.2 建立模型及模型假设

- (1) 建立方式和时角坐标系保持一致，吸收塔中心的投影为 xy 平面的原点；
- (2) 太阳光是平行光束（太阳光是具有一定锥形角的一束锥形光线，但是角度非常小，此处假设为平行光）；
- (3) 假设春分日期为 3 月 21 日；
- (4) 假设镜面中心与镜面支撑点重合。
- (5) 为简化计算，阴影遮挡效率和截断效率均设为常数，分别为 0.95 和 1。（在问题一中提出了有效计算阴影遮挡效率和截断效率的方法）

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	°
φ	当地纬度，北正	°
ω	太阳时角	°
ST	当地时间	h
δ	太阳赤纬角	°
D	以春分作为第 0 天起算的天数	-
DNI	法向直接辐射辐照度	kW/m^2
G_0	太阳常数	取 $1.366\text{kW}/\text{m}^2$
H	海拔高度	km
E_{field}	定日镜场的输出热功率	kW
N	定日镜总数	-
A_i	第 i 面定日镜采光面积	m^2
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	取常数 0.92
d_{HR}	镜面中心到集热器中心的距离	m
n	平面镜划分为网格的行数和列数	-
(x_0, y_0, z_0)	定日镜中心坐标	m
$\vec{n} = (a, b, c)$	定日镜法向量	-
(x_1, y_1, z_1)	选定并研究的某网格中心坐标	m
$\overline{\eta_{\text{cos}}^k}$	第 k 圈的余弦效率	-
(x_i, y_i, z_i)	其他定日镜的中心坐标	m

五、问题一模型

5.1 模型的建立

5.1.1 假设条件

基于对计算速度和准确度的考虑，为在一定程度上提高计算速度并获得必要的计算精度，在进行效率计算之前先做以下假设：

1、计算定日镜的阴影遮挡时，将定日镜网格化处理，划分为由 $n * n$ 个网格，其中 n 取为奇数，（此时可以保证一定可以找到中心的一块），用每个网格的中心代替这个网格进行光线处理。

2、不考虑定日镜的影子对结果的影响。

3、假设阴影和遮挡只发生在目标定日镜周围的一定范围之内。

5.1.2 建立模型

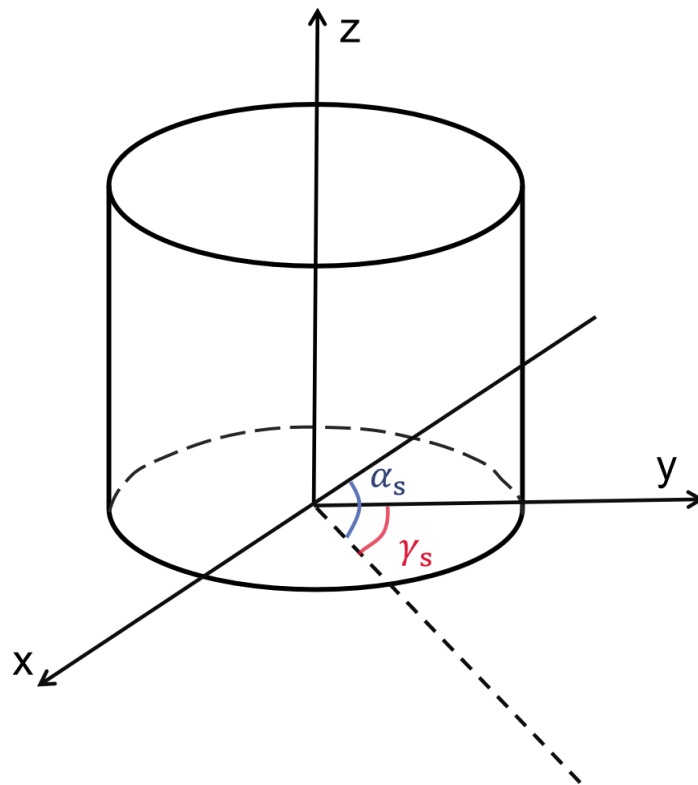


图3 三维坐标系的建立

建立空间三维坐标系，建立方式和时角坐标系保持一致，吸收塔中心的投影为 xy 平面的原点，见图3 从镜场中依次选择一面定日镜作为待判定的目标定日镜，判断是否会受吸收塔的阴影的影响，判断是否会受到其他定日镜的遮挡影响，

5.2 模型的求解

5.2.1 镜面方程

建立三维坐标系，首先计算镜面的法向量，在确定时间天数 D 和当地时间 ST 的条件下，计算出太阳赤纬角 δ ，利用公式 $\sin \alpha_s = \cos \delta \cos \varphi \cos \omega + \sin \delta \sin \varphi$ 计算出太阳高度角 α_s ，利用公式 $\cos \gamma_s = \frac{\sin \delta - \sin \alpha_s \sin \varphi}{\cos \alpha_s \cos \varphi}$ 计算出太阳方位角 γ_s ，此公式最后形式为 $\cos \gamma_s = C$ （ C 是确定的常数），若 $\arccos C$ 在 $[0, \pi]$ 区间有解 C_1 ，则可以解得

$$\gamma_s = \begin{cases} C_1 & , ST < 12 \\ 2\pi - C_1, & ST \geq 12 \end{cases}$$

最终解得太阳光线的方向即光入射的方向向量

$$\vec{n}_{\text{太阳}} = (-\cos \alpha_s \sin \gamma_s, -\cos \alpha_s \cos \gamma_s, -\sin \alpha_s)$$

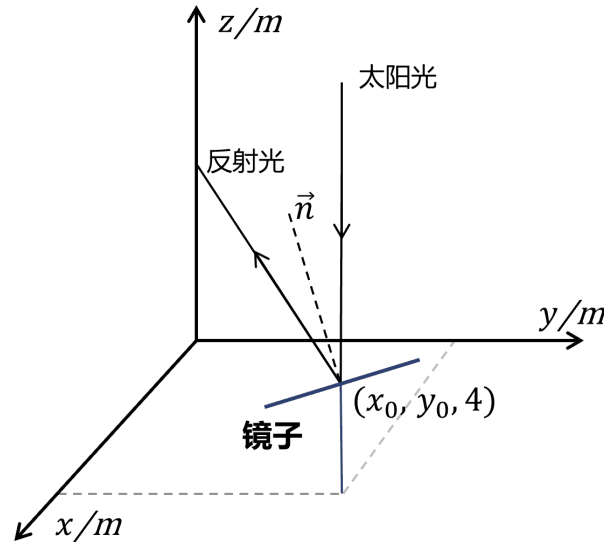


图 4 反射光线示意图

由附件中给出的定日镜的位置可以确定定日镜中心的坐标 $(x_0, y_0, 4)$ ，假设太阳光是平行光，同时题干要求：定日镜在工作时，控制系统根据太阳的位置实时控制定日镜的法向，使得太阳中心点发出的光线经定日镜中心反射后指向集热器中心。集热器中心坐标已知，在本题建立的坐标系下为 $(0, 0, 80)$ ，由集热器中心坐标和定日镜中心的坐标可以确定反射光线，同时已经确定方向的太阳光线经过定日镜中心，由此可以确定入射光线。由光的反射定律可以计算出定日镜的法向量 \vec{n} ，设定日镜中心坐标为 (x_0, y_0, z_0) （已知），设定日镜法向量为 $\vec{n} = (a, b, c)$ （已经求解）。则定日镜所在平面的方程为

$$a(x - x_0) + b(y - y_0) + c(z - z_0) = 0 \quad (1)$$

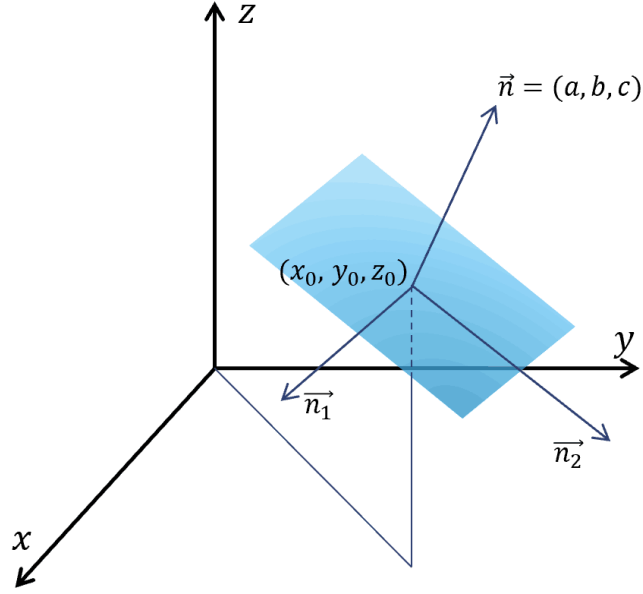


图 5 定日镜镜面方程示意图

以镜面中心为原点选取定日镜面上两条互相垂直的向量 \vec{n}_1 和 \vec{n}_2 ，由题干可知，定日镜底座由纵向转轴和水平转轴组成，平面反射镜安装在水平转轴上。纵向转轴的轴线与地面垂直，可以控制反射镜的方位角。所以可以选取到镜面上与 Z 轴平行的直线 l ，可列得关于直线 l 的方程为

$$\begin{cases} a(x - x_0) + b(y - y_0) = 0 \\ z = z_0 \end{cases}$$

则直线 l 的方向向量 \vec{n}_1 为 $(-b, a, 0)$ ，利用 $\vec{n} \perp \vec{n}_2$ 和 $\vec{n}_1 \perp \vec{n}_2$ 可以解得 $\vec{n}_2 = (-ac, -bc, a^2 + b^2)$

下面考虑约束条件，题目中规定定日镜尺寸为 $6m \times 6m$ ，由此可列出镜面方程（约束方程）

$$\begin{cases} -3 \leq ((x, y, z) - (x_0, y_0, z_0)) \cdot \frac{\vec{n}_1}{|\vec{n}_1|} \leq 3 \\ -3 \leq ((x, y, z) - (x_0, y_0, z_0)) \cdot \frac{\vec{n}_2}{|\vec{n}_2|} \leq 3 \\ a(x - x_0) + b(y - y_0) + c(z - z_0) = 0 \end{cases}$$

将上面所求值带入此镜面方程可得

$$\begin{cases} a(x - x_0) + b(y - y_0) + c(z - z_0) = 0 \\ -3 \leq [-b(x - x_0) + a(y - y_0)] \cdot \frac{1}{\sqrt{a^2 + b^2}} \leq 3 \\ -3 \leq [-ac(x - x_0) - bc(y - y_0) + (a^2 + b^2)(z - z_0)] \cdot \frac{1}{\sqrt{(a^2 + b^2)(a^2 + b^2 + c^2)}} \leq 3 \end{cases} \quad (2)$$

5.2.2 网格化处理-微分法处理

将定日镜 $n * n$ 等分为奇数块，其中 n 取为奇数，用每个网格的中心代替这个网格进行光线处理。检查反射光线是否会被遮挡时，可以转化成对每个网格的中心的反射光线进行判断。定日镜网格化处理之后分成等大小的奇数块，其中第 $\frac{n+1}{2}$ 行 $\frac{n+1}{2}$ 列的网格的中心坐标为 (x_0, y_0, z_0) ，则坐标为 (i, j) 的网格的中心点在镜场坐标系下的坐标为

$$(x_0, y_0, z_0) + \frac{\vec{n}_1}{|\vec{n}_1|} \cdot \left(i - \frac{n+1}{2}\right) \cdot \frac{6}{n} + \frac{\vec{n}_2}{|\vec{n}_2|} \cdot \left(j - \frac{n+1}{2}\right) \cdot \frac{6}{n} \quad (3)$$

即

$$\begin{cases} x_i = x_0 + \frac{6}{n} \cdot \left[\frac{(-ac) \cdot (j - \frac{n+1}{2})}{\sqrt{(a^2+b^2)(a^2+b^2+c^2)}} + \frac{-b \cdot (i - \frac{n+1}{2})}{\sqrt{a^2+b^2}} \right], \\ y_i = y_0 + \frac{6}{n} \cdot \left[\frac{(-bc) \cdot (j - \frac{n+1}{2})}{\sqrt{(a^2+b^2)(a^2+b^2+c^2)}} + \frac{a \cdot (i - \frac{n+1}{2})}{\sqrt{a^2+b^2}} \right], \\ z_i = z_0 + \frac{6}{n} \cdot \frac{(a^2+b^2) \cdot (j - \frac{n+1}{2})}{\sqrt{(a^2+b^2)(a^2+b^2+c^2)}} \end{cases} \quad (4)$$

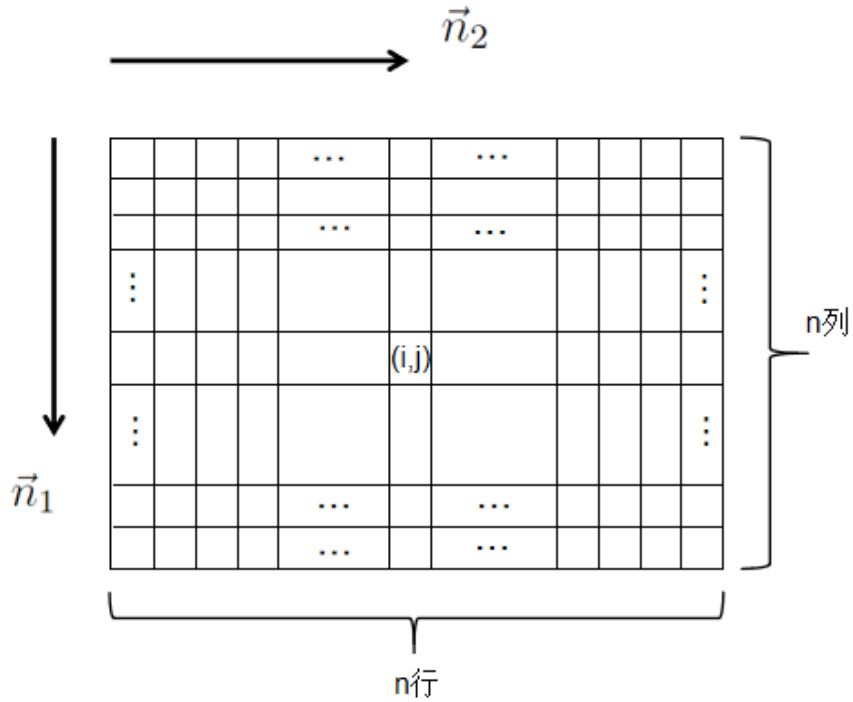
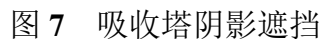


图 6 网格化微分处理示意图

5.2.3 有效面积

(1) 吸收塔的阴影遮挡


$$x^2 + y^2 = \frac{49}{4} \quad , 0 \leq z \leq 84$$

由前文可知，太阳光光入射的方向向量为

则当光落在第一部分时, 根据建立的数学模型, 需要限制的条件可以表示为以下数学形式:

$$\begin{cases} -3.5 \leq x \cdot \cos \gamma_s - y \cdot \sin \gamma_s \leq 3.5 \\ 0 \leq -x \cdot \sin \gamma_s - y \cdot \cos \gamma_s \leq \frac{84}{\tan \alpha_s} \\ x \cdot \sin \gamma_s + y \cdot \cos \gamma_s - (z - 84) \cdot \frac{1}{\tan \alpha_s} \geq 0 \end{cases} \quad (5)$$

$$\left(x - \frac{\sin \gamma_s}{\tan \alpha_s} \cdot (z - 84)\right)^2 + \left(y - \frac{\cos \gamma_s}{\tan \alpha_s} \cdot (z - 84)\right)^2 \leq \frac{49}{4} \quad (6)$$

11

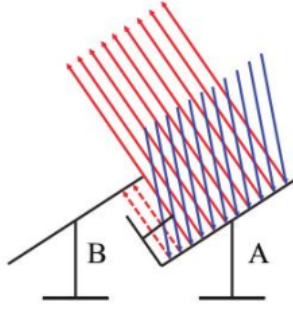


图 8 遮挡

为了分析定日镜彼此之间的影响情况，我们可以先选定其中一个定日镜，分析这个定日镜的反射光线是否会被其他定日镜遮挡。首先，需要对定日镜进行筛选，只有其他定日镜和高塔距离 d 满足

$$d \leq \sqrt{x_0^2 + y_0^2}$$

才有可能挡住选定的定日镜的反射光线。下面分析筛选出满足上述条件的定日镜是否会遮挡选定的定日镜的反射光线，该定日镜的中心坐标为 (x_0, y_0, z_0) ，选择其上某个网格的中心，坐标为 (x_1, y_1, z_1) ，则经过该网格中心的反射光线方程为

$$\frac{x - x_1}{-\cos \alpha_s \sin \gamma_s} = \frac{y - y_1}{-\cos \alpha_s \cos \gamma_s} = \frac{z - z_1}{-\sin \alpha_s} \quad (7)$$

将该式和其他镜面约束方程联立，如果存在解，则说明此时被遮挡。

(3) 分析其反射光线是否可以打在集热器上

求解方程

$$\begin{cases} x - x_1 = \frac{\sin \gamma_s}{\tan \alpha_s} (z - z_1) \\ y - y_1 = \frac{\cos \gamma_s}{\tan \alpha_s} (z - z_1) \end{cases} \quad (8)$$

并代入 $x^2 + y^2 = \frac{49}{4}$ 求解，如果存在 $z \in [76, 84]$ 则说明可以打在集热器上。至此，有效面积的影响因素已分析完毕。下给出阴影遮挡效率和集热器截断效率计算公式，其中面积均用网格个数等效替代。

5.2.4 算法描述

将定日镜 $n * n$ 等分为奇数块，其中 n 取为奇数，用每个网格的中心代替这个网格进行光线处理。检查反射光线是否会被遮挡时，可以转化成对每个网格的中心的反射光线进行判断。

$$\eta_{sb} = \frac{\text{总面积} - \text{阴影面积}}{\text{总面积}}$$

$$\eta_{trunc} = \frac{\text{有效面积}}{\text{总面积} - \text{阴影面积}}$$

5.3 求解结果

按题要求，将计算的相关数据填入表中

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	0.5776	0.6505	0.95	1	0.6179
2 月 21 日	0.5529	0.6227	0.95	1	0.5915
3 月 21 日	0.5130	0.5778	0.95	1	0.5489
4 月 21 日	0.5381	0.6060	0.95	1	0.5757
5 月 21 日	0.6280	0.7072	0.95	1	0.6719
6 月 21 日	0.6570	0.7399	0.95	1	0.7029
7 月 21 日	0.6270	0.7061	0.95	1	0.6708
8 月 21 日	0.5321	0.5992	0.95	1	0.5693
9 月 21 日	0.5119	0.5765	0.95	1	0.5477
10 月 21 日	0.5578	0.6282	0.95	1	0.5968
11 月 21 日	0.5783	0.6513	0.95	1	0.6187
12 月 21 日	0.5793	0.6525	0.95	1	0.6198

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 (kW/m ²)
0.5711	0.6432	0.95	1	38.3827	0.6110

* 部分数据四舍五入保留 4 位小数

六、问题二模型

6.1 模型的建立

(1) 在问题一的基础上，我们将阴影遮挡效率和截断效率设为常数进行计算求解，并用问题一中的平均 DNI 代替 DNI 进行计算。和问题一建立的三维坐标系类似。

(2) 将吸收塔的位置简化设置，设其坐标 (x_0, y_0, z_0) 为 $(0, 0, z_0)$ ，其中 z_0 是 $80m$ 。

(3) 在题目二中，限制所有的定日镜尺寸及安装高度相同，我们设镜面高度为 m_a ，镜面宽度为 m_b ，安装高度为 m_h 。

(4) 查阅大量资料和文献之后，结合实际情况综合考虑，最终关于定日镜的排放方式提出两种方案：

方案一：定日镜场采用径向交错的辐射网格状圆形布局方式 [6]。共设 55 圈，每一圈的定日镜数量 N 相同，均为 50 面。

方案二：设定日镜的排放方式是圆形排列，每一圈以吸收塔中心在 xy 平面的投影点位圆心，从 X 轴上选定的位置开始等距排放。由内向外各圈的定日镜数量为公差为 1、递增的等差数列，最内圈数量为 30，共设 50 圈，共有 2725 面定日镜。

选定两种方案的原因：

方案一均匀圆形排列，径向交错布置，可以提高对于太阳光的利用率；

方案二由内向外呈等差数列排布，内疏外密，这种情况更容易受到吸收塔的阴影和其他因素的遮挡影响，对光的利用率会略低。

6.2 模型的求解

6.2.1 约束条件

根据建立的数学模型，定日镜的坐标可以表示为 (x_i, y_i, m_h) ， $1 \leq i \leq 50$ 。设计定日镜场时，安装高度要在 $2m$ 至 $6m$ 之间，安装高度必须保证镜面在绕水平转轴旋转时不会触及以上，以上相关的前提约束条件可以转化为以下数学形式：

$$\begin{cases} 2 \leq m_h \leq 6 \\ 2 \leq m_a \leq 2 \cdot m_h \\ m_a \leq m_b \leq 8 \\ 25 < (x_i - x_j)^2 + (y_i - y_j)^2, \quad 1 \leq i, j \leq 50 \end{cases} \quad (9)$$

由建立的数学模型可知，吸收塔的坐标中的 (x_0, y_0, z_0) 为 $(0, 0, 80m)$ 。均匀圆形排列排放的每一圈里都有 50 面定日镜，所以可以写出定日镜在三维坐标系下的坐标表示：

$$\begin{cases} (r_k \cdot \cos \frac{m \cdot 2\pi}{N}, r_k \cdot \sin \frac{m \cdot 2\pi}{N}, m_h) & k \text{ 是奇数 } k = 1, 3, \dots, 55, \\ (r_k \cdot \cos \frac{m \cdot 2\pi + \pi}{N}, r_k \cdot \sin \frac{m \cdot 2\pi + \pi}{N}, m_h) & k \text{ 是偶数 } k = 2, 4, \dots, 54, \end{cases} \quad (10)$$

其中需要保证满足前提条件 $r_k \cdot \sin \frac{2\pi}{N} \geq 5, r_k > 100, m = 1, \dots, 50, N = 50$

对应约束条件的形式转换为：

$$\begin{cases} 2 \leq m_h \leq 6 \\ 2 \leq m_a \leq m_b \leq 8 \\ \frac{m_a}{2} \leq m_h \\ r_k \cdot \sin \frac{2\pi}{N} \geq 5 \\ r_k > 100, k = 1, \dots, 55 \\ r_1 < r_2 < \dots < r_{55} \end{cases} \quad (11)$$

因为 $r_1 < r_2 < \dots < r_{55}$ ，所以只需要 r_1 满足对应的约束条件即可。因此上式可以化简为：

$$\begin{cases} 2 \leq m_h \leq 6 \\ 2 \leq m_a \leq m_b \leq 8 \\ \frac{m_a}{2} \leq m_h \\ r_1 \cdot \sin \frac{2\pi}{N} \geq 5 \\ r_1 > 100 \\ E = DNI \cdot \sum_{i=1}^N m_a \cdot m_b \cdot \bar{\eta}_i = 60\text{MW} \end{cases} \quad (12)$$

6.2.2 计算求解

定日镜场的额定年平均输出热功率为 60MW，根据附录中提供的计算定日镜场的输出热功率 E_{field} 公式，当定日镜场达到额定功率 E 时， $E = E_{\text{field}}$ ，展开形式为：

$$E = DNI \cdot \sum_{i=1}^N m_a \cdot m_b \cdot \bar{\eta}_i = 60\text{MW} \quad (13)$$

其中 $\bar{\eta}_i$ 是 60 个时间点对应的 η_i 的平均值。根据附录提供的公式，在 $\eta = \eta_{\text{sb}} \eta_{\text{cos}} \eta_{\text{at}} \eta_{\text{trunc}} \eta_{\text{ref}}$ 式子中， η_{sb} 、 η_{trunc} 和 η_{ref} 均是常数。

第一步，需要先进行 η_{cos} 的求解：

计算对于确定位置坐标的定日镜的 η_{cos} ，根据 η_{cos} 的定义，存在如下等式关系

$$\eta_{\text{cos}} = \cos \theta = |\vec{n}_i \cdot \vec{n}_{\text{太阳}}|$$

其中

$$\vec{n}_i = \frac{\vec{n}_{\text{太阳}} + \frac{(x_i, y_i, m_h) - (x_0, y_0, z_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (m_h - z_0)^2}}}{2}$$

带入数据即

$$\vec{n}_i = \frac{\vec{n}_{\text{太阳}} + \frac{(x_i, y_i, m_h - 80)}{\sqrt{x_i^2 + y_i^2 + (m_h - 80)^2}}}{2}$$

下面计算第 k 圈的 $\bar{\eta}$:

此时公式 $\bar{\eta} = \eta_{sb} \bar{\eta}_{\cos} \eta_{at} \eta_{trunc} \eta_{ref}$ 式子中, η_{sb} 、 η_{trunc} 和 η_{ref} 均是常数。

因此只需要考虑 $\bar{\eta}_{\cos}$ 和 η_{at} 的计算。

(1) 计算 η_{at}

根据公式可知, 其中 d_{HR} 为镜面中心到集热器中心的距离, 我们考虑在三维坐标系下 d_{HR} 的值, 表达式为 $d_{HR} = \sqrt{(r_k)^2 + (m_h - 80)^2}$

(2) 计算 η_{\cos} :

计算第 k 圈的 η_{\cos}^k , 转化为数学形式为:

$$\begin{aligned} \bar{\eta}_{\cos}^k &= \frac{1}{60} \sum_{i=1}^{60} \vec{n}_{\text{太阳}i} (\vec{n}_{k1} + \cdots + \vec{n}_{k20}) \\ &= \frac{1}{60} \sum_{i=1}^{60} \vec{n}_{\text{太阳}i} \cdot \frac{1}{2} \cdot \left[(\vec{n}_{\text{太阳}1} + \cdots + \vec{n}_{\text{太阳}50}) + 50 \cdot \frac{(0, 0, m_h - 80)}{\sqrt{r_k^2 + (m_h - 80)^2}} \right] \\ &= \frac{1}{60} \sum_{i=1}^{60} \left[25 + 25 \cdot \vec{n}_{\text{太阳}i} \cdot \frac{(0, 0, m_h - 80)}{\sqrt{r_k^2 + (m_h - 80)^2}} \right] \end{aligned} \quad (14)$$

第 k 圈上单个定日镜的余弦效率, 需要将 $\bar{\eta}_{\cos}^k$ 除以每一圈的定日镜的数目 50。

若需要求解题目二, 则需要使 $\frac{60MW}{N \cdot m_a \cdot m_b}$ 最小。

6.3 求解结果

按题要求, 将计算的相关数据填入表中

注:

注: 为计算便利, 我们设定额定功率的参数为 $59MW$ 到 $61MW$ 在计算单位面积镜面平均输出热功率计算时, 采用近似计算公式

$$\frac{\text{单位面积镜面平均输出热功率}}{\text{年平均输出热功率}} \cdot 60MW$$

* 部分数据四舍五入保留 4 位小数

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功率 (kW/m ²)
1 月 21 日	0.4856	0.5742	0.95	1	0.4545
2 月 21 日	0.5045	0.5965	0.95	1	0.4915
3 月 21 日	0.5239	0.6194	0.95	1	0.5183
4 月 21 日	0.5414	0.6400	0.95	1	0.5361
5 月 21 日	0.5511	0.6515	0.95	1	0.5443
6 月 21 日	0.5541	0.6550	0.95	1	0.5466
7 月 21 日	0.5510	0.6514	0.95	1	0.5442
8 月 21 日	0.5407	0.6392	0.95	1	0.5355
9 月 21 日	0.5230	0.6183	0.95	1	0.5171
10 月 21 日	0.5021	0.5936	0.95	1	0.4875
11 月 21 日	0.4840	0.5723	0.95	1	0.4507
12 月 21 日	0.4775	0.5646	0.95	1	0.4338

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	年平均输出功率 (MW)	单位面积镜面平均输出功率 (kW/m ²)
0.5199	0.6147	0.95	1	61.0798	0.4961

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, 0, 80m)	7.7328m × 5.6877m	4.39	2750	120949.95

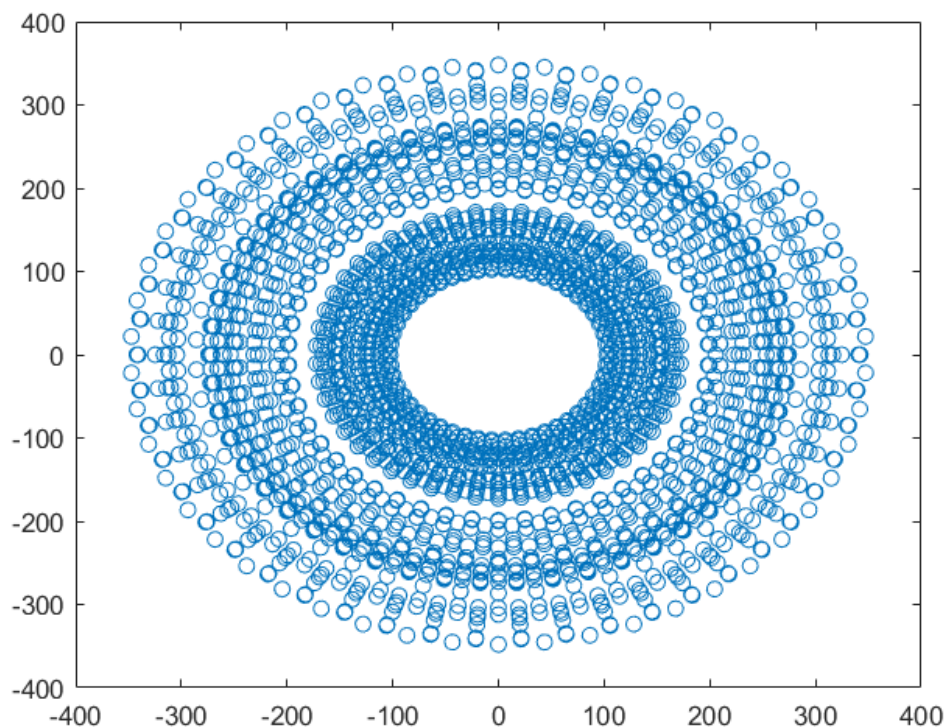


图 9 定日镜坐标显示

七、问题三模型

7.1 模型的建立

由问题二的计算结果可以看出，在其他条件相同时，方案一中定日镜均匀排布、径向交错的方法求解出来的单位镜面面积年平均输出热功率较大。所以在本题中，我们建立模型时，选择方案一中定日镜场的排布方式，结合实际情况和计算的需求做出合理的假设，规定每 5 圈定日镜的尺寸、高度都一致，其他模型的细节和问题二一致。

7.2 模型的求解

为了简化计算，我们选择在问题二建立模型的基础上，保证每 5 圈定日镜尺寸、高度一致，共设 55 圈，每圈镜子的数量为 50。

7.3 求解结果

按题要求，将计算的相关数据填入表中。

注：为计算便利，我们设定额定功率的参数为 $59MW$ 到 $61MW$ 在计算单位面积镜

面平均输出热功率计算时，采用近似计算公式

$$\frac{\text{单位面积镜面平均输出热功率}}{\text{年平均输出热功率}} \cdot 60\text{MW}$$

因为功率和题干中额定功率不一致，所以在计算出结果后要对计算结果进行修正，表格中的数据已经经过修正

* 部分数据四舍五入保留 4 位小数

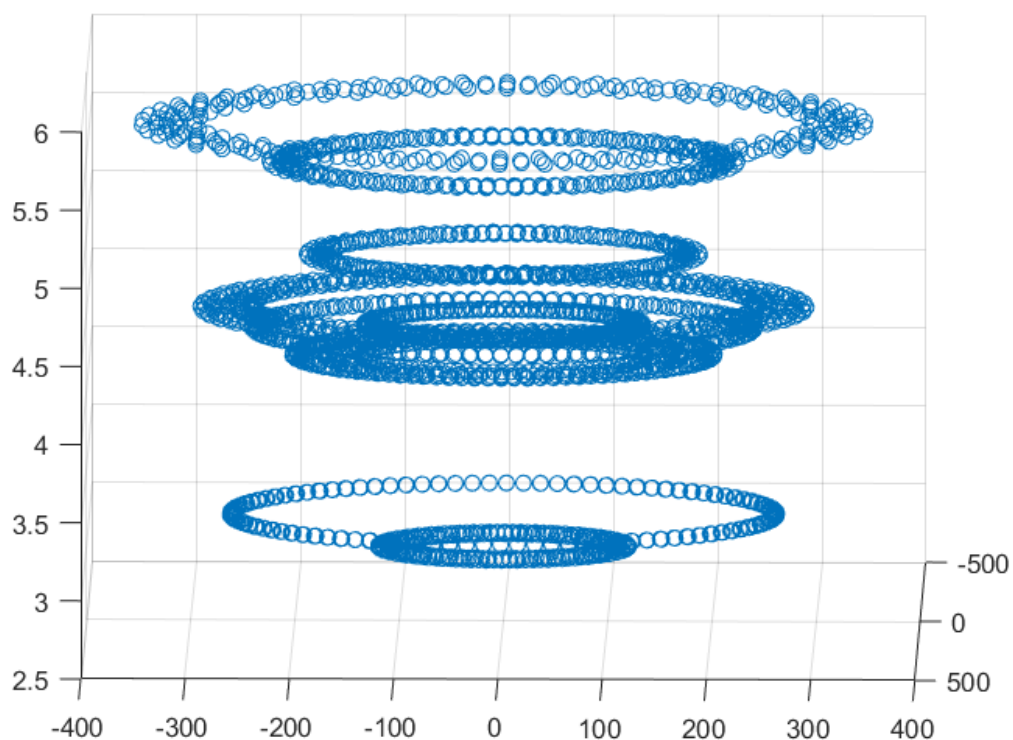


图 10 定日镜坐标显示

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 (kW/m ²)
1 月 21 日	0.4853	0.5737	0.95	1	0.4541
2 月 21 日	0.5041	0.5959	0.95	1	0.4910
3 月 21 日	0.5234	0.6187	0.95	1	0.5178
4 月 21 日	0.5408	0.6391	0.95	1	0.5356
5 月 21 日	0.5504	0.6505	0.95	1	0.5438
6 月 21 日	0.5534	0.6540	0.95	1	0.5461
7 月 21 日	0.5503	0.6503	0.95	1	0.5437
8 月 21 日	0.5401	0.6383	0.95	1	0.5350
9 月 21 日	0.5224	0.6174	0.95	1	0.5166
10 月 21 日	0.5017	0.5930	0.95	1	0.4870
11 月 21 日	0.4837	0.5718	0.95	1	0.4502
12 月 21 日	0.4773	0.5642	0.95	1	0.4333

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 (kW/m ²)
0.5194	0.6139	0.95	1	59.8932	0.5054

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, 0, 80m)	-	-	2750	118717.847

八、模型的评价

8.1 模型的优点

- 采用网格化处理的方法，利用微分的思想，当镜面被分成的网格足够小时，可以使用网格中心代替整个网格进行光线处理
- 利用遗传算法可以对简化模型进行较好的优化
- 符合实际要求，科学可行

8.2 模型的缺点改进

- 在网格化处理镜面时， n 应该尽量大，这样才能保证使用网格中心代替整个网格的合理性
- 问题二三中设定参数过多，导致模型过于简化，可继续优化

8.3 模型的推广

- 问题一中可以考虑定日镜的阴影的影响。考虑每一面定日镜在其他定日镜上的阴影遮挡，以此进行模型的优化
- 问题二和问题三中的模型可以减少参数的设定，例如塔的位置，以此进行进一步优化

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(6):4.DOI:10.3969/j.issn.1006-8554.2021.06.002.
- [2] 李雅雯. 塔式太阳能定日镜场聚光系统控制策略研究 [D]. 华北电力大学 (保定)[2023-09-10].
- [3] 张耀明. 太阳能热发电技术 [J]. 山西能源与节能, 2009.DOI:CNKI:SUN: SXJL.0.2009-03-005.
- [4] 高博, 刘建兴, 孙浩, 等. 基于自适应引力搜索算法的定日镜场优化布置 [J]. 太阳能学报, 2022, 43(10):119-125.DOI:10.19912/j.0254-0096.tynxb.2021-0397.
- [5] 程小龙, 尹延国, 马少波. 塔式电站定日镜场布局的优化设计研究 [J]. 能源与环境, 2018(2):4.DOI:CNKI:SUN:FJNJ.0.2018-02-036.
- [6] 关弘扬, 张艳梅, 廖文俊, 等. 不同定日镜场排布形式对镜场光学效率的影响 [J]. 装备制造机械, 2017(1):5.DOI:10.3969/j.issn.1662-0555.2017.01.002.

附录 A 文件列表

文件名	文件描述
<i>main1.m</i>	问题一的主函数
<i>main2.m</i>	问题二的主函数
<i>main3.m</i>	问题三的主函数
<i>result.m</i>	求解结果的目的函数
<i>cal_nvec.m</i>	计算法向量函数
<i>result2.xlsx</i>	问题二结果表格
<i>result3.xlsx</i>	问题三结果表格
<i>result.zip</i>	求解问题的所有 <i>matlab</i> 代码压缩包

附录 B 代码

2.1 问题一

cal_alps_gams.m

```
1 function [alps,gams] = cal_alps_gams(D,phi,ST)
2 % alps 太阳高度角
3 % gams 太阳方位角
4 % 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 % ST 为当地时间
6 % del 太阳赤纬角
7 % phi 当地纬度 北纬为正
8 % ome 太阳时角
9     ome = (pi/12) * (ST - 12) ;
10    del = asin( sin(2*pi*D/365) * sin(2*pi*23.45/365)) ;
11    alps = asin(cos(del) * cos(phi) * cos(ome) + sin(del) * sin(phi)) ;
12    gams = acos((sin(del) - sin(alps) * sin(phi)) / (cos(alps) * cos(phi))) ;
13 end
```

cal_dhr.m

```
1 function dhr = cal_dhr(mir_loc,mir_row,hot_x,hot_y,hot_z)
2 %dhr 镜面中心到集热器中心的距离
3 %mir_loc 镜面中心坐标
4 %mir_num 镜子数量
5 %hot_xyz 集热器中心坐标
6     hot = repmat([hot_x,hot_y,hot_z],mir_row);
```

```

7     dhr = ((mir_loc(:,1) - hot(:,1)).^2 + (mir_loc(:,2) - hot(:,2)).^2 + (mir_loc
    (:,3) - hot(:,3)).^2).^0.5;
8 end

```

cal_DNI.m

```

1 function DNI = cal_DNI(alps)
2 % DNI 法向直接辐射辐照度
3 % alps 太阳高度角
4     H = 3.0 ;
5     a = 0.4237 - 0.00821 * (6 - H)^2;
6     b = 0.5055 + 0.00595 * (6.5 - H)^2 ;
7     c = 0.2711 + 0.01858 * (2.5 - H)^2 ;
8     DNI = 1.366 * (a + b * exp(-c/(sin(alps))));
9 end

```

cal_eta.m

```

1 function eta = cal_eta(shadow_loss,miss_loss,total_loss,dhr,eta_cos,div_num)
2 %yita 定日镜的光学效率
3 %shadowloss 阴影遮挡损失
4 %inline 入射光线向量
5 %nvec 法线向量
6 %dhr 镜面中心到集热器中心距离
7 %Er 集热器接受的能量
8 %Ef 镜面全反射能量
9 %El 阴影遮挡损失能量
10 eta_sb = 1 - shadow_loss/total_loss;
11 if dhr <= 1000
12     eta_at = 0.99321 - 0.0001176 * dhr + 0.0000000197 * dhr^2 ;
13 else
14     eta_at = exp(-0.0001106 * dhr); %来自参考文献
15 end
16 eta_trunc = (div_num * div_num - shadow_loss - miss_loss) / (div_num * div_num -
    shadow_loss) ;
17 eta = eta_sb * eta_cos * eta_at * eta_trunc * 0.92;
18 end

```

cal_nvec.m

```

1 function [nvec_x,nvec_y,nvec_z] = cal_nvec(tower_x,tower_y,tower_z,mir_x,mir_y,
    mir_z,alps,gams)
2 %nvec_xyz 定日镜的法向量
3 %tower_xyz 吸收塔的坐标
4 %mir_xyz 定日镜的坐标
5 %alps 太阳高度角
6 %gams 太阳方位角
7     inline = [-cos(alps)*sin(gams) ,-cos(alps)*cos(gams) ,-sin(alps)];
8     outline = [(mir_x-tower_x)/((mir_x-tower_x)^2+(mir_y-tower_y)^2+(mir_z-tower_z)
    ^2),(mir_y-tower_y)/((mir_x-tower_x)^2+(mir_y-tower_y)^2+(mir_z-tower_z)^2) ,(

```

```

    mir_z-tower_z)/((mir_x-tower_x)^2+(mir_y-tower_y)^2+(mir_z-tower_z)^2)];
9     nvec = (inline + outline) / 2;
10    nvec_x = nvec(:,1);
11    nvec_y = nvec(:,2);
12    nvec_z = nvec(:,3);
13 end

```

cal_sl.m

```

1 function [total,shadow,miss] = cal_sl(x0,y0,z0,nvec_x0,nvec_y0,nvec_z0,x1,y1,z1,
    nvec_x1,nvec_y1,nvec_z1,tower_x,tower_y,tower_z,alps,gams,row,col,div_num,
    total_loss_num,shadow_loss_num,miss_loss_num)
2 %xyz0 反射镜坐标
3 %nvec_xyz0 反射镜法向量
4 %xyz1 遮挡镜坐标
5 %nvec_xyz1 遮挡镜法向量
6 %tower_xyz 塔坐标
7 %alps,gams 太阳高度角与方位角
8 %row col第row行col列网格
9 %div_num 分的数量 div_num * div_num
10 %total_loss_num shadow_loss_num miss_loss_num无效网格个数
11 %j,k 日期时间
12     x2 = x0 + (nvec_y0 / div_num) * (((-nvec_x0 * nvec_z0) * (col - (div_num + 1) /
        2) / (sqrt(nvec_x0 ^2 + nvec_y0 ^2)))) + (-nvec_y0 * (row - (div_num + 1) / 2) /
        (sqrt(nvec_x0 ^2 + nvec_y0 ^2)))) ;
13     y2 = y0 + (nvec_y0 / div_num) * (((-nvec_y0 * nvec_z0) * (col - (div_num + 1) /
        2) / (sqrt(nvec_x0 ^2 + nvec_y0 ^2)))) + (nvec_x0 * (row - (div_num + 1) / 2) /
        (sqrt(nvec_x0 ^2 + nvec_y0 ^2)))) ;
14     z2 = z0 + (nvec_y0 / div_num) * ((nvec_x0 ^2 + nvec_y0 ^2) * (col - (div_num +
        1) / 2) / (sqrt(nvec_x0 ^2 + nvec_y0 ^2))) ; % 网格的坐标
15     syms f1(x,y,z) f2(x,y,z) f3(x,y,z) f4(x,y,z) f5(x,y,z);
16     f1(x,y,z) = x2 + nvec_x1 * (z-z2)/ nvec_z1 - x;
17     f2(x,y,z) = y2 + nvec_y1 * (z-z2)/nvec_z1 - y;%反射光线方程
18     f3(x,y,z) = (-nvec_y1 * (x-x1) + nvec_x1 * (y-y1))/(nvec_x1 ^2 + nvec_y1 ^2);
19     f4(x,y,z) = (-nvec_x1 * nvec_z1 * (x-x1)-nvec_y1 * nvec_z1 * (y - y1) + (
        nvec_x1 ^2+ nvec_y1 ^2) * (z - z1))/(sqrt(nvec_x1^2+nvec_y1^2)) ;
20     f5(x,y,z) = (x-tower_x)^2 + (y-tower_y)^2 ; %塔的xy平面方程
21     eqns1 = [f1(x,y,z) == 0 , f2(x,y,z) == 0, f3(x,y,z) >= -3,f3(x,y,z) <= 3,f4(x,y
        ,z) >= -3,f4(x,y,z) <= 3]; %平面镜区域约束
22     sol1 = solve(eqns1,x,y,z,'Real',true) ;
23     if isempty(sol1)%是否被定日镜遮挡
24         total_loss_num = total_loss_num + 1;
25         miss_loss_num = miss_loss_num + 1;
26     else
27         if (((abs((cos(gams) * x2 - sin(gams) * y2)) <= 3.5) && ((-sin(gams) * x2 -
            cos(gams) * y2) >=0) && (tan(alps) * (-sin(gams) * x2 -cos(gams) * y2) <= 84) &&
            ((sin(gams) * x2 + cos(gams) * y2 - (z2 - 84) / tan(alps)) >=0)) || ((x2 - sin(
            gams) * (z2 - 84) /tan(alps))^2 + (y2 - cos(gams) * (z2 - 84) /tan(alps))^2 <=

```



```

49/4))
28     shadow_loss_num = shadow_loss_num + 1; % 被塔的影子遮挡
29     total_loss_num = total_loss_num + 1;
30     else
31         eqns2 = [f1(x,y,z) == 0 , f2(x,y,z) == 0, f5(x,y,z) == 49, z - tower_z <=
4, z - tower_z >= -4];
32         sol2 = solve(eqns2 ,x ,y ,z , 'Real', true) ;
33         if isempty(sol2)
34             total_loss_num = total_loss_num + 1;
35             miss_loss_num = miss_loss_num + 1;
36         end %没有打到吸收塔+1
37     end
38
39 end % 被挡住 +1
40 total = total_loss_num;
41 shadow = shadow_loss_num;
42 miss = miss_loss_num ;
43 end

```

main1.m

```

1 %% 第一题
2
3 %% 清空工作区
4 clear;
5 clc;
6 tic; %开启计时器
7
8 %% 初始化参数
9 mir_data = xlsread('C:\Users\86185\Desktop\a\题目\附件.xlsx') ; %读取定日镜坐标
10 height = 4.*ones(m,1); %定日镜的高度
11 mir_loc = [mir_data height] ; %定日镜xyz坐标
12 longitude_E = deg2rad(98.5) ;
13 latitude_N = deg2rad(39.4) ; %经纬度信息
14 tower = [0,0,80] ; %吸收塔坐标和高度
15 time = [9 10.5 12 13.5 15]; %时间点
16 disp("初始化结束")
17 %% 计算太阳高度角和太阳方位角
18 %2023年春分3.21
19 D = [-59, -28, 0, 31, 61, 92, 122, 153, 184, 214, 245, 275] ;
20 phi = latitude_N;
21 ST = time ;
22 alps = zeros(12,5) ;
23 gams = zeros(12,5) ; %初始化数据
24 for i = 1 : 12
25     for j = 1 : 5
26         [alps(i,j), gams(i,j)] = cal_alps_gams(D(i), phi, ST(j)) ; %计算高度角和方位角
27     end

```

```

28 end
29 gams(:,4) = 2*pi - gams(:,4);
30 gams(:,5) = 2*pi - gams(:,5);% ST >= 12
31 gams(:, :) = real(gams(:, :));
32 disp("太阳方位角与高度角计算完成")
33 %% 计算DNI法向直接辐射辐照度
34 DNI = zeros(size(alps)) ;
35 for i = 1 : 12
36     for j = 1 : 5
37         DNI(i,j) = cal_DNI(alps(i,j)) ;
38     end
39 end
40 disp("DNI is ok")
41 %% 计算dhr
42 [mir_row,mir_col] = size(mir_loc);
43 DHR = cal_dhr(mir_loc, mir_row, 0, 0, 80);
44 disp("dhr is ok")
45 %% 计算平面镜法向量
46 nvec_x = zeros(mir_row,12,5);
47 nvec_y = zeros(mir_row,12,5);
48 nvec_z = zeros(mir_row,12,5);
49 for i = 1 : mir_row
50     for j = 1 : 12
51         for k = 1 : 5
52             [nvec_x(i,j,k),nvec_y(i,j,k),nvec_z(i,j,k)] = cal_nvec(0,0,80,mir_loc(i
,1),mir_loc(i,2),mir_loc(i,3),alps(j,k),gams(j,k));
53         end
54     end
55 end
56 nvec_x = real(nvec_x);
57 nvec_y = real(nvec_y);
58 nvec_z = real(nvec_z);
59 disp("平面镜法向量计算结束")
60 %% 计算余弦效率
61 %入射光线法向量
62 %inline = [-cos(alps)*sin(gams) ,-cos(alps)*cos(gams) ,-sin(alps)];
63 for i = 1 : mir_row
64     for j = 1 : 12
65         for k = 1 : 5
66             inline = [-cos(alps(j,k))*sin(gams(j,k)) ,-cos(alps(j,k))*cos(gams(j,k
)) ,-sin(alps(j,k))]; %入射光线法向量
67             eta_cos(i,j,k) = abs(dot([nvec_x(i,j,k),nvec_y(i,j,k),nvec_z(i,j,k)],
inline)) ;
68         end
69     end
70 end
71 %计算平均余弦效率

```

```

72 eta_cos_ave = sum(eta_cos,[1,3])./(5 * mir_row);
73 disp("余弦效率计算完成")
74 %% 计算遮挡面积
75
76 %平面方程 nevx(x-x0)+nevy(y-y0)+nevz(z - z0) = 0
77 %反射光线法向量 outline = [mir_x-tower_x ,mir_y-tower_y ,mir_z-tower_z];
78 %平面约束方程1 -3 <= (-nvec_y(x-x0) + nvec_x(y-y0))/(nvec_x ^2 + nvec_y ^2) <=3
79 %平面约束方程2 -3 <= (-nvec_x * nvec_z(x-x0)-nvec_y * nvec_z(y - y0) + (nvec_x ^2
80 %+ nvec_y ^2)(z - z0))/(sqrt((a^2+b^2)*(a^2 + b^2 +c^2)))
81
82 total_loss_num = ones(mir_row,12,5); % total_loss_num 无效网格个数
83 shadow_loss_num = zeros(mir_row,12,5); %阴影损失
84 miss_loss_num = zeros(mir_row,12,5); %未击中吸收塔
85 div_num = 1001 ; % 分块数量
86 times = 0;
87 for i = 1 : mir_row
88     for j = 1 : 12
89         for k = 1 : 5
90             for row = 1 : div_num
91                 for col = 1 : div_num
92                     for l = 1 : mir_row
93                         temp = total_loss_num(i,j,k);
94                         if i ~= l && dot(mir_loc(l,:),mir_loc(l,:)) < dot(mir_loc
(i,:),mir_loc(i,:))
95                             break;
96                         else
97                             [total_loss_num(i,j,k),shadow_loss_num(i,j,k),
miss_loss_num(i,j,k) ] = cal_sl(mir_loc(i,1),mir_loc(i,2),mir_loc(i,3),nvec_x(i,
j,k),nvec_y(i,j,k),nvec_z(i,j,k),mir_loc(l,1),mir_loc(l,2),mir_loc(l,3),nvec_x(l
,j,k),nvec_y(l,j,k),nvec_z(l,j,k),0,0,80,alps(j,k),gams(j,k), row,col,div_num,
total_loss_num(i,j,k),shadow_loss_num(i,j,k),miss_loss_num(i,j,k));
98                             end
99                             if (total_loss_num(i,j,k)) > temp
100                                 %出现损失情况结束
101                                 break
102                             end
103                         end
104                     end
105                 end
106             end
107         end
108     end
109 %计算平均遮挡效率
110 eta_sb_ave = (sum(shadow_loss_num,[1,3]))./(div_num*div_num*mir_row)./5 ;
111 %计算平均截断效率
112 temp1 = div_num^2 .* ones(mir_row,12,5);
113 eta_trunc_ave = sum((temp1 - total_loss_num)./(temp1 - shadow_loss_num),[1,3])

```

```

        ./(5*mir_row) ;
114 %% 计算光学效率
115 for i = 1 : mir_row
116     for j = 1 : 12
117         for k = 1 : 5
118             eta_time(i,j,k) = cal_eta(shadow_loss_num(i,j,k),miss_loss_num(i,j,k),
                total_loss_num(i,j,k),DHR(i),eta_cos(i,j,k),div_num);
119         end
120     end
121 end
122 %具体日期时间镜子的光学效率
123 %计算日定日镜场的平均光学效率
124 eta_ave_data = sum(eta_time,[1,3])./(5*mir_row);
125 disp("效率计算完成")
126 %% 单位面积镜面平均输出热功率
127 for i = 1 : 12
128     for j = 1 : 5
129         E(i,j) = DNI(i,j) * eta_ave_data(j) * (1 - eta_sb_ave(i));
130     end
131 end
132 E_ave = sum(E ,2)./5;
133 toc;
134 disp("end")

```

2.2 问题二

2.2.1 结果求取

cal_alps_gams.m

```

1 function [alps,gams] = cal_alps_gams(D,phi,ST)
2 % alps 太阳高度角
3 % gams 太阳方位角
4 % 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 % ST 为当地时间
6 % del 太阳赤纬角
7 % phi 当地纬度 北纬为正
8 % ome 太阳时角
9     ome = (pi/12) * (ST - 12) ;
10     del = asin( sin(2*pi*D/365) * sin(2*pi*23.45/365)) ;
11     alps = asin(cos(del) * cos(phi) * cos(ome) + sin(del) * sin(phi)) ;
12     gams = acos((sin(del) - sin(alps) * sin(phi)) / ( cos(alps) * cos(phi))) ;
13 end

```

cal_dhr.m

```

1 function dhr = cal_dhr(mir_loc,mir_row,hot_x,hot_y,hot_z)

```

```

2 %dhr 镜面中心到集热器中心的距离
3 %mir_loc 镜面中心坐标
4 %mir_num 镜子数量
5 %hot_xyz 集热器中心坐标
6     hot = repmat([hot_x,hot_y,hot_z],mir_row);
7     dhr = ((mir_loc(:,1) - hot(:,1)).^2 + (mir_loc(:,2) - hot(:,2)).^2 + (mir_loc
        (:,3) - hot(:,3)).^2).^0.5;
8 end

```

cal_DNI.m

```

1 function DNI = cal_DNI(alps)
2 % DNI 法向直接辐射辐照度
3 % alps 太阳高度角
4     H = 3.0 ;
5     a = 0.4237 - 0.00821 * (6 - H)^2;
6     b = 0.5055 + 0.00595 * (6.5 - H)^2 ;
7     c = 0.2711 + 0.01858 * (2.5 - H)^2 ;
8     DNI = 1.366 * (a + b * exp(-c/(sin(alps))));
9 end

```

cal_eta.m

```

1 function eta = cal_eta(dhr,eta_cos,eta_sb,eta_trunc)
2 %yita 定日镜的光学效率
3 %shadowloss 阴影遮挡损失
4 %inline 入射光线向量
5 %nvec 法线向量
6 %dhr 镜面中心到集热器中心距离
7 %Er 集热器接受的能量
8 %Ef 镜面全反射能量
9 %El 阴影遮挡损失能量
10 if dhr <= 1000
11     eta_at = 0.99321 - 0.0001176 * dhr + 0.0000000197 * dhr^2 ;
12 else
13     eta_at = exp(-0.0001106 * dhr); %来自参考文献
14 end
15 eta = eta_sb * eta_cos * eta_at * eta_trunc * 0.92;
16 end

```

cal_nvec.m

```

1 function [nvec_x,nvec_y,nvec_z] = cal_nvec(tower_x,tower_y,tower_z,mir_x,mir_y,
        mir_z,alps,gams)
2 %nvec_xyz 定日镜的法向量
3 %tower_xyz 吸收塔的坐标
4 %mir_xyz 定日镜的坐标
5 %alps 太阳高度角
6 %gams 太阳方位角
7     inline = [-cos(alps)*sin(gams) ,-cos(alps)*cos(gams) ,-sin(alps)];

```

```

8     total = sqrt((mir_x-tower_x)^2+(mir_y-tower_y)^2+(mir_z-tower_z)^2);
9     outline = [(mir_x-tower_x)/total,(mir_y-tower_y)/total ,(mir_z-tower_z)/total];
10    nvec = (inline + outline) / 2;
11    nvec_x = nvec(:,1);
12    nvec_y = nvec(:,2);
13    nvec_z = nvec(:,3);
14 end

```

cal_sl.m

```

1 function [total,shadow,miss] = cal_sl(x0,y0,z0,nvec_x0,nvec_y0,nvec_z0,x1,y1,z1,
    nvec_x1,nvec_y1,nvec_z1,tower_x,tower_y,tower_z,alps,gams,row,col,div_num,
    total_loss_num,shadow_loss_num,miss_loss_num)
2 %xyz0 反射镜坐标
3 %nvec_xyz0 反射镜法向量
4 %xyz1 遮挡镜坐标
5 %nvec_xyz1 遮挡镜法向量
6 %tower_xyz 塔坐标
7 %alps,gams 太阳高度角与方位角
8 %row col第row行col列网格
9 %div_num 分的数量 div_num * div_num
10 %total_loss_num shadow_loss_num miss_loss_num无效网格个数
11 %j,k 日期时间
12     x2 = x0 + (nvec_y0 / div_num) * (((-nvec_x0 * nvec_z0) * (col - (div_num + 1) /
        2) / (sqrt(nvec_x0 ^2 + nvec_y0 ^2))) + (-nvec_y0 * (row - (div_num + 1) / 2) /
        (sqrt(nvec_x0 ^2 + nvec_y0 ^2)))) ;
13     y2 = y0 + (nvec_y0 / div_num) * (((-nvec_y0 * nvec_z0) * (col - (div_num + 1) /
        2) / (sqrt(nvec_x0 ^2 + nvec_y0 ^2))) + (nvec_x0 * (row - (div_num + 1) / 2) /
        (sqrt(nvec_x0 ^2 + nvec_y0 ^2)))) ;
14     z2 = z0 + (nvec_y0 / div_num) * ((nvec_x0 ^2 + nvec_y0 ^2) * (col - (div_num +
        1) / 2) / (sqrt(nvec_x0 ^2 + nvec_y0 ^2))) ; % 网格的坐标
15     syms f1(x,y,z) f2(x,y,z) f3(x,y,z) f4(x,y,z) f5(x,y,z);
16     f1(x,y,z) = x2 + nvec_x1 * (z-z2)/ nvec_z1 - x;
17     f2(x,y,z) = y2 + nvec_y1 * (z -z2) /nvec_z1 - y;%反射光线方程
18     f3(x,y,z) = (-nvec_y1 * (x-x1) + nvec_x1 * (y-y1))/(nvec_x1 ^2 + nvec_y1 ^2);
19     f4(x,y,z) = (-nvec_x1 * nvec_z1 * (x-x1)-nvec_y1 * nvec_z1 * (y - y1) + (
        nvec_x1 ^2+ nvec_y1 ^2) * (z - z1))/(sqrt(nvec_x1^2+nvec_y1^2)) ;
20     f5(x,y,z) = (x-tower_x)^2 + (y-tower_y)^2 ; %塔的xy平面方程
21     eqns1 = [f1(x,y,z) == 0 , f2(x,y,z) == 0, f3(x,y,z) >= -3,f3(x,y,z) <= 3,f4(x,y
        ,z) >= -3,f4(x,y,z) <= 3]; %平面镜区域约束
22     sol1 = solve(eqns1,x,y,z,'Real',true) ;
23     if isempty(sol1)%是否被定日镜遮挡
24         total_loss_num = total_loss_num + 1;
25         miss_loss_num = miss_loss_num + 1;
26     else
27         if (((abs((cos(gams) * x2 - sin(gams) * y2)) <= 3.5) && ((-sin(gams) * x2 -
            cos(gams) * y2) >=0) && (tan(alps) * (-sin(gams) * x2 -cos(gams) * y2) <= 84) &&
            ((sin(gams) * x2 + cos(gams) * y2 - (z2 - 84) / tan(alps)) >=0)) || ((x2 - sin(

```

```

gams) * (z2 - 84) /tan(alps))^2 + (y2 - cos(gams) * (z2 - 84) /tan(alps))^2 <=
49/4))
28         shadow_loss_num = shadow_loss_num + 1; % 被塔的影子遮挡
29         total_loss_num = total_loss_num + 1;
30     else
31         eqns2 = [f1(x,y,z) == 0 , f2(x,y,z) == 0,f5(x,y,z) == 49, z - tower_z <=
4,z - tower_z >= -4];
32         sol2 = solve(eqns2 ,x ,y ,z , 'Real',true) ;
33         if isempty(sol2)
34             total_loss_num=total_loss_num + 1;
35             miss_loss_num= miss_loss_num + 1;
36         end %没有打到吸收塔+1
37     end
38
39 end % 被挡住 +1
40 total = total_loss_num;
41 shadow = shadow_loss_num;
42 miss = miss_loss_num ;
43 end

```

main2_2.m

```

1 %% 第二问
2 tic;
3 clear;
4 clc;
5
6 %% 初始化
7 ret = [132.758373    107.0983616 151.228255   251.0839529 123.9059299 286.3469349
312.6514044 272.577485   165.9978693 199.6186359 104.6813386 172.5810455
272.832824 266.2402695 227.4790032 248.9953365 133.9751647 231.5721848
110.765223 245.3282343 324.7464474 312.1433231 253.5829274 210.2131414
304.2790922 198.7345125 274.3924298 113.6745338 222.7938798 150.8900731
119.2069625 154.8734911 342.1512827 164.7421712 262.4574946 161.3006176
259.8159588 303.8013548 340.8562035 136.2633661 126.9956364 199.917561
341.3947033 124.4272316 102.4679283 209.8451861 271.1057057 347.9998718
316.3913381 168.4501654 143.7050576 153.5385148 231.4462909 263.3848351
205.5575001 7.732806654 5.687691213 4.390039262
8 ] ; %模型优化后的zbest
9 [mir_x,mir_y,mir_z] = result(ret);
10 mir_x = mir_x(:);
11 mir_y = mir_y(:);
12 mir_z = mir_z(:);
13 mir_loc = [mir_x,mir_y,mir_z];
14 writematrix(mir_loc, '第二问坐标.xlsx', 'Sheet',1, 'Range', 'A2 :C2751');%定日镜坐标
15 longitude_E = deg2rad(98.5) ;
16 latitude_N = deg2rad(39.4) ; %经纬度信息
17 tower = [0,0,80] ; %吸收塔坐标和高度

```

```

18 time = [9 10.5 12 13.5 15]; %时间点
19
20 %% 计算太阳高度角和太阳方位角
21 %2023年春分3.21
22 D = [-59, -28, 0, 31, 61, 92, 122, 153, 184, 214, 245, 275] ;
23 phi = latitude_N;
24 ST = time ;
25 alps = zeros(12,5) ;
26 gams = zeros(12,5) ; %初始化数据
27 for i = 1 : 12
28     for j = 1 : 5
29         [alps(i,j),gams(i,j)] = cal_alps_gams(D(i),phi,ST(j)) ; %计算高度角和方位角
30     end
31 end
32 gams(:,4) = 2*pi - gams(:,4);
33 gams(:,5) = 2*pi - gams(:,5);% ST >= 12
34 gams(:, :) = real(gams(:, :));
35 disp("太阳方位角与高度角计算完成")
36 %% 计算DNI法向直接辐射辐照度
37 DNI = zeros(size(alps)) ;
38 for i = 1 : 12
39     for j = 1 : 5
40         DNI(i,j) = cal_DNI(alps(i,j)) ;
41     end
42 end
43 disp("DNI is ok")
44 %% 计算dhr
45 [mir_row,mir_col] = size(mir_loc);
46 DHR = cal_dhr(mir_loc, mir_row, 0, 0, 80);
47 disp("dhr is ok")
48 %% 计算平面镜法向量(待简化)
49 nvec_x = zeros(mir_row,12,5);
50 nvec_y = zeros(mir_row,12,5);
51 nvec_z = zeros(mir_row,12,5);
52 for i = 1 : mir_row
53     for j = 1 : 12
54         for k = 1 : 5
55             [nvec_x(i,j,k),nvec_y(i,j,k),nvec_z(i,j,k)] = cal_nvec(0,0,80,mir_loc(i
,1),mir_loc(i,2),mir_loc(i,3),alps(j,k),gams(j,k));
56         end
57     end
58 end
59 nvec_x = real(nvec_x);
60 nvec_y = real(nvec_y);
61 nvec_z = real(nvec_z);
62 disp("平面镜法向量计算结束")
63 %% 计算余弦效率

```



```

64 for i = 1 : mir_row
65     for j = 1 : 12
66         for k = 1 : 5
67             inline = [-cos(alps(j,k))*sin(gams(j,k)) , -cos(alps(j,k))*cos(gams(j,k)
68             )) , -sin(alps(j,k))]; %入射光线法向量
69             eta_cos(i,j,k) = abs(dot([nvec_x(i,j,k),nvec_y(i,j,k),nvec_z(i,j,k)],
70             inline)) ;
71         end
72     end
73 end
74 %计算平均余弦效率
75 eta_cos_ave = sum(eta_cos,[1,3])./(5 * mir_row);
76 writematrix(eta_cos_ave.', 'result2.xlsx', 'Sheet', 1, 'Range', 'B2 : B13');
77 disp("余弦效率计算完成")
78
79 %% 计算光学效率
80 eta_sb = 0.95;
81 eta_trunc = 1;
82 for i = 1 : mir_row
83     for j = 1 : 12
84         for k = 1 : 5
85             eta_time(i,j,k) = cal_eta(DHR(i),eta_cos(i,j,k),eta_sb,eta_trunc);
86         end
87     end
88 end
89 %具体日期时间镜子的光学效率
90 %计算日定日镜场的平均光学效率
91 eta_ave_data = sum(eta_time,[1,3])./(5*mir_row);
92 disp("效率计算完成")
93 writematrix(eta_ave_data.', 'result2.xlsx', 'Sheet', 1, 'Range', 'A2 : A13');
94 %% 单位面积镜面平均输出热功率
95 for i = 1 : 12
96     for j = 1 : 5
97         E(i,j) = DNI(i,j) * eta_ave_data(j) ;
98     end
99 end
100 E_ave = sum(E ,2)./5;
101 writematrix(E_ave, 'result2.xlsx', 'Sheet', 1, 'Range', 'E2 : E13');
102 disp("end")
103 toc;

```

result.m

```

1 function [mir_x,mir_y,mir_z] = result(ret)
2     for i = 1 : 2 : 55
3         for k = 1 : 50
4             mir_x(i,k) = ret(:,i) * cos(k * pi / 25);
5             mir_y(i,k) = ret(:,i) * sin(k * pi / 25);

```

```

6         mir_z(i,k) = ret(:,end) ;
7     end
8 end
9 for i = 2 : 2 : 55
10     for k = 1 : 50
11         mir_x(i,k) = ret(:,i) * cos(( 2 * k * pi + pi)/50);
12         mir_y(i,k) = ret(:,i) * sin((2 * k * pi + pi)/50);
13         mir_z(i,k) = ret(:,end) ;
14     end
15 end
16 end

```

2.2.2 求最优解——遗传算法

Code.m

```

1 function [ ret ] = Code( lenchrom, bound, alps, gams)
2 %±¼°□□½«±□-±□□□□□□□□±□□□□□□□³□□»~□,□□□□
3 %lenchrom    input:□□□峇□□
4 %bound        input:±□-μ□□□ □X
5 %ret          output:□□□□,□□□ □
6
7 while 1
8     pick = rand(1,lenchrom);
9     ret = bound(:,1)' + (bound(:,2) - bound(:,1))'.*pick;    %□□□□□ □
10    %4□□□□□□□□□□□□□□
11    flag = Test(lenchrom, bound, ret, alps, gams);
12    if flag                                             %□□□□□□□²¿¿□□Y-□□³□□»•
13        break
14    end
15 end
16 end

```

Cross.m

```

1 function [ ret ] = Cross( pcross,lenchrom,chrom,sizepop,bound,alps,gams)
2 %±¼°□□□□□»²□□□□
3 %pcross      input:½»²□□□□
4 %lenchrom    input:□□□□½°□
5 %chrom       inout:□□□□□/□□□
6 %sizepop     input:□□□¹□ḡ
7 %ret         output:½»²□□□□□□□
8 for n = 1:sizepop
9     %□□□□□□□},□□□□□□□□»²□
10    while 1
11        pick = rand(1,2);
12        if prod(pick)
13            break

```

```

14     end
15 end
16 index = ceil(pick*sizepop);
17 %1/2»2□□□□□□□□□□
18 while 1
19     pick = rand;
20     if pick
21         break
22     end
23 end
24 if pick > pcross
25     continue;
26 end
27 while 1 %flag == 0
28     %□□□□□□□□□□λ□□
29     while 1
30         pick = rand;
31         if pick
32             break
33         end
34     end
35     pos = ceil(pick*lenchrom);          %□□□□□□□□□-1/2□□□□»2□
36     pick = rand;                       %1/2»2瀧□
37     v1 = chrom(index(1),pos);
38     v2 = chrom(index(2),pos);
39     chrom(index(1),pos) = pick*v2 + (1-pick)*v1;
40     chrom(index(2),pos) = pick*v1 + (1-pick)*v2;    %1/2»2□□□□□
41     flag = [Test(lenchrom,bound,chrom(index(1),:),alps,gams) ...
42     Test(lenchrom,bound,chrom(index(2),:),alps,gams)];
43     if prod(flag)
44         break
45     end
46 end
47 end
48 ret = chrom;
49 end

```

E_limit.m

```

1 function [flag] = E_limit(ret, alps, gams)
2 % ret 种群
3 % alps 太阳高度角
4 % gams 太阳方位角
5 % flag 标志量
6 flag = 0;
7 eta_cos = 0;
8 DNI_ave = 0.9686 ;
9 eta_sb = 0.95;

```

```

10 eta_ref = 0.92;
11 for i = 1 : 55
12     eta_at = 0.99321 - 0.000176 * sqrt(ret(:,i)^2 + (ret(:,end) - 80)^2) +
        0.0000000197*(ret(:,i)^2 + (ret(:,end) - 80)^2);
13     for j = 1 : 12
14         for k = 1 : 5
15             eta_cos = eta_cos + 25 + 25*dot([-cos(alps(j,k)) * sin(gams(j,k)), -cos
                (alps(j,k)) * cos(gams(j,k)), -sin(alps(j,k))],[0,0,ret(:,end)-80])/(sqrt(ret(:,
                i)^2 + (ret(:,end)-80)^2));
16         end
17     end
18     eta(i) = eta_at * eta_cos * eta_sb * eta_ref / 60 ;
19     eta_cos = 0;
20 end
21 eta_ave = sum(eta) / 55;
22 E = eta_ave * DNI_ave * 55 * ret(:,end - 2) * ret(:,end - 1);
23 if E < 62000 && E > 59000
24     flag = 1;
25 end
26 end

```

E_show.m

```

1 function [E] = E_show(ret, alps, gams)
2 % ret 种群
3 % alps 太阳高度角
4 % gams 太阳方位角
5 % flag 标志量
6
7 R = ret(:,1 : end - 3); %逐步往外层
8 L = ret(:,end - 2:end - 1); %宽大于高
9 R = sort(R);
10 L = sort(L);
11 ret = [R,L,ret(end)];
12
13 E = 0;
14 eta_cos = 0;
15 DNI_ave = 0.9686 ;
16 eta_sb = 0.95;
17 eta_ref = 0.92;
18 for i = 1 : 55
19     eta_at = 0.99321 - 0.000176 * sqrt(ret(:,i)^2 + (ret(:,end) - 80)^2) +
        0.0000000197*(ret(:,i)^2 + (ret(:,end) - 80)^2);
20     for j = 1 : 12
21         for k = 1 : 5
22             eta_cos = eta_cos + 25 + 25*dot([-cos(alps(j,k)) * sin(gams(j,k)), -cos
                (alps(j,k)) * cos(gams(j,k)), -sin(alps(j,k))],[0,0,ret(:,end)-80])/...
23                 (sqrt(ret(:,i)^2 + (ret(:,end)-80)^2));

```



```

31 %初始化种群 GA初始化
32 GApop = Select(GApop, fitness, popsize);
33 %1/2选择
34 GApop = Cross(pc, lenchrom, GApop, popsize, bound, alps, gams);
35 %±初始化
36 GApop = Mutation(pm, lenchrom, GApop, popsize, [n, maxgen], bound, alps, gams);
37 disp(n);
38 for m = 1:popsize
39     fitness(m) = Fun(GApop(m,:));
40     if fitness(m) > fitnesszbest
41         zbest = GApop(m,:);
42         fitnesszbest = fitness(m);
43     end
44 end
45 y(n) = fitnesszbest;
46 end
47 y = y(end);
48 E = E_show(zbest, alps, gams);
49 end

```

main2.m

```

1 %%
2 %% 清空工作区
3 clear;
4 clc;
5 tic
6 alps = [0.308034749 0.479218728 0.542797023 0.479218728 0.308034749;
7     0.435317223 0.626614089 0.700033134 0.626614089 0.435317223;
8     0.578065947 0.795070156 0.883136602 0.795070156 0.578065947;
9     0.723764002 0.972576887 1.084295896 0.972576887 0.723764002;
10    0.818297398 1.091819612 1.230836123 1.091819612 0.818297398;
11    0.850781283 1.133363666 1.286774142 1.133363666 0.850781283;
12    0.817213864 1.090436258 1.229034224 1.090436258 0.817213864;
13    0.717606422 0.964926062 1.075324922 0.964926062 0.717606422;
14    0.570365776 0.785868001 0.872994973 0.785868001 0.570365776;
15    0.418506012 0.607024208 0.679013305 0.607024208 0.418506012;
16    0.297694773 0.467324258 0.53018622 0.467324258 0.297694773;
17    0.256033135 0.41949092 0.479562341 0.41949092 0.256033135;
18 ];
19 %太阳高度角 gams = xlsread('太阳高度角.xlsx');
20 gams = xlsread('太阳方位角.xlsx');
21 %% 遗传算法优化参数
22 [zbest,y,E] = GA(alps,gams);
23 disp(zbest)
24 disp(y)
25 disp(E)
26 toc

```

Mutation.m

```

1 function [ ret ] = Mutation( pmutation,lenchrom,chrom,sizepop,pop,bound,alps,gams)
2 %±¼°±±±±±±±±±±±±±±
3 %pmutation      input:±±±±±±±±
4 %lenchrom       input:±±±±±±±±
5 %chrom          input:±±±±±±/±±±
6 %sizepop        input:±±±±¹±±
7 %pop            input:µ±±±±±µ±±±»´±±±±°±±±±±±±±±»´±±±±±±±±
8 %ret            output:±±±±±±±±±±±±
9 for n = 1:sizepop
10     %±±±±±±±±±,±±±±±±±±±±±±±±
11     while 1
12         pick = rand;
13         if pick
14             break
15         end
16     end
17     index = ceil(pick*sizepop);
18     %±±±±±±±±±±±±±±±±±±±±±±±±±±
19     pick = rand;
20     if pick > pmutation
21         continue;
22     end
23     while 1
24         %±±±±±±±±±±
25         while 1
26             pick = rand;
27             if pick
28                 break
29             end
30         end
31         pos = ceil(pick*lenchrom);      %±±±±±±±±±±±±±±±±±±±±
32         v = chrom(n,pos);
33         v1 = v - bound(pos,1);
34         v2 = bound(pos,2) - v;
35         pick = rand;                    %±±±±±±±±±±
36         if pick > 0.5
37             delta = v2*( 1 - pick^( ( 1-pop(1)/pop(2) )^2 ) );
38             chrom(index,pos) = v + delta;
39         else
40             delta = v1*( 1 - pick^( ( 1 - pop(1)/pop(2) )^2 ) );
41             chrom(index,pos) = v - delta;
42         end                            %±±±±±±±±±±
43         if Test(lenchrom,bound,chrom(n,:),alps,gams)
44             break
45         end

```

```
46     end
47 end
48 ret = chrom;
49 end
```

Select.m

```

1 function [ ret, fitness ] = Select( individuals, fitness, sizepop )
2 %±¼°□□□□□□´□□□□□e□□□□□□□□□□□k□□□L»²□□□□□
3 %individuals      input:□□□□□y
4 %fitness          input:□□□¹□
5 %sizepop          input:□□□¹□g
6 %ret              outout:¾¹□□□□□□□□□
7
8 % □□□ k ``
9
10 % fitness = 1./fitness;           %³□□□□□¹□□□□□□□□□□¹□□□□□€
11 sumfitness = sum(fitness);         %□□□□□□□¹□
12 sumf = fitness./sumfitness;        %,□□□□□□□¹□□□□□□□□□□□¹□& □
13 index = zeros(1,length(sizepop));
14 for n = 1:sizepop                 %□sizepop´□□□□□
15     while 1
16         pick = rand;
17         if pick
18             break
19         end
20     end
21     for m = 1:sizepop
22         pick = pick - sumf(m);
23         %□□□□□□□¹□□□k□□□□
24         if pick < 0
25             index(n) = m;
26             break;               %□□□½□□□□□□□□□□□» •
27         end
28     end
29 end
30 individuals = individuals(index,:);
31 fitness = fitness(index);
32 ret = individuals;
33 end

```

Test.m

```
1 function [ flag ] = Test( lenchrom, bound, ret ,alps,gams)
2 %lenchrom    input:00000000
3 %bound       input:±0-μ0000 1X
4 %code        input:00000,000 0
5 %flag        output:¿00000000±0-
6 %³00±0-
```



```

7 flag = 0;
8 %1: i
9 %0: 2» i
10 R = ret(:,1 : end - 3); %
11 L = ret(:,end - 2:end - 1); %i
12 R = sort(R);
13 L = sort(L);
14 ret = [R,L,ret(end)];
15 for n = 1:lenschrom
16     if ret(n) > bound(n,1) && ret(n) < bound(n,2) && E_limit(ret,alps,gams) && ret
        (:,1) * sin(pi/25) >= 5 && ret(:,end - 1) <= 2*ret(:,end)
17         flag = 1;
18         break
19     end
20 end
21 end

```

2.2.3 求最优解 (2)

Code.m

```

1 function [ ret ] = Code( lenchrom, bound, alps, gams)
2 %±¼°□□½«±□-±□□□□□□□□ 壬□□□□□□□□³□□»~□,□□□□
3 %lenchrom    input:□□□ 峇 □□
4 %bound        input:±□-μ□□□ 1X
5 %ret          output:□□□□,□□□ □
6
7 while 1
8     pick = rand(1,lenchrom);
9     ret = bound(:,1)' + (bound(:,2) - bound(:,1))'.*pick;    %□□□□□ □
10    %4□□□□□□□□□□□□□□□□
11    flag = Test(lenchrom, bound, ret, alps, gams);
12    if flag                                                %□□□□□□□□²¿¿□□Y~□□³□□»•
13        break
14    end
15 end
16 end

```

Cross.m

```

1 function [ ret ] = Cross( pcross,lenchrom,chrom,sizepop,bound,alps,gams)
2 %±¼°□□□□□»²□□□□
3 %pcross      input:½»²□□□□
4 %lenchrom    input:□□□□½□□
5 %chrom       inout:□□□□□/□□□
6 %sizepop     input:□□□¹□ḡ
7 %ret         output:½»²□□□□□□□
8 for n = 1:sizepop

```

```

9      %%%%%%%%%},%%%%%%%%H»²
10     while 1
11         pick = rand(1,2);
12         if prod(pick)
13             break
14         end
15     end
16     index = ceil(pick*sizepop);
17     %½»²%%%%%%%%
18     while 1
19         pick = rand;
20         if pick
21             break
22         end
23     end
24     if pick > pcross
25         continue;
26     end
27     while 1 %flag == 0
28         %%%%%%%%%λ
29         while 1
30             pick = rand;
31             if pick
32                 break
33             end
34         end
35         pos = ceil(pick*lenchrom);          %%%%%%%%%¼H»²
36         pick = rand;                        %½»²灌
37         v1 = chrom(index(1),pos);
38         v2 = chrom(index(2),pos);
39         chrom(index(1),pos) = pick*v2 + (1-pick)*v1;
40         chrom(index(2),pos) = pick*v1 + (1-pick)*v2;    %½»²
41         flag = [Test(lenchrom,bound,chrom(index(1),:),alps,gams) ...
42                 Test(lenchrom,bound,chrom(index(2),:),alps,gams)];
43         if prod(flag)
44             break
45         end
46     end
47 end
48 ret = chrom;
49 end

```

D_limit.m

```

1 function [flag] = E_limit( lenchrom, bound, ret, alps, gams)
2 %lenchrom    input:染色体长度
3 %bound       input:变量的取值范围
4 %code        input:染色体的编码值

```

```

5 %flag      output:可行性标志变量
6 flag = 0;
7 eta_cos = 0;
8 DNI_ave = 0.9686 ;
9 eta_sb = 0.95;
10 eta_ref = 0.92;
11 for i = 1 : 48
12     eta_at = 0.99321 - 0.000176 * sqrt(ret(:,i)^2 + (ret(:,end) - 80)^2) +
        0.0000000197*(ret(:,i)^2 + (ret(:,end) - 80)^2);
13     for j = 1 : 12
14         for k = 1 : 5
15             eta_cos = eta_cos + 5 + 1 * (i - 1) + (5 + 1 * (i - 1))*dot([-cos(alps(
                j,k)) * sin(gams(j,k)), -cos(alps(j,k)) * cos(gams(j,k)), -sin(alps(j,k))],[0,0,
                ret(:,end)-80])/(sqrt(ret(:,i)^2 + (ret(:,end)-80)^2));
16         end
17     end
18     eta(i) = eta_at * eta_cos * eta_sb * eta_ref / 60 ;
19     eta_cos = 0;
20 end
21 eta_ave = sum(eta) / 40;
22 E = eta_ave * DNI_ave * 40 * ret(:,end - 2) * ret(:,end - 1);
23 if E < 62000 && E > 59000
24     flag = 1;
25 end
26 end

```

E_limit.m

```

1 function [flag] = E_limit( lenchrom, bound, ret, alps, gams)
2 %lenchrom    input:染色体长度
3 %bound       input:变量的取值范围
4 %code        input:染色体的编码值
5 %flag        output:可行性标志变量
6 flag = 0;
7 eta_cos = 0;
8 DNI_ave = 0.9686 ;
9 eta_sb = 0.95;
10 eta_ref = 0.92;
11 for i = 1 : 48
12     eta_at = 0.99321 - 0.000176 * sqrt(ret(:,i)^2 + (ret(:,end) - 80)^2) +
        0.0000000197*(ret(:,i)^2 + (ret(:,end) - 80)^2);
13     for j = 1 : 12
14         for k = 1 : 5
15             eta_cos = eta_cos + 5 + 1 * (i - 1) + (5 + 1 * (i - 1))*dot([-cos(alps(
                j,k)) * sin(gams(j,k)), -cos(alps(j,k)) * cos(gams(j,k)), -sin(alps(j,k))],[0,0,
                ret(:,end)-80])/(sqrt(ret(:,i)^2 + (ret(:,end)-80)^2));
16         end
17     end

```

```

18     eta(i) = eta_at * eta_cos * eta_sb * eta_ref / 60 ;
19     eta_cos = 0;
20 end
21 eta_ave = sum(eta) / 40;
22 E = eta_ave * DNI_ave * 40 * ret(:,end - 2) * ret(:,end - 1);
23 if E < 62000 && E > 59000
24     flag = 1;
25 end
26 end

```

E_show.m

```

1 function [E] = E_show( lenchrom, bound, ret, alps, gams)
2 %lenchrom    input:染色体长度
3 %bound      input:变量的取值范围
4 %code       input:染色体的编码值
5 %flag       output:可行性标志变量
6
7 R = ret(:,1 : end - 3); %逐步往外层
8 L = ret(:,end - 2:end - 1); %宽大于高
9 R = sort(R);
10 L = sort(L);
11 ret = [R,L,ret(end)];
12
13 E = 0;
14 eta_cos = 0;
15 DNI_ave = 0.9686 ;
16 eta_sb = 0.95;
17 eta_ref = 0.92;
18 for i = 1 : 48
19     eta_at = 0.99321 - 0.000176 * sqrt(ret(:,i)^2 + (ret(:,end) - 80)^2) +
20         0.000000197*(ret(:,i)^2 + (ret(:,end) - 80)^2);
21     for j = 1 : 12
22         for k = 1 : 5
23             eta_cos = eta_cos + 5 + 1 * (i - 1) + (5 + 1 * (i - 1))*dot([-cos(alps(
24                 j,k)) * sin(gams(j,k)), -cos(alps(j,k)) * cos(gams(j,k)), -sin(alps(j,k))],[0,0,
25                 ret(:,end)-80])/(sqrt(ret(:,i)^2 + (ret(:,end)-80)^2));
26         end
27     end
28     eta(i) = eta_at * eta_cos * eta_sb * eta_ref / 60 ;
29     eta_cos = 0;
30 end
31 eta_ave = sum(eta) / 40;
32 E = eta_ave * DNI_ave * 40 * ret(:,end - 2) * ret(:,end - 1);
33 end

```

Fun.m


```

42         if fitness(m) > fitnesszbest
43             zbest = GApop(m,:);
44             fitnesszbest = fitness(m);
45         end
46     end
47     y(n) = fitnesszbest;
48 end
49 y = y(end);
50 E = E_show(lenchrom,bound,zbest,alps,gams);
51 end

```

main2.m

```

1 %%
2 %% 清空工作区
3 clear;
4 clc;
5 tic
6 alps = [0.308034749 0.479218728 0.542797023 0.479218728 0.308034749;
7         0.435317223 0.626614089 0.700033134 0.626614089 0.435317223;
8         0.578065947 0.795070156 0.883136602 0.795070156 0.578065947;
9         0.723764002 0.972576887 1.084295896 0.972576887 0.723764002;
10        0.818297398 1.091819612 1.230836123 1.091819612 0.818297398;
11        0.850781283 1.133363666 1.286774142 1.133363666 0.850781283;
12        0.817213864 1.090436258 1.229034224 1.090436258 0.817213864;
13        0.717606422 0.964926062 1.075324922 0.964926062 0.717606422;
14        0.570365776 0.785868001 0.872994973 0.785868001 0.570365776;
15        0.418506012 0.607024208 0.679013305 0.607024208 0.418506012;
16        0.297694773 0.467324258 0.53018622 0.467324258 0.297694773;
17        0.256033135 0.41949092 0.479562341 0.41949092 0.256033135;
18 ];
19
20 gams = xlsread('太阳方位角.xlsx');
21 %% 遗传算法优化参数
22 [zbest,y,E] = GA(alps,gams);
23 R = zbest(:,1 : end - 3); %逐步往外层
24 L = zbest(:,end - 2:end - 1); %宽大于高
25 R = sort(R);
26 L = sort(L);
27 zbest = [R,L,zbest(end)];
28 disp(zbest)
29 disp(y)
30 disp(E)
31 toc

```

Mutation.m

```

1 function [ ret ] = Mutation( pmutation,lenchrom,chrom,sizepop,pop,bound,alps,gams)
2 %±¼°□□□□□□□□□□

```

```

3 %pmutation      input:±0000000
4 %lenchrom       input:000峴10
5 %chrom          input:00000/000
6 %sizepop        input:000±00
7 %pop            input:μ±0000μ00»´000°00000000»´0000000
8 %ret            output:±0000000000
9 for n = 1:sizepop
10     %0000000,00000006000
11     while 1
12         pick = rand;
13         if pick
14             break
15         end
16     end
17     index = ceil(pick*sizepop);
18     %±0000000000»•000006000
19     pick = rand;
20     if pick > pmutation
21         continue;
22     end
23     while 1
24         %±000λ00
25         while 1
26             pick = rand;
27             if pick
28                 break
29             end
30         end
31         pos = ceil(pick*lenchrom);      %00000000000000λ00
32         v = chrom(n,pos);
33         v1 = v - bound(pos,1);
34         v2 = bound(pos,2) - v;
35         pick = rand;                    %±00  0
36         if pick > 0.5
37             delta = v2*( 1 - pick^( ( 1-pop(1)/pop(2) )^2 ) );
38             chrom(index,pos) = v + delta;
39         else
40             delta = v1*( 1 - pick^( (1 - pop(1)/pop(2) )^2 ) );
41             chrom(index,pos) = v - delta;
42         end                            %±0000000
43         if Test(lenchrom,bound,chrom(n,:),alps,gams)
44             break
45         end
46     end
47 end
48 ret = chrom;
49 end

```

Select.m

```

1 function [ ret, fitness ] = Select( individuals, fitness, sizepop )
2 %±¼°□□†□ÿ□´□□□□□e□□□□□□□□□□□□□k□□□L»²□□□□□
3 %individuals      input: □□□□□Y
4 %fitness          input: □□□†□
5 %sizepop          input: □□□¹□g
6 %ret              output: ¾¹□□□□□□□□□□
7
8 %   □□□ k¨
9
10 % fitness = 1./fitness;                %³□□□□†□□□□□□□□□□†□□□□□€
11 sumfitness = sum(fitness);             %□□□□□□†□
12 sumf = fitness ./sumfitness;           %, □□□□□†□□□□□□□□□□†□ & □
13 index = zeros(1,length(sizepop));
14 for n = 1:sizepop                      %□sizepop´□□□□□
15     while 1
16         pick = rand;
17         if pick
18             break
19         end
20     end
21     for m = 1:sizepop
22         pick = pick - sumf(m);
23         %□□□□□†□□Θ□k□□□
24         if pick < 0
25             index(n) = m;
26             break;                    %□□□½₂□□□□□□□□□» •
27         end
28     end
29 end
30 individuals = individuals(index,:);
31 fitness = fitness(index);
32 ret = individuals;
33 end

```

Test.m

```
1 function [ flag ] = Test( lenchrom, bound, ret ,alps,gams)
2 %lenchrom      input:000峴10
3 %bound         input:±0-μ000 1X
4 %code          input:0000,000 0
5 %flag          output:¿0000000±0-
6 %³00±0-
7 flag = 0;
8 %1:¿000
9 %0:²»¿000
10 R = ret(:,1 : end - 3); %0000000
11 L = ret(:,end - 2:end - 1); %¿00000
```



```

12 R = sort(R);
13 L = sort(L);
14 ret = [R,L,ret(end)];
15 for n = 1:lenchrom
16     if ret(n) > bound(n,1) && ret(n) < bound(n,2) && E_limit(lenchrom,bound,ret,
        alps,gams) && ret(:,end - 1) <= 2*ret(:,end) && D_limit(lenchrom,bound,ret)
17         flag = 1;
18         break
19     end
20 end
21 end

```

2.3 问题三

2.3.1 结果求解

cal_alps_gams.m

```

1 function [alps,gams] = cal_alps_gams(D,phi,ST)
2 % alps 太阳高度角
3 % gams 太阳方位角
4 % 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 % ST 为当地时间
6 % del 太阳赤纬角
7 % phi 当地纬度 北纬为正
8 % ome 太阳时角
9     ome = (pi/12) * (ST - 12) ;
10    del = asin( sin(2*pi*D/365) * sin(2*pi*23.45/365)) ;
11    alps = asin(cos(del) * cos(phi) * cos(ome) + sin(del) * sin(phi)) ;
12    gams = acos((sin(del) - sin(alps) * sin(phi)) / ( cos(alps) * cos(phi))) ;
13 end

```

cal_dhr.m

```

1 function dhr = cal_dhr(mir_loc,mir_row,hot_x,hot_y,hot_z)
2 %dhr 镜面中心到集热器中心的距离
3 %mir_loc 镜面中心坐标
4 %mir_num 镜子数量
5 %hot_xyz 集热器中心坐标
6     hot = repmat([hot_x,hot_y,hot_z],mir_row);
7     dhr = ((mir_loc(:,1) - hot(:,1)).^2 + (mir_loc(:,2) - hot(:,2)).^2 + (mir_loc
        (:,3) - hot(:,3)).^2).^0.5;
8 end

```

cal_DNI.m

```

1 function DNI = cal_DNI(alps)
2 % DNI 法向直接辐射辐照度

```

```

3 % alps 太阳高度角
4     H = 3.0 ;
5     a = 0.4237 - 0.00821 * (6 - H)^2;
6     b = 0.5055 + 0.00595 * (6.5 - H)^2 ;
7     c = 0.2711 + 0.01858 * (2.5 - H)^2 ;
8     DNI = 1.366 * (a + b * exp(-c/(sin(alps))));
9 end

```

cal_eta.m

```

1 function eta = cal_eta(dhr,eta_cos,eta_sb,eta_trunc)
2 %yita 定日镜的光学效率
3 %shadowloss 阴影遮挡损失
4 %inline 入射光线向量
5 %nvec 法线向量
6 %dhr 镜面中心到集热器中心距离
7 %Er 集热器接受的能量
8 %Ef 镜面全反射能量
9 %El 阴影遮挡损失能量
10 if dhr <= 1000
11     eta_at = 0.99321 - 0.0001176 * dhr + 0.0000000197 * dhr^2 ;
12 else
13     eta_at = exp(-0.0001106 * dhr); %来自参考文献
14 end
15 eta = eta_sb * eta_cos * eta_at * eta_trunc * 0.92;
16 end

```

cal_nvec.m

```

1 function [nvec_x,nvec_y,nvec_z] = cal_nvec(tower_x,tower_y,tower_z,mir_x,mir_y,
    mir_z,alps,gams)
2 %nvec_xyz 定日镜的法向量
3 %tower_xyz 吸收塔的坐标
4 %mir_xyz 定日镜的坐标
5 %alps 太阳高度角
6 %gams 太阳方位角
7     inline = [-cos(alps)*sin(gams) ,-cos(alps)*cos(gams) ,-sin(alps)];
8     total = sqrt((mir_x-tower_x)^2+(mir_y-tower_y)^2+(mir_z-tower_z)^2);
9     outline = [(mir_x-tower_x)/total,(mir_y-tower_y)/total ,(mir_z-tower_z)/total];
10    nvec = (inline + outline) / 2;
11    nvec_x = nvec(:,1);
12    nvec_y = nvec(:,2);
13    nvec_z = nvec(:,3);
14 end

```

cal_sl.m

```

1 function [total,shadow,miss] = cal_sl(x0,y0,z0,nvec_x0,nvec_y0,nvec_z0,x1,y1,z1,
    nvec_x1,nvec_y1,nvec_z1,tower_x,tower_y,tower_z,alps,gams,row,col,div_num,
    total_loss_num,shadow_loss_num,miss_loss_num)

```

```

2 %xyz0 反射镜坐标
3 %nvec_xyz0 反射镜法向量
4 %xyz1 遮挡镜坐标
5 %nvec_xyz1 遮挡镜法向量
6 %tower_xyz 塔坐标
7 %alps,gams 太阳高度角与方位角
8 %row col第row行col列网格
9 %div_num 分的数量 div_num * div_num
10 %total_loss_num shadow_loss_num miss_loss_num无效网格个数
11 %j,k 日期时间
12     x2 = x0 + (nvec_y0 / div_num) * (((-nvec_x0 * nvec_z0) * (col - (div_num + 1) /
        2) / (sqrt(nvec_x0 ^2 + nvec_y0 ^2))) + (-nvec_y0 * (row - (div_num + 1) / 2) /
        (sqrt(nvec_x0 ^2 + nvec_y0 ^2)))) ;
13     y2 = y0 + (nvec_y0 / div_num) * (((-nvec_y0 * nvec_z0) * (col - (div_num + 1) /
        2) / (sqrt(nvec_x0 ^2 + nvec_y0 ^2))) + (nvec_x0 * (row - (div_num + 1) / 2) /
        (sqrt(nvec_x0 ^2 + nvec_y0 ^2)))) ;
14     z2 = z0 + (nvec_y0 / div_num) * ((nvec_x0 ^2 + nvec_y0 ^2) * (col - (div_num +
        1) / 2) / (sqrt(nvec_x0 ^2 + nvec_y0 ^2))) ; % 网格的坐标
15     syms f1(x,y,z) f2(x,y,z) f3(x,y,z) f4(x,y,z) f5(x,y,z);
16     f1(x,y,z) = x2 + nvec_x1 * (z-z2)/ nvec_z1 - x;
17     f2(x,y,z) = y2 + nvec_y1 * (z -z2) /nvec_z1 - y;%反射光线方程
18     f3(x,y,z) = (-nvec_y1 * (x-x1) + nvec_x1 * (y-y1))/(nvec_x1 ^2 + nvec_y1 ^2);
19     f4(x,y,z) = (-nvec_x1 * nvec_z1 * (x-x1)-nvec_y1 * nvec_z1 * (y - y1) + (
        nvec_x1 ^2+ nvec_y1 ^2) * (z - z1))/(sqrt(nvec_x1^2+nvec_y1^2)) ;
20     f5(x,y,z) = (x-tower_x)^2 + (y-tower_y)^2 ; %塔的xy平面方程
21     eqns1 = [f1(x,y,z) == 0 , f2(x,y,z) == 0, f3(x,y,z) >= -3,f3(x,y,z) <= 3,f4(x,y
        ,z) >= -3,f4(x,y,z) <= 3]; %平面镜区域约束
22     sol1 = solve(eqns1,x,y,z,'Real',true) ;
23     if isempty(sol1)%是否被定日镜遮挡
24         total_loss_num = total_loss_num + 1;
25         miss_loss_num = miss_loss_num + 1;
26     else
27         if (((abs((cos(gams) * x2 - sin(gams) * y2)) <= 3.5) && ((-sin(gams) * x2 -
            cos(gams) * y2) >=0) && (tan(alps) * (-sin(gams) * x2 -cos(gams) * y2) <= 84) &&
            ((sin(gams) * x2 + cos(gams) * y2 - (z2 - 84) / tan(alps)) >=0)) || ((x2 - sin(
            gams) * (z2 - 84) /tan(alps))^2 + (y2 - cos(gams) * (z2 - 84) /tan(alps))^2 <=
            49/4))
28             shadow_loss_num = shadow_loss_num + 1; % 被塔的影子遮挡
29             total_loss_num = total_loss_num + 1;
30         else
31             eqns2 = [f1(x,y,z) == 0 , f2(x,y,z) == 0,f5(x,y,z) == 49, z - tower_z <=
                4,z - tower_z >= -4];
32             sol2 = solve(eqns2 ,x ,y ,z ,'Real',true) ;
33             if isempty(sol2)
34                 total_loss_num=total_loss_num + 1;
35                 miss_loss_num= miss_loss_num + 1;
36             end %没有打到吸收塔+1

```

```

37         end
38
39     end % 被挡住 +1
40     total = total_loss_num;
41     shadow = shadow_loss_num;
42     miss = miss_loss_num ;
43 end

```

main.m

```

1 %% 第二问
2 tic;
3 clear;
4 clc;
5
6 %% 初始化
7 ret = [103.1564882  104.6967184 110.4261043 118.4611272 120.0432118 121.5808756
        122.9592307 130.1645589 131.7448211 133.887868  136.4237897 136.9701781
        161.2801785 168.3884154 170.75142   170.8290786 172.8452757 176.8685481
        180.5859606 188.2365497 193.9134366 194.2434875 197.2891748 201.1445132
        202.404949  203.6460565 206.1515852 214.1812849 215.1717824 224.1214086
        229.6150256 230.0964079 237.1296064 240.3913712 240.8809778 241.5416767
        242.5831556 243.0165298 255.4290513 258.8787525 259.5638978 260.0374288
        260.6185532 261.931686  262.2872784 264.9475489 279.2063159 279.7870819
        288.0881581 290.2774258 296.5782613 314.8480576 323.8717687 333.7464721
        346.8378663 5.023331903 7.785279233 7.580295666 7.892114853 6.300147807
        6.576719878 7.604540298 7.701937532 5.120641378 5.295472457 6.145115478
        7.603703553 5.464192916 6.915766321 6.698987907 7.475496312 5.647860811
        6.464013759 5.581412162 6.407037445 5.540113616 7.572202721 2.976455454
        4.396028239 4.194904347 4.845064343 4.203337237 5.438564028 4.376607456
        4.516108197 3.183481851 4.507653114 5.682135814
8 ]
9 [mir_x,mir_y,mir_z] = result(ret);
10 mir_x = mir_x(:);
11 mir_y = mir_y(:);
12 mir_z = mir_z(:);
13 mir_loc = [mir_x,mir_y,mir_z];
14 writematrix(mir_loc,'第三问坐标.xlsx','Sheet',1,'Range','A2 :C2751');%定日镜坐标
15 longitude_E = deg2rad(98.5) ;
16 latitude_N = deg2rad(39.4) ; %经纬度信息
17 tower = [0,0,80] ; %吸收塔坐标和高度
18 time = [9 10.5 12 13.5 15]; %时间点
19
20 %% 计算太阳高度角和太阳方位角
21 %2023年春分3.21
22 D = [-59, -28, 0, 31, 61, 92, 122, 153, 184, 214, 245, 275] ;
23 phi = latitude_N;
24 ST = time ;

```

```

25 alps = zeros(12,5) ;
26 gams = zeros(12,5) ; %初始化数据
27 for i = 1 : 12
28     for j = 1 : 5
29         [alps(i,j),gams(i,j)] = cal_alps_gams(D(i),phi,ST(j)) ; %计算高度角和方位角
30     end
31 end
32 gams(:,4) = 2*pi - gams(:,4);
33 gams(:,5) = 2*pi - gams(:,5);% ST >= 12
34 gams(:, :) = real(gams(:, :));
35 disp("太阳方位角与高度角计算完成")
36 %% 计算DNI法向直接辐射辐照度
37 DNI = zeros(size(alps)) ;
38 for i = 1 : 12
39     for j = 1 : 5
40         DNI(i,j) = cal_DNI(alps(i,j)) ;
41     end
42 end
43 disp("DNI is ok")
44 %% 计算dhr
45 [mir_row,mir_col] = size(mir_loc);
46 DHR = cal_dhr(mir_loc, mir_row, 0, 0, 80);
47 disp("dhr is ok")
48 %% 计算平面镜法向量(待简化)
49 nvec_x = zeros(mir_row,12,5);
50 nvec_y = zeros(mir_row,12,5);
51 nvec_z = zeros(mir_row,12,5);
52 for i = 1 : mir_row
53     for j = 1 : 12
54         for k = 1 : 5
55             [nvec_x(i,j,k),nvec_y(i,j,k),nvec_z(i,j,k)] = cal_nvec(0,0,80,mir_loc(i
,1),mir_loc(i,2),mir_loc(i,3),alps(j,k),gams(j,k));
56         end
57     end
58 end
59 nvec_x = real(nvec_x);
60 nvec_y = real(nvec_y);
61 nvec_z = real(nvec_z);
62 disp("平面镜法向量计算结束")
63 %% 计算余弦效率
64 for i = 1 : mir_row
65     for j = 1 : 12
66         for k = 1 : 5
67             inline = [-cos(alps(j,k))*sin(gams(j,k)) , -cos(alps(j,k))*cos(gams(j,k
)) , -sin(alps(j,k))]; %入射光线法向量
68             eta_cos(i,j,k) = abs(dot([nvec_x(i,j,k),nvec_y(i,j,k),nvec_z(i,j,k)],
inline)) ;

```

```

69         end
70     end
71 end
72 %计算平均余弦效率
73 eta_cos_ave = sum(eta_cos,[1,3])./(5 * mir_row);
74 writematrix(eta_cos_ave.','result2.xlsx','Sheet',1,'Range','B2 : B13');
75 disp("余弦效率计算完成")
76
77 %% 计算光学效率
78 eta_sb = 0.95;
79 eta_trunc = 1;
80 for i = 1 : mir_row
81     for j = 1 : 12
82         for k = 1 : 5
83             eta_time(i,j,k) = cal_eta(DHR(i),eta_cos(i,j,k),eta_sb,eta_trunc);
84         end
85     end
86 end
87 %具体日期时间镜子的光学效率
88 %计算日定日镜场的平均光学效率
89 eta_ave_data = sum(eta_time,[1,3])./(5*mir_row);
90 disp("效率计算完成")
91 writematrix(eta_ave_data.','result2.xlsx','Sheet',1,'Range','A2 : A13');
92 %% 单位面积镜面平均输出热功率
93 for i = 1 : 12
94     for j = 1 : 5
95         E(i,j) = DNI(i,j) * eta_ave_data(j) ;
96     end
97 end
98 E_ave = sum(E ,2)./5;
99 writematrix(E_ave,'result2.xlsx','Sheet',1,'Range','E2 : E13');
100 disp("end")
101 toc;

```

result.m

```

1 function [mir_x,mir_y,mir_z] = result(ret)
2 %RESULT 此处显示有关此函数的摘要
3 % 此处显示详细说明
4     for i = 1 : 2 : 55
5         for k = 1 : 50
6             mir_x(i,k) = ret(:,i) * cos(k * pi / 25);
7             mir_y(i,k) = ret(:,i) * sin(k * pi / 25);
8             mir_z(i,k) = ret(:,78+floor((i - 1)/5)) ;
9         end
10    end
11    for i = 2 : 2 : 55
12        for k = 1 : 50

```

```

13     mir_x(i,k) = ret(:,i) * cos(( 2 * k * pi + pi)/50);
14     mir_y(i,k) = ret(:,i) * sin((2 * k * pi + pi)/50);
15     mir_z(i,k) = ret(:,78+floor((i - 1)/5)) ;
16 end
17 end
18 % writematrix(mir_x,'第二问坐标.xlsx','Sheet',1,'Range','A2 :A2751');
19 % writematrix(mir_y,'第二问坐标.xlsx','Sheet',1,'Range','B2 :B2751');
20 % writematrix(mir_z,'第二问坐标.xlsx','Sheet',1,'Range','C2 :C2751');
21 end

```

2.3.2 优化模型——遗传算法

Code.m

```

1 function [ ret ] = Code( lenchrom,alps,gams)
2 %±¼°□□/¼«±□-±□□□□□□□ □□□□□□□³□□»~□,□□□□
3 %lenchrom    input:□□□ □□ □□
4 %bound       input:±□-μ□□□ □X
5 %ret         output:□□□□,□□□ □
6
7 while 1
8     pick = rand(1,lenchrom);
9     ret = bound(:,1)' + (bound(:,2) - bound(:,1)).*pick;    %□□□□ □
10    %/□□□□□□□□□□□□□□
11    flag = Test(lenchrom, bound, ret,alps,gams);
12    if flag                                                %□□□□□□□²¿¿□□□-□□³□□»•
13        break
14    end
15 end
16 end

```

Cross.m

```

1 function [ ret ] = Cross( pcross, lenchrom, chrom, sizepop, bound, alps, gams)
2 %±¼°□□□□□»²□□□□
3 %pcross      input:½»²□□□□
4 %lenchrom    input:□□□□½±¼°
5 %chrom       inout:□□□□□/□□□□
6 %sizepop     input:□□□¹□ḡ
7 %ret         output:½»²□□□□□□□□
8 for n = 1:sizepop
9     %□□□□□□□}, □□□□□□□□»²□
10    while 1
11        pick = rand(1,2);
12        if prod(pick)
13            break
14        end
15    end

```

```

16     index = ceil(pick*sizepop);
17     %1/2»2个染色体
18     while 1
19         pick = rand;
20         if pick
21             break
22         end
23     end
24     if pick > pcross
25         continue;
26     end
27     while 1 %flag == 0
28         %染色体交叉
29         while 1
30             pick = rand;
31             if pick
32                 break
33             end
34         end
35         pos = ceil(pick*lenchrom); %染色体交叉点
36         pick = rand; %1/2»2个染色体
37         v1 = chrom(index(1),pos);
38         v2 = chrom(index(2),pos);
39         chrom(index(1),pos) = pick*v2 + (1-pick)*v1;
40         chrom(index(2),pos) = pick*v1 + (1-pick)*v2; %1/2»2个染色体
41         flag = [Test(lenchrom,bound,chrom(index(1),:),alps,gams) ...
42             Test(lenchrom,bound,chrom(index(2),:),alps,gams)];
43         if prod(flag)
44             break
45         end
46     end
47 end
48 ret = chrom;
49 end

```

E_limit.m

```

1 function [flag] = E_limit( lenchrom, bound, ret, alps, gams)
2 %lenchrom    input:染色体长度
3 %bound      input:变量的取值范围
4 %code       input:染色体的编码值
5 %flag       output:可行性标志变量
6 flag = 0;
7 eta_cos = 0;
8 DNI_ave = 0.9686 ;
9 eta_sb = 0.95;
10 eta_ref = 0.92;
11 for i = 1 : lenchrom/4

```



```

12     eta_at = 0.99321 - 0.000176 * sqrt(ret(:,i)^2 + (ret(:,lenchrom * 3 / 4 + i) -
13         80)^2) + 0.0000000197*(ret(:,i)^2 + (ret(:,lenchrom * 3 / 4 + i) - 80)^2);
14     for j = 1 : 12
15         for k = 1 : 5
16             eta_cos = eta_cos + 25 + 25*dot([-cos(alps(j,k)) * sin(gams(j,k)), -cos
17                 (alps(j,k)) * cos(gams(j,k)), -sin(alps(j,k))],[0,0,ret(:,lenchrom * 3 / 4)-80])
18                 /(sqrt(ret(:,i)^2 + (ret(:,lenchrom * 3 / 4)-80)^2));
19         end
20     end
21     eta(i) = eta_at * eta_cos * eta_sb * eta_ref / 60 ;
22     eta_cos = 0;
23     E = 0;
24     for i = 1 : lenchrom / 4
25         E = E + eta(i) * DNI_ave *ret(:,lenchrom/4+1+2*(i-1)) * ret(:,lenchrom/4+2+2*(i
26             -1)) * 50;
27     end
28 end

```

H_limit.m

```

1 function [flag] = H_limit(ret,lenchrom)
2     count = 0;
3     for i = 1 : lenchrom / 4
4         if ret(:,2 * (i-1) + lenchrom + 1) <= 2 *ret(:,lenchrom * 3 / 4+i)
5             count = count + 1;
6         end
7     end
8     if count == 55
9         flag = 1;
10    else
11        flag = 0;
12    end
13 end

```

E_show.m

```

1 function [E] = E_show( lenchrom, bound, ret, alps, gams)
2 %lenchrom    input:染色体长度
3 %bound       input:变量的取值范围
4 %code        input:染色体的编码值
5 %flag        output:可行性标志变量
6
7 flag = 0;
8 eta_cos = 0;
9 DNI_ave = 0.9686 ;

```



```

14 bound = [bound; repmat([2,6],55,1)];
15
16 %%%%%%%%%X
17 %%
18 %%%%%%%%%
19 GApop = zeros(popsize, lenchrom);
20 fitness = zeros(popsize,1);
21 for n = 1:popsize
22     disp(n)
23     %%%%%%%%%
24     GApop(n,:) = Code(lenchrom, bound, alps, gams); %%%%%%%%%
25     %%%%%%%%%
26     fitness(n) = Fun(GApop(n,:));
27 end
28 %%%δ%%
29 [bestfitness, bestindex] = max(fitness); %%%λ%%
30 zbest = GApop(bestindex,:); %%%
31 fitnesszbest = bestfitness; %%%
32 y = zeros(1, maxgen);
33 %%
34 %%%
35 for n = 1:maxgen
36     %%% GA%%
37     GApop = Select(GApop, fitness, popsize);
38     %%%
39     GApop = Cross(pc, lenchrom, GApop, popsize, bound, alps, gams);
40     %%%
41     GApop = Mutation(pm, lenchrom, GApop, popsize, [n, maxgen], bound, alps, gams);
42     disp(n);
43     for m = 1:popsize
44         fitness(m) = Fun(GApop(m,:));
45         if fitness(m) > fitnesszbest
46             zbest = GApop(m,:);
47             fitnesszbest = fitness(m);
48         end
49     end
50     y(n) = fitnesszbest;
51 end
52 y = y(end);
53 E = E_show(lenchrom, bound, zbest, alps, gams);
54 end

```

main3.m

```

1 %%
2 %% 清空工作区
3 clear;
4 clc;

```



```

20     end
21     for m = 1:sizepop
22         pick = pick - sumf(m);
23         %%%%%%%%%%100k0000
24         if pick < 0
25             index(n) = m;
26             break;          %%%1/2,00000000000» •
27         end
28     end
29 end
30 individuals = individuals(index,:);
31 fitness = fitness(index);
32 ret = individuals;
33 end

```

Test.m

```

1 function [ flag ] = Test( lenchrom, bound, ret ,alps,gams)
2 %lenchrom    input:000峴10
3 %bound       input:±0-μ000 1x
4 %code        input:0000,000 0
5 %flag        output:¿0000000±0-
6 %300±0-
7 flag = 0;
8 %1:¿000
9 %0:2»¿000
10 R = ret(:,1 : lenchrom / 4); %0000000
11 L = ret(:,lenchrom / 4 + 1:lenchrom / 2); %¿00000
12 R = sort(R);
13 L = sort(L);
14 ret = [R,L,ret(lenchrom / 2 + 1:end)];
15 for n = 1:lenchrom
16     if ret(n) > bound(n,1) && ret(n) < bound(n,2) && E_limit(lenchrom,bound,ret,
17         alps,gams) && ret(:,1) * sin(pi/25) >= 5 && H_limit(ret,lenchrom)
18         flag = 1;
19         break
20     end
21 end

```