武漢理工大學

数学建模暑期培训论文

第 4 题

基于太阳能电池板铺设安装的非线性优化模型

第 21 组

姓名	方向
肖善	建模（组长）
易雨谦	建模（&写作）
林荣武	编程

2016 年 8 月 18 日

基于太阳能电池板铺设安装的非线性优化模型

摘要

本文针对太阳能电池板的铺设安装问题建立了非线性优化模型，并针对四个问题分别用鱼群算法和逐步法进行求解。

对于问题一，题中给出了太阳的方位角 91 度，高度角 48 度，电池板高 1.8 米，宽可任意拼接，以及屋顶的几何参数，要求设计电池板的三个安装参数：安装间距、倾角、方位角使得瞬时光照面积最大。首先，以瞬时光照面积最大为目标函数，屋顶几何参数为约束条件建立非线性规划模型，然后用鱼群算法求解。得到结果为最佳倾角 42 度，最优方位角 91 度，最优间距 2.7 米，最大面积表达式为 S 为 1.8（$W_{1\max}+W_{2\max}+W_{3\max}+W_{4\max}$ ） 经分析我们认为结果较为准确，符合参文献且符合实际。

对于问题二和问题三，给出了武汉某地经度、纬度、和海拔高度以及一天内时间点，高度角和方位角；一个月内年、月、日、时、分、高度角和方位角等离散数据等离散数据，分别要求最大日产能和最大月产能，首先，基于产能与单位面积单位时间的辐射强度、辐射面积、辐射时间的关系建立日（月）产能函数关系式，利用所给的附件的数据，先将离散的数据连续化；接着，在累计时间段上进行积分求解累计产能变化。最后，以日出日落时角等为约束条件，以日产能（月产能）最大为目标函数建立非线性优化模型，得到问题二、问题三倾角分别为 12 度和 13 度，认为结果正确且符合实际。

对于问题四，在问题三的基础上增加考虑“散射”的情况，散射光用来发电的发电量比无阴影的要减少约 10％～20％，而在实际生活一般实际工况中，太阳能电池板是以一定倾角和方位角安装在屋顶上，前排电池板不可避免的会对后排电池板造成遮蔽，因此需要考虑到覆盖问题。首先，通过对问题的几何分析，得出了在太阳直射太阳能板情况下的阴影覆盖面积的表达式，接着，沿用问题三中的辐射量和产能的表达式以及相应的日出日落约束条件，再次将时间连续化，并建立一个连续时间情况下的太阳能板产能函数，建立优化模型，最后，利用逐步算法求解到产能最大时对应的太阳能板倾角等待求参数，求解结果为倾角 30 度，最优间距为 1.1 米。

最后，本题推导的公式，对于坡向朝北的场地同样适用，仅需将朝北的坡度定义为负值，朝南的坡度定义为正值。这种设计方法突破了原有的固定阵列间距设计方法，可为任意坡角斜坡场地提供解决方案。

关键词：电池板铺设安装 非线性优化 鱼群算法 逐步法

一、问题重述

在太阳能电池板大小一定的情况下，为了尽可能多的获得太阳的照射而增加产能，安装之前，有必要根据房屋的情况对太阳能电池板的安装间距 D、倾角α和方位角β等参数进行设计。要解决如下问题

（1） 请以屋顶的几何参数和太阳能电池板各几何参数(高1.8米×宽W米,因为可以拼接，其中W可以任意)，并假设太阳高度角$h=48^0$和太阳方位角$\theta=91^0$。以此时刻点参数的具体值，请设计安装间距 D、倾角α和方位角β等参数，使之获得最大瞬时光照面积。

（2） 太阳的高度角随时间和空间变化，如果以某一天为例，以武汉某地为对象，设计安装间距 D、倾角α和方位角β等参数，使获得最大日产能。

（3） 如果在一个月的时间里考虑，设计安装间距 D、倾角α和方位角β等参数，使获得最大月产能。

（4） 如果将散射光也考虑在内，对问题（3）进行讨论。为了方便工程施工人员铺设安装太阳能电池板，请设计一个应用程序（图形用户界面），为在任一地点铺设安装提供参数支持。

二、问题分析

2.1 问题一的分析

问题一是一个优化问题，本节以瞬时光照面积最大为目标函数，屋顶几何参数为约束条件建立非线性规划模型，用鱼群算法求解。题中给出了太阳的方位角 91 度，高度角 48 度，电池板高 1.8 米，宽可任意拼接，以及屋顶的几何参数，要求设计电池板的三个安装参数：安装间距、倾角、方位角使得瞬时光照面积最大。对于无论大型地面电站还是屋顶电站，选址均希望地势平坦坡面朝南向阳，但在光伏电站实际选址和规划中，难免会遇到北坡场地或屋面是人字形的建筑物，如本题就属于人字形建筑物，4 块屋顶一半坡面朝向北方，一半坡面朝向南方且均带有一定的方位角，其中 1、3 属于斜北坡屋顶，2、4 属于斜南坡屋顶，且 1、2 属于对称屋顶，坡度等几何参数相同，同理可知 3、4。对于本题的太阳

光线方位角及高度角，斜南坡场 2、4 电池板应顺坡摆放，斜北坡场 1、3 电池板应逆坡摆放，由于上文所述对称以及同类关系，先仅以 2 号屋顶为例进行计算，经分析，为了使光照面积最大，首先要使太阳光入射方向与电池板法向量平行，即太阳能电池板的法线在水平面上的投影与正南方向的夹角等于太阳光线方位角，定义铺设间距 D 为电池板前端与下一块电池板前端的垂直距离，接着，由几何关系建立铺设间距 D 关于电池板倾角的函数表达式，而铺设行数 n 与铺设间距 D 有关，由屋顶几何约束 n、D，最终将决策变量转换为倾角α 、行数 n。最后，用鱼群算法求解非线性规划问题。本题推导的公式，对于坡向朝北的场地同样适用，仅需将朝北的坡度定义为负值，朝南的坡度定义为正值，即可得到任意朝向任意坡度的最佳倾角。

2.2 问题二、三的分析

在问题一的基础上，问题二三要求的是最大产能问题，本文将二三问合起来进行求解，第二、三问给出了武汉某地经度、纬度、和海拔高度以及一天内时间点，高度角和方位角，一个月内年、 月、 日、时、分、高度角和方位角等离散数据等离散数据，分别要求最大日产能和最大月产能。首先，由产能与辐射量、辐射面积、辐射时间这三个变量有关得到产能计算公式，基于产能公式，利用所给的附件的数据，先将离散的数据连续化；接着，一在个累计时间段上进行积分求解累计产能变化。最后，以日出日落时角等为约束条件，以日产能（月产能）最大为目标函数建立非线性优化模型。

2.3 问题四的分析

对于问题四，在问题三的基础上增加了“散射”的情况，由于在实际生活一般实际工况中，太阳能电池板是以一定倾角和方位角安装在屋顶上，前排电池板不可避免的会对后排电池板造成遮蔽，因此需要考虑到覆盖问题。首先，通过对问题的几何分析，得出了在太阳直射太阳能板情况下的阴影覆盖面积的表达式，接着，沿用问题三中的辐射量和产能的表达式以及相应的日出日落约束条件，再次将时间连续化，并建立一个连续时间情况下的太阳能板产能函数，最后，利用逐步算法求解到产能最大时对应的太阳能板倾角等待求参数。

三、模型假设

1、假设屋面平整坡度没有突变点；
2、问题一二三太阳板的面法线与太阳光线共面；
3、假设太阳以平行光照射地球；
4、同一天内不考虑地球公转影响，即同一纬度正午太阳高度角相同；
5、.忽略大气层对太阳光折射影响。

四、符号说明

符号	含义
h	太阳高度角
θ	太阳方位角
L	太阳能电池板高度
W	太阳能电池板宽度
D	太阳能电池板安装间距
α	太阳能电池板倾角
β	太阳能电池板方位角
n	屋顶放置电池板的行数
F	产能
I	单位面积单位时间辐射量
γ	太阳入射角
φ	纬度
ϕ	经度
w_0, w_e	水平方向日出、日落时角
δ	赤纬角
w	时角
η	散射光发电权重

五、模型建立与求解

5.1 问题一模型的建立与求解

5.1.1 几何关系计算

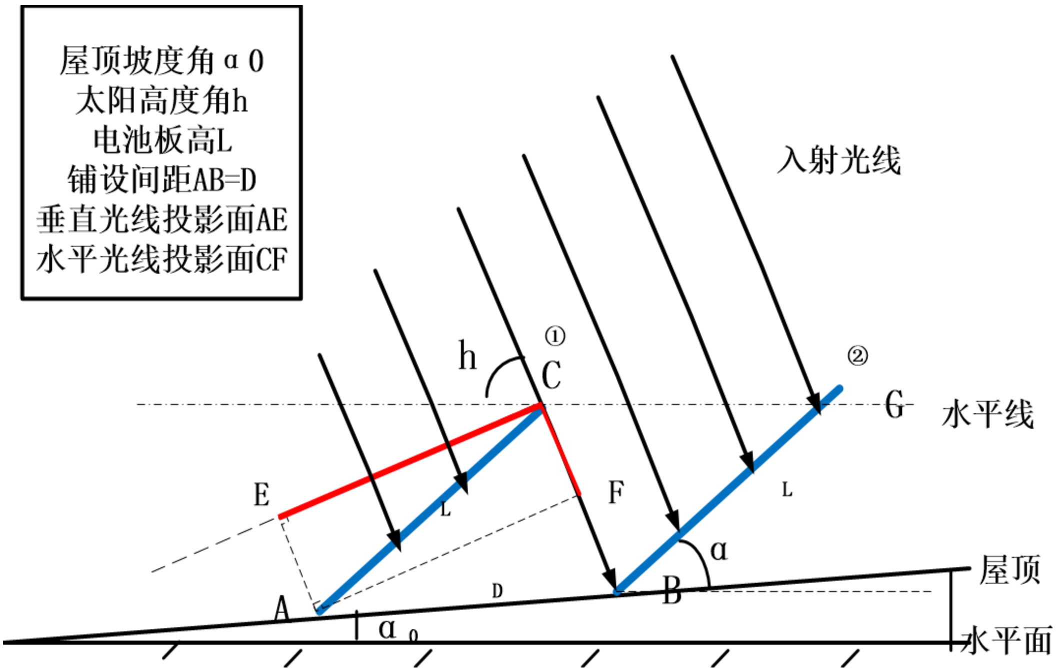

图 1 斜南坡场顺坡度摆放入射剖面图

（1）铺设距离 D 的计算

图 1 为屋顶 2 的入射剖面图，AC 为第一行电池板，BC 为第二行电池板，α 为 AC 与水平面夹角，则 $\angle CAB=\alpha-\alpha_0$，h 为入射光线与水平面夹角则 $\angle CBA=h+\alpha_0$，则在 $\triangle ABC$ 中，$\angle ACB=\pi-(\alpha+\mathrm{h})$ 定义铺设间距 D 为电池板前端与下一块电池板前端的垂直距离，由正弦定理有：

$$\frac{AB}{\sin(\angle \mathrm{ACB})}=\frac{AC}{\sin(\angle CBA)} \text{ 即: } \frac{D}{\sin(\alpha+h)}=\frac{L}{\sin(h+\alpha_0)}$$，由此得到 D 的表达式：

$$D=\frac{L\times\sin(\alpha+\mathrm{h})}{\sin(\alpha_0+\mathrm{h})}$$

（2）有效投影面积的计算

将 AC 面分别在垂直于光线和平行于光线方向进行分解投影，定义 AC 面垂直于光线的投影面 CE 为有效光照面积，$\angle ACB=\pi-(\alpha+\mathrm{h})$，则有效面积 s 表达式，：

$$s = Lw\sin(\alpha +$$

对于 n 行电池板总的光照面积为：$S = \sum_{i=1}^{n} Lw_i \sin(\alpha + \mathrm{h})$

(3)屋顶对角线约束计算

由图 2 屋顶的俯视图可以看到屋顶对角线与正南方向的夹角为 u-v,而电池板法线在屋顶的投影与正南方向的夹角为β ，则对角线与电池板法线投影线的夹角为$\pi - (\mathrm{u} - \mathrm{v} + \beta)$ ，其中$\sin u = \frac{a}{\sqrt{a^2 + b^2}}$ ，$\cos u = \frac{b}{\sqrt{a^2 + b^2}}$ ，$v = 42^0$，$\beta = \theta$

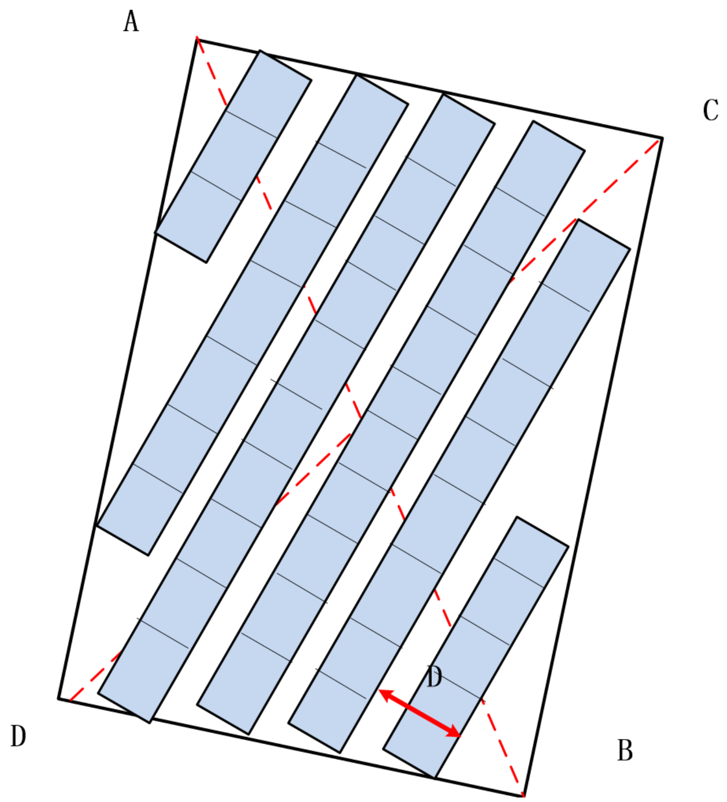

图 2 屋顶俯视图

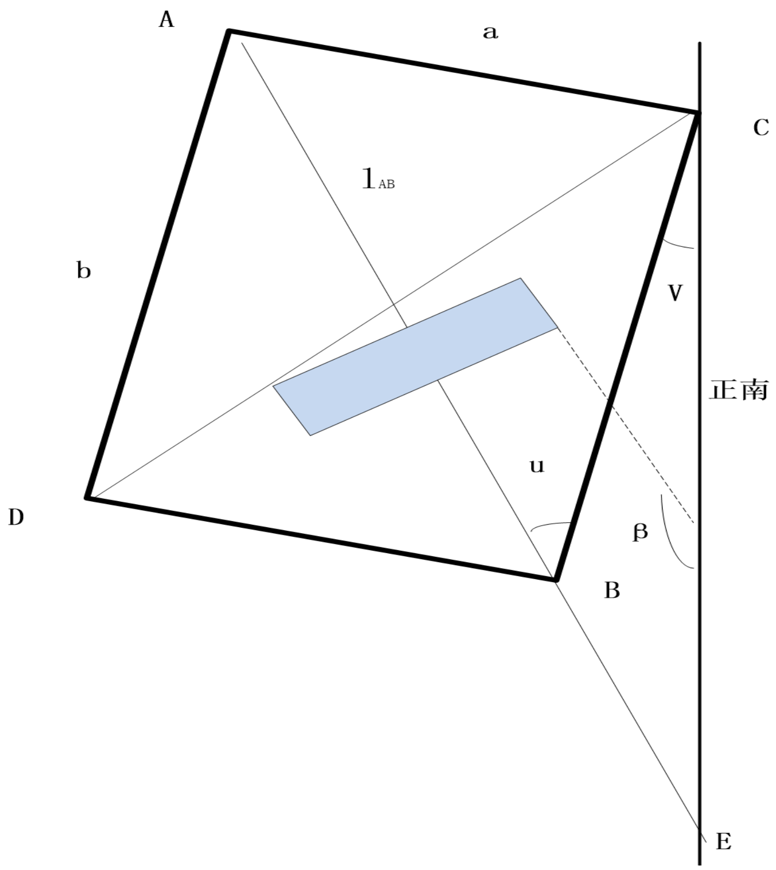

图 3 屋顶俯视图角度关系

5.1.2 模型的建立

考虑到光照面积最大，首先要使太阳光入射方向与电池板法向量平行，即太阳能电池板的法线在水平面上的投影与正南方向的夹角等于太阳光线方位角 $\beta=\theta$，定义铺设间距 D 为电池板前端与下一块电池板前端的垂直距离，由屋顶几何约束 n、D，最终将决策变量转换为倾角 α 、行数 n。

以总有效光照面积最大为目标函数，屋顶对角线尺寸为约束条件，得到非线性优化模型如下：

$$\max nL\sin(\alpha+\mathrm{h})$$

$$s.t\begin{cases} D=\dfrac{L\times\sin(\alpha+\mathrm{h})}{\sin(\alpha_0+\mathrm{h})} \\ nD\le l_{AB}\cos(\pi-(\mathrm{u}-\mathrm{v}+\beta)\le(\mathrm{n}+1)\mathrm{D} \\ \beta=\theta \\ \sin u=\dfrac{a}{\sqrt{a^2+b^2}} \\ \cos u=\dfrac{b}{\sqrt{a^2+b^2}} \end{cases}$$

由于宽度可以任意拼接，因此每当倾角α 、间距 D 确定，电池板摆放角度确定，总能确定一个最大宽度$w_i=W_i\max$ 与之对应，因此在求解目标函数 S 时可以不考虑宽度，此时$S=\sum_{i=1}^{n}Lw_i\sin(\alpha+\mathrm{h})$,则$Y_s=nL\sin(\alpha+\mathrm{h})$

5.1.3 模型求解

以α 、n 为决策变量用鱼群算法对目标函数进行求解，算法流程如下：

Step1.初始化种群，包括种群大小、移动步长、感知范围、拥挤度因子、尝试次数及种群的最大迭代次数等参数。

Step2.根据目标函数计算每条人工鱼的适应值，找出并记录最大适应度值对应的人工鱼，即最优解。

Step3.对当前鱼分别尝试执行聚群算子和追尾算子，执行适应度改善较大的算子；否则则执行觅食算子。

Step4.更新最优解。

Step5.判断是否满足结束条件，若满足，输出最优解；若不满足，返回 step3，继续循环。

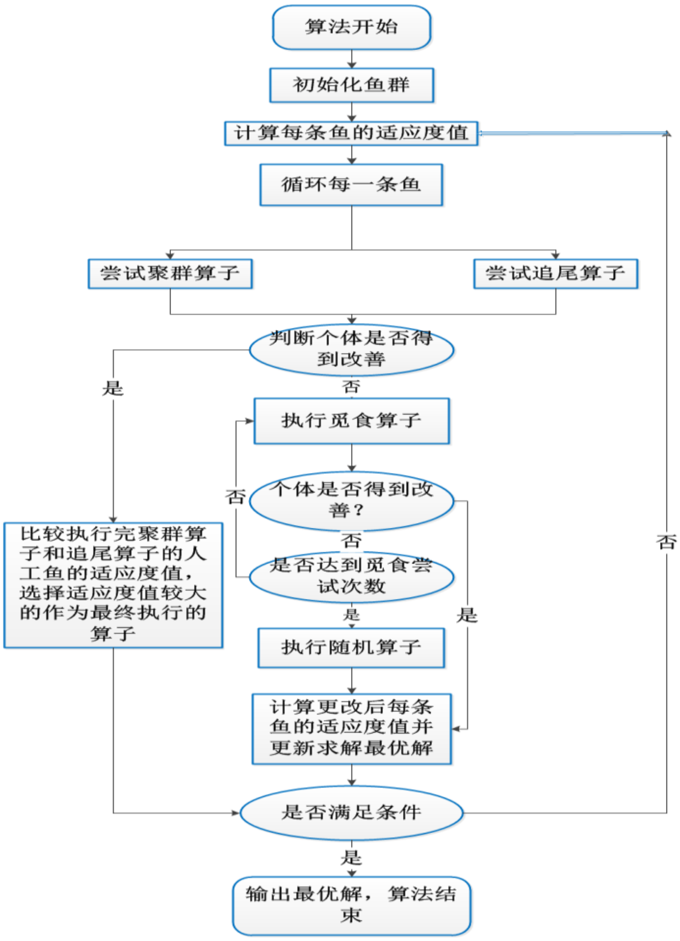

图 4　鱼群算法流程

5.1.5 结果分析

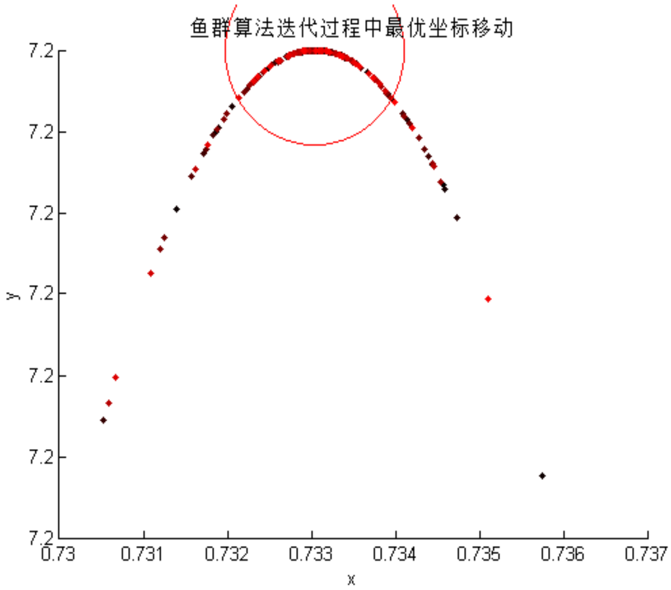

图 5 鱼群算法最优解结果

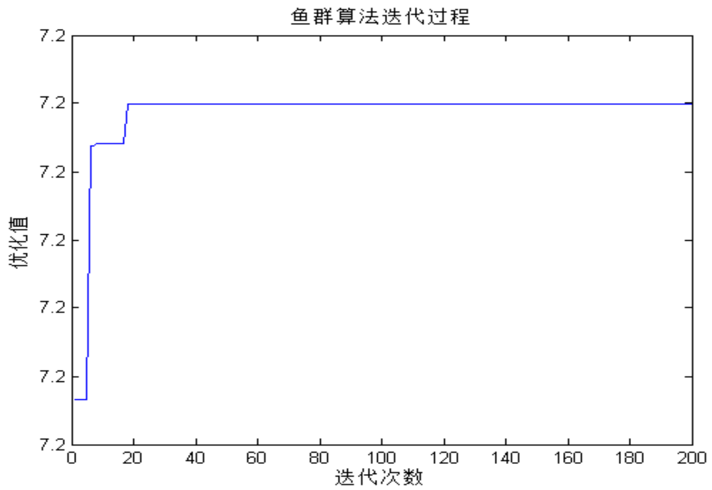

图 6 鱼群算法迭代次数

求得最佳倾角弧度为 0.733，$\alpha=42^0$，n=4,D=2.7 ，$Y_s=10.738$ ，则此时最大面积 S 为 1.8（$W_{1\max}+W_{2\max}+W_{3\max}+W_{4\max}$ ）

表 1 第一问结果

α	n	D	Y_s	S
42 度	4	2.7m	10.738	1.8（$W_{1\max}+W_{2\max}+W_{3\max}+W_{4\max}$ ）

规划图如下：

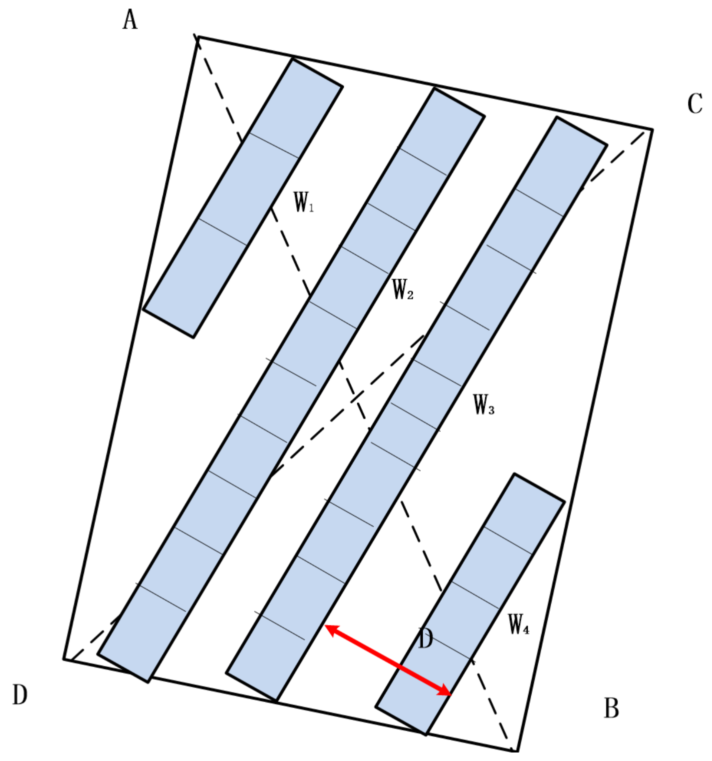

图 6 规划结果图

5.2 问题二、三模型的建立与求解

5.2.1 建模准备

（1） 产能表达式的确定

当太阳能板在朝正南方向的时候能够保证所能接收到的太阳辐射的量及强度是最大的，所以认为太阳能板是面向南边，而且其面的发现与太阳光线是共面的，即 “正投”。通过量纲运算规则，推导产能与辐射量，辐射面积及时间的表达式[3]如下：

$$F=\int_{t_1}^{t_2} ISdt$$

这里 F 表示产能（J），I 表示辐射量（$\frac{J}{m^2s}$），S 表示辐射面积，dt 表示时间微元，t_1,t_2 为辐射计时的始末时间。即产能可以表示为单位面积单位时间辐射量在面积和时间上的积分，但基于本题太阳能板面积是一个定值，所以 S 相对于 dt 来说是一个常数。、

（2） 辐射量表达式的确定

辐射量[4]是指太阳照射到太阳能板上太阳能板所接受的量，与太阳能板所处的地理位置（经纬度），海拔高度和太阳的高度角方位角有关，为了说明辐射量的具体表达式含义，再给出两个地理学上的概念，即时角和赤纬。

日地中心连线与赤道的夹角称为赤纬角，赤道以北为正，南为负，变化范围为-23.5 至 23.5（度），可由 Copper 近似公式[5]得到

$$\delta=23.45\sin(\frac{2\pi(284+\mathrm{n})}{365})$$

其中 n 为一年中的第几天。

时角是以正午 12 点为 0 度开始计算，每一小时为 15 度，上午为负下午为正，即：

$$\omega=15((t_i+\frac{\phi-120}{15})-12)$$

其中 t_i 表示当前时刻，ϕ 为地点的经度。

进一步确定出辐射量的表达式如下：

$$I=I_0\cos\gamma$$

其中：$$\begin{cases}\cos\gamma=\sin(\varphi-\alpha)\sin\delta+\cos(\varphi-\alpha)\cos\delta\cos w_e \\ w_e=\arccos(-\tan(\varphi-\alpha)\tan\delta) \\ \sin\theta=\cos\delta\dfrac{\sin\varphi}{\cosh} \\ I_0=\dfrac{24}{\pi}\times1353(1+0.33\cos\dfrac{360n}{365})(\cos\varphi\cos\delta\sin w_0+\dfrac{\pi}{180}\omega_0\sin\varphi\sin\delta) \\ \omega_0=\arccos(-\tan\varphi\tan\delta)\end{cases}$$

γ表示太阳入射角，h 为太阳高度角，φ为纬度，α为太阳能板的倾角，w_0, w_e表示水平方向的日出日落的时角。

5.2.2 模型的建立：

问题二三都是给出一系列的关于不同时间下太阳高度角及方向角的数据，还有此点的地理位置即纬度和经度，通过假设和简化设置太阳能板方向角与太阳方向角一致，以倾角范围 0 度至 90 度，步长设置为$\Delta\alpha$可调节，采用逐步搜索算法，以太阳日出日落作为约束条件，对每时刻太阳辐射量累加，以每日时段或者一个月时段为标准，对在太阳高度角和太阳方向角h,θ下的辐射量比较，使辐射量最大。同时我们认为这种照射是无散射无覆盖的情况的，即辐射面积是定值，由此我们将求解产能最大转化为求解累计辐射量最大，所建立的优化模型如下：

$$\max\int_{t_1}^{t_2} I dt$$

$$\begin{cases}T_{日出}=12(1-\dfrac{\omega_e}{15}) \\ T_{日落}=12(1+\dfrac{\omega_e}{15})\end{cases}$$

由于我们不考虑覆盖的情况，最终求得的太阳能板的最佳倾角为

5.2.3 模型的求解：

采用逐步法[7]求解：我们将采用逐步算法计算最佳倾斜角，基本思想是以倾斜角 β 为自变量，倾斜角的范围为 0~90°，步长为 1°． 以日出日落的日照时间作为约束条件，对于每小时的太阳辐射强度进行累加 ．最终以每日作为标准，对 90 个倾斜角的阳辐射强度进行比较，辐射强度最大的对应为最佳倾斜角 ．在全晴天的基础上，建立倾斜放置的光伏板表面太阳辐射能数学模型

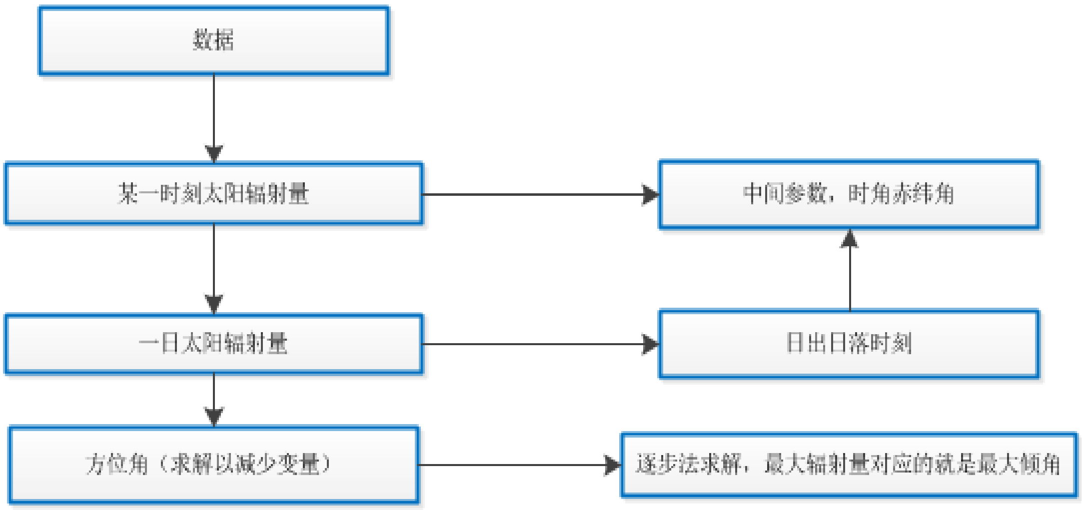

图 7 逐步算法流程图

5.2.4 结果分析

问题二求得太阳能电池板倾角 12 度，问题三求得倾角 18 度：

表二 问题二结果显示

问题序号	板倾斜度
问题二	12

表三 问题三结果显示

问题序号	板倾斜度
问题三	18

5.3 问题四

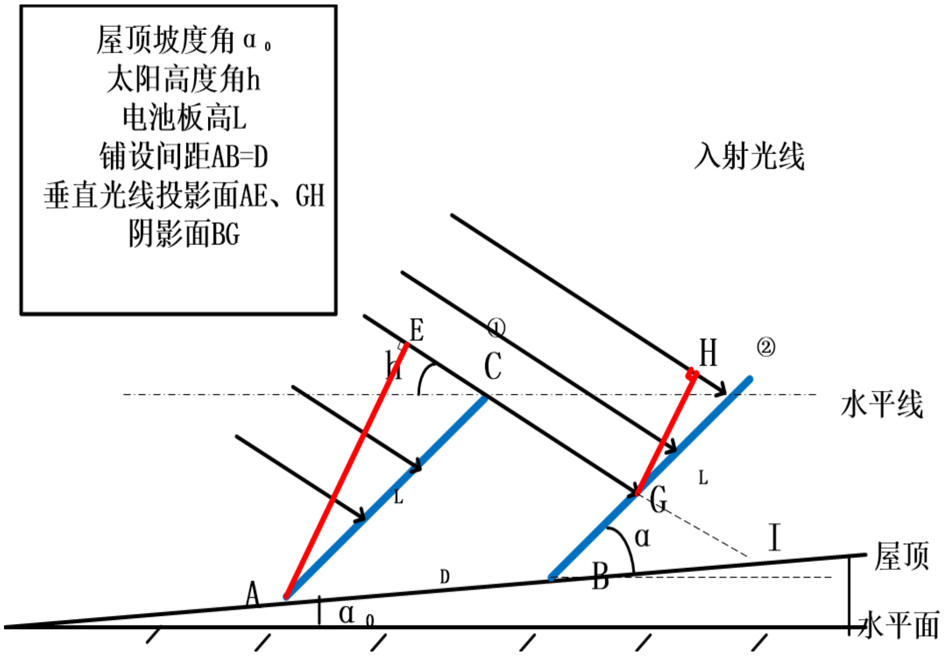

图 8 考虑散射有遮盖剖面图

5.3.1 阴影面积计算

如图 8 所示，AC 为第一行电池板，BC 为第二行电池板，则 BG 为遮盖面积也就是散射光部分，下面计算遮盖面 BG：

延长 EG 交屋顶面 AB 于 I,由相似关系：

$$\frac{l_{阴}}{L}=\frac{BG}{AC}=\frac{BI}{AI}$$

又由：

$$\begin{cases} AE = L\cos(\alpha+\mathrm{h}) \\ \mathrm{A}\,I = \dfrac{AE}{\sin(\alpha_0+\mathrm{h})} \\ BI = AI - AB \end{cases}$$

解得：

$$l_{阴}=\mathrm{L}\times(1-\frac{D\sin(\alpha_0+\mathrm{h})}{L\cos(\alpha+\mathrm{h})})$$

5.3.2 模型的建立

问题四要考虑到有散射的状况，散射光用来发电的发电量比无阴影的要减少约 10％～20％,则散射权重η=0.8 散射的原因是在实际生活中太阳能板是排成行，前排的电池板会对后排电池板造成遮蔽，遮蔽部分通过散射产能，这就需要计算出有遮蔽阴影时的面积，而这个面积和一系列角度参数有关的，这就又回到了第一问中的覆盖面积中，考虑到散射，结合问题一中的约束条件建立完整的模型如下：

$$\max \int_{t_1}^{t_2} IS_{照}dt+\eta\int_{t_1}^{t_2} IS_{阴}dt$$

$$\begin{cases} T_{日出}=12(1-\dfrac{\omega_e}{15}) \\ T_{日落}=12(1+\dfrac{\omega_e}{15}) \\ S_{阴}=\omega l(1-\dfrac{D\sin(\alpha_0+\mathrm{h})}{l\cos(\alpha+\mathrm{h})})\sin\alpha \\ S_{全}=\omega l\sin\alpha \\ S_{照}=S_{全}-S_{阴} \end{cases}$$

5.3.3 模型求解与结果

求得最优间距 D 为 1.1 米，电池板倾角 30 度

表四 问题四结果

问题序号	间距（米）	板角倾斜
问题四	1.1	30

5.4 问题五 GUI 界面设计

在 matlab 中编写 GUI 界面如下：

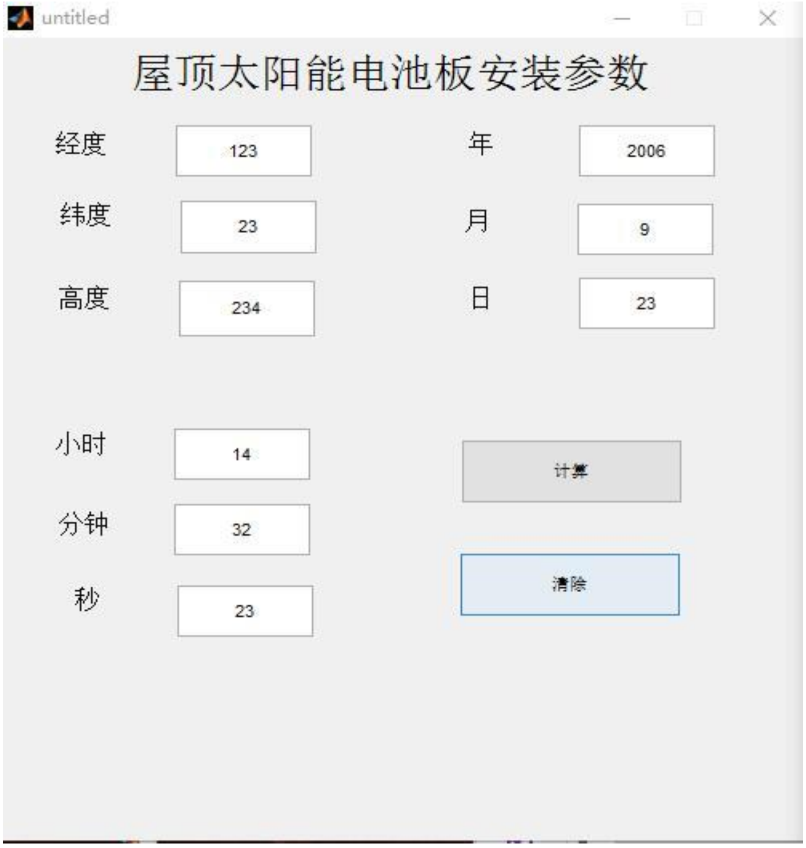

图 9 GUI 输入界面

图 10 GUI 输出界面

六、 模型的评价和改进

6.1.模型优点：

1. 问题一通过建立了一个无覆盖的约束优化模型，达到了模型的简化效果，并选用了鱼群算法，得到了较为精确的结果。

2. 问题二三将求解最大日产能问题转化为求解最大辐射量及最大辐射面积，并且将离散的时间数据连续化，并且采用了逐步算法，达到了较为精确可靠的结果。

6.2.模型缺点：

1. 逐步算法的步距影响到了结果的准确同时也决定了时间的开销，故在数据规模过大时，算法时间效率较低。

2. 模型未将反射辐射量和其他实际因素考虑进去，对于实际的情况，模型存在一定的误差。

6.3 模型的改进：

1.对于鱼群算法，可以考虑与其他智能算法相结合，混合优化，比如粒子群算法，加强算法收敛性，或考虑视野的改进，使用自适应步长的方式改进。

2 对于模型，还可以将天气情况与经济效率一起考虑，使模型更能贴近现实。

6.4 模型推广：

本题推导的公式，对于坡向朝北的场地同样适用，仅需将朝北的坡度定义为负值，朝南的坡度定义为正值，代入公式，即可得到任意朝向任意坡度的最佳倾角等铺排方式。

参考文献

[1]赵心旭,丁少婷,王义康.太阳能小屋光伏电池铺设倾角动态优化与分析[J].装备制造与教育,2014,02:36-39.
[2]赵国喜,代林帅.太阳能小屋的最优设计[J].新乡学院学报(自然科学版),2013,05:327-328+332.
[3]潘艳秋,易德富,钱丽梅,赵波.太阳能小屋设计中光伏电池的最优铺设[J].云南民族大学学报(自然科学版),2013,S1:31-35.
[4]刘月华,吕永标,雷春霞,王义康.基于贪婪算法的太阳能小屋光伏电池组合分析[J].电子科技,2013,04:102-105+109.
[5]李云,章毅,黄阳.基于多目标混合整数规划模型的光伏发电系统优化设计[J].硅谷,2013,16:19-22.
[6]周克元,周后福.太阳能小屋设计的数学模型[J].太原师范学院学报(自然科学版),2013,03:104-108.
[7]赵心旭,丁少婷,王义康.光伏电池铺设倾角的动态优化与分析[J].电子科技,2014,12:96-99.
[8]丁明,刘盛,徐志成. 光伏阵列改进优化设计方法与应用[J]. 中国电机工程学报,2013,34:2-9.
[9]申政,吕建,杨洪兴,蒋英. 太阳辐射接受面最佳倾角的计算与分析[J]. 天津城市建设学院学报,2009,01:61-64+75.
[10]董庆来,任治豪,曾晓楠,梅哲. 太阳能光伏电池板铺设方案的优化设计与研究[J]. 河南科学,2012,12:1788-1791.
[11]李杰慧,刘祖明,李景天. 固定式独立光伏系统中光伏方阵设计[A]. 中国太阳能学会.21 世纪太阳能新技术——2003 年中国太阳能学会学术年会论文集[C].中国太阳能学会:,2003:3.
[12]周长友,杨智勇,杨胜铭. 北坡场地光伏电站阵列间距设计[J]. 华电技术,2013,06:14-17+85.
[13]杨茜. 主动式太阳能技术与办公建筑设计一体化研究[D].重庆大学,2010.
[14]杨金焕,毛家俊,陈中华. 不同方位倾斜面上太阳辐射量及最佳倾角的计算[J]. 上海交通大学学报,2002,07:1032-1036.
[15]魏生贤,李明,季旭,林文贤,郑土逢,李国良,罗熙. 太阳能平板集热器件纵横比与板间距优化[J]. 农业工程学报,2010,11:225-230.

附录

第一问matlab代码：

```
    clc
clear  all
close  all
tic
figure(1);hold  on
%ezplot('x*sin(10*pi*x)+2',[-1,3]);
%%  参数设置
fishnum=100;  %生成 50 只人工鱼
MAXGEN=200;  %最多迭代次数
try_number=100;%最多试探次数
visual=2;  %感知距离
delta=0.618;  %拥挤度因子
step=0.1;  %步长
%%  初始化鱼群
%lb_ub=[0.0997,pi/2,1];
lb_ub=[0.6,0.8,1];
X=AF_init(fishnum,lb_ub);
LBUB=[];
for  i=1:size(lb_ub,1)
        LBUB=[LBUB;repmat(lb_ub(i,1:2),lb_ub(i,3),1)];
end
gen=1;
BestY=-1*ones(1,MAXGEN);  %每步中最优的函数值
BestX=-1*ones(1,MAXGEN);  %每步中最优的自变量
besty=-100;  %最优函数值
Y=AF_foodconsistence(X);
%plot(X,Y);
while  gen<=MAXGEN
        fprintf(1,'%d\n',gen);
        for  i=1:fishnum
                    %%  聚群行为
                [Xi1,Yi1]=AF_swarm(X,i,visual,step,delta,try_number,L
BUB,Y);
                  %%  追尾行为
                [Xi2,Yi2]=AF_follow(X,i,visual,step,delta,try_number,
LBUB,Y);
                if  Yi1>Yi2
                        X(:,i)=Xi1;
                        Y(1,i)=Yi1;
                else
```

```
                        X(:,i)=Xi2;
                        Y(1,i)=Yi2;
                end
        end
        [Ymax,index]=max(Y);
        figure(1);
        plot(X(1,index),Ymax,'.','color',[gen/MAXGEN,0,0])
        if  Ymax>besty
                besty=Ymax;
                bestx=X(:,index);
                BestY(gen)=Ymax;
                [BestX(:,gen)]=X(:,index);
        else
                BestY(gen)=BestY(gen-1);
                [BestX(:,gen)]=BestX(:,gen-1);
        end
        gen=gen+1;
end
plot(bestx(1),besty,'ro','MarkerSize',100)
xlabel('x')
ylabel('y')
title('鱼群算法迭代过程中最优坐标移动')

%%  优化过程图
figure
plot(1:MAXGEN,BestY)
xlabel('迭代次数')
ylabel('优化值')
title('鱼群算法迭代过程')
disp(['最优解 X：',num2str(bestx,'%1.4f')])
disp(['最优解 Y：',num2str(besty,'%1.4f')])
bestx*180/pi
toc

function [Y]=AF_foodconsistence(X)
cosy=53.2/56.16;
siny=18/56.16;
l=1.8;%长度
lD=56.1888;%对角线长度
beta=91*(pi/180);%太阳方位角
h=(pi/180)*48;%太阳高度角
a0=(pi/180)*5.71;%坡度角
a1=(pi/180)*42;%屋顶偏角
fishnum=size(X,2);
```

```
n_=0;
T=1.8/18;
for i=1:fishnum
    %D=(l*sin(X(i))+T*l*cos(X(i))*cos(beta-a1)/tan(h))/(1-T*cos(beta-a1)/tan(h))+l*cos(X(i));
    D=sin(pi-(X(i)+h))*l/sin(h-a0);
   % D=sin((X(i)+h))*l/sin(a0+h);

m=lD*cos(a1)*(cosy*cos(beta)+siny*sin(beta))-sin(a1)*(siny*cos(beta)-cosy*sin(beta));
    for n=1:200
        if  (n*D<=m)&&((n+1)*D>m)
            n_=n;
            break
        end
    end
     Y(1,i)=n_*l*sin(X(i)+h);
End

function D=AF_dist(Xi,X)
% 计算第 i 条鱼与所有鱼的位置，包括本身。
% 输入：
% Xi   第 i 条鱼的当前位置
% X    所有鱼的当前位置
% 输出：
% D    第 i 条鱼与所有鱼的距离
col=size(X,2);
D=zeros(1,col);
for j=1:col
    D(j)=norm(Xi-X(:,j));
End

    function  [Xnext,Ynext]=AF_follow(X,i,visual,step,deta,try_number,LBUB,lastY)
%  追尾行为
%  输入：
%  X                        所有人工鱼的位置
%  i                        当前人工鱼的序号
%  visual              感知范围
%  step                  最大移动步长
%  deta                  拥挤度
%  try_number     最大尝试次数
%  LBUB                  各个数的上下限
%  lastY               上次的各人工鱼位置的食物浓度
```

```
%   输出:
%   Xnext               Xi 人工鱼的下一个位置
%   Ynext               Xi 人工鱼的下一个位置的食物浓度
Xi=X(:,i);
D=AF_dist(Xi,X);
index=find(D>0  &  D<visual);
nf=length(index);
if  nf>0
        XX=X(:,index);
        YY=lastY(index);
        [Ymax,Max_index]=max(YY);
        Xmax=XX(:,Max_index);
        Yi=lastY(i);
        if  Ymax/nf>deta*Yi;
                Xnext=Xi+rand*step*(Xmax-Xi) rm(Xmax-Xi);
                for  i=1:length(Xnext)
                        if    Xnext(i)>LBUB(i,2)
                          Xnext(i)=LBUB(i,2);
                        end
                        if    Xnext(i)<LBUB(i,1)
                          Xnext(i)=LBUB(i,1);
                        end
                end
                Ynext=AF_foodconsistence(Xnext);
        else
                [Xnext,Ynext]=AF_prey(X(:,i),i,visual,step,try_number
,LBUB,lastY);
        end
else
        [Xnext,Ynext]=AF_prey(X(:,i),i,visual,step,try_number,LBUB,la
stY);
end

function X=AF_init(Nfish,lb_ub)
% 输入:
% Nfish 鱼群大小
% lb_ub 鱼的活动范围

% 输出:
% X     产生的初始人工鱼群

% example:
% Nfish=3;
```

```
% lb_ub=[-3.0,12.1,1;4.1,5.8,1];
%% 这里的 lb_ub 是 2 行 3 列的矩阵，每行中前两个数是范围的上下限，第 3 个数是在该范围内的数的个数
% X=Inital(Nfish,lb_ub)
%% 就是产生[-3.0,12.1]内的数 1 个，[4.1,5.8]内的数 1 个
%% 两个数一组，这样的数一共 Nfish 个
row=size(lb_ub,1);
X=[];
for i=1:row
    lb=lb_ub(i,1);
    ub=lb_ub(i,2);
    nr=lb_ub(i,3);
    for j=1:nr
        X(end+1,:)=lb+(ub-lb)*rand(1,Nfish);
    end
end

    function  [Xnext,Ynext]=AF_prey(Xi,ii,visual,step,try_number,LBUB,lastY)
%  觅食行为
%  输入：
%  Xi                    当前人工鱼的位置
%  ii                    当前人工鱼的序号
%  visual              感知范围
%  step                  最大移动步长
%  try_number     最大尝试次数
%  LBUB                  各个数的上下限
%  lastY               上次的各人工鱼位置的食物浓度

%  输出：
%  Xnext               Xi 人工鱼的下一个位置
%  Ynext               Xi 人工鱼的下一个位置的食物浓度

Xnext=[];
Yi=lastY(ii);
for  i=1:try_number
        Xj=Xi+(2*rand(length(Xi),1)-1)*visual;
        Yj=AF_foodconsistence(Xj);
        if  Yi<Yj
                Xnext=Xi+rand*step*(Xj-Xi) rm(Xj-Xi);
                for  i=1:length(Xnext)
                        if    Xnext(i)>LBUB(i,2)
                                Xnext(i)=LBUB(i,2);
                        end
```

```
                        if    Xnext(i)<LBUB(i,1)
                                Xnext(i)=LBUB(i,1);
                        end
                end
                Xi=Xnext;
                break;
        end
end

%  随机行为
if  isempty(Xnext)
        Xj=Xi+(2*rand(length(Xi),1)-1)*visual;
        Xnext=Xj;
        for  i=1:length(Xnext)
                if    Xnext(i)>LBUB(i,2)
                        Xnext(i)=LBUB(i,2);
                end
                if    Xnext(i)<LBUB(i,1)
                        Xnext(i)=LBUB(i,1);
                end
        end
end
Ynext=AF_foodconsistence(Xnext);

    function  [Xnext,Ynext]=AF_swarm(X,i,visual,step,deta,try_number,
LBUB,lastY)
%  聚群行为
%  输入:
%  X                       所有人工鱼的位置
%  i                       当前人工鱼的序号
%  visual             感知范围
%  step                  最大移动步长
%  deta                  拥挤度
%  try_number     最大尝试次数
%  LBUB                  各个数的上下限
%  lastY                 上次的各人工鱼位置的食物浓度

%  输出:
%  Xnext                 Xi 人工鱼的下一个位置
%  Ynext                 Xi 人工鱼的下一个位置的食物浓度
Xi=X(:,i);
D=AF_dist(Xi,X);
index=find(D>0  &  D<visual);
nf=length(index);
```

```
if  nf>0
        for  j=1:size(X,1)
                Xc(j,1)=mean(X(j,index));
        end
        Yc=AF_foodconsistence(Xc);
        Yi=lastY(i);
        if  Yc/nf>deta*Yi
                Xnext=Xi+rand*step*(Xc-Xi) rm(Xc-Xi);
                for  i=1:length(Xnext)
                        if    Xnext(i)>LBUB(i,2)
                          Xnext(i)=LBUB(i,2);
                        end
                        if    Xnext(i)<LBUB(i,1)
                          Xnext(i)=LBUB(i,1);
                        end
                end
                Ynext=AF_foodconsistence(Xnext);
        else
                [Xnext,Ynext]=AF_prey(Xi,i,visual,step,try_number,LBU
B,lastY);
        end
else
        [Xnext,Ynext]=AF_prey(Xi,i,visual,step,try_number,LBUB,lastY)
;
end
```

问题二代码：

```
    I=zeros(91);
    for a=0:90
        a1=a+1;%倾角
        a=a*pi/180;%转换成弧度
        for b=0:90%方位角
            b1=b+1;
            b=b*pi/180;%转化成弧度
            sinbb=sin(a);
            cosyb=cos(b);
            cosbb=cos(a);
            sinyb=sin(b);
            for i=1:14001
                cosas=cos(data1(i,2)*pi/180);
                cosys=cos(data1(i,3)*pi/180);
                sinys=sin(data1(i,3)*pi/180);
                sinas=sin(data1(i,2)*pi/180);

cosy=sum([-cosas*cosys,-cosas*sinys,-sinas].*[-sinbb*cosyb,-sinbb*sinyb,-cosbb])/n
```

```
orm([-cosas*cosys,-cosas*sinys,-sinas])*norm([-sinbb*cosyb,-sinbb*sinyb,-cosbb]);
                t=I(a1,b1)+cosy;
                I(a1,b1)=t;
            end
        end
    end
    [ao,bo]=find(I==max(max(I))%输出最大倾角和方位角
```
问题三代码：

```
    load('C:\Users\lrw\Desktop\data.mat')
     Isc=1353;
     wei=30.5286*pi/180;
     dong=114.3556;
     s=(23.45*sin(2*pi*(284+n)/365))*pi/180;%赤纬角
    for a=0:90
        m=a+1;
        a=a*pi/180;
        I(m)=0;
        we=acos(-tan(wei-a)*tan(s));%带斜面
        wo=acos(-tan(wei)*tan(s));%水平面
        we=min(we,wo);

I0=24/pi*Isc*(1+0.033)*cos((360*n/365))*(cos(wei)*cos(s)*sin(wo)+pi*wo*sin(wei)
*sin(s)/180);
        Tstart=12*(1-wo/15);
        Tend=12*(1+we/15);
        for i=1:14001
            if data1(i,1)>=Tstart&&data1(i,1)<=Tend
                b=data1(i,3);
                h=data1(i,2);
                t=data1(i,1)+4*(120-dong);%太阳时
                w=(15*pi/180)*(t-12);%时角
                y=asin(cos(s)*sin(wei)/cos(h));

cosy=(sin(wei)*cos(b)-cos(wei)*cos(y)*sin(a))+(cos(wei)*cos(a)+sin(wei)*cos(y)*si
n(a))*cos(s)*cos(w)+sin(y)*sin(a)*sin(w);
                I(m)=I(m)+I0*cosy;
            end
        end
    end
```
问题四代码：

```
    load('C:\Users\lrw\Desktop\data.mat')
     Isc=1353;
     wei=30.5286*pi/180;
     dong=114.3556;
```

```
    s=(23.45*sin(2*pi*(284+n)/365))*pi/180;%赤纬角
    for a=0:90
        m=a+1;
        a=a*pi/180;
        I(m)=0;
        we=acos(-tan(wei-a)*tan(s));%带斜面
        wo=acos(-tan(wei)*tan(s));%水平面
        we=min(we,wo);

I0=24/pi*Isc*(1+0.033)*cos((360*n/365))*(cos(wei)*cos(s)*sin(wo)+pi*wo*sin(wei)
*sin(s)/180);
        Tstart=12*(1-wo/15);
        Tend=12*(1+we/15);
        for i=1:14001
            if data1(i,1)>=Tstart&&data1(i,1)<=Tend
                b=data1(i,3);
                h=data1(i,2);
                t=data1(i,1)+4*(120-dong);%太阳时
                w=(15*pi/180)*(t-12);%时角
                y=asin(cos(s)*sin(wei)/cos(h));

cosy=(sin(wei)*cos(b)-cos(wei)*cos(y)*sin(a))+(cos(wei)*cos(a)+sin(wei)*cos(y)*si
n(a))*cos(s)*cos(w)+sin(y)*sin(a)*sin(w);
                I(m)=I(m)+I0*cosy*(sin(b)-D*sin(data3(i,3))/cos(b+h)*sin(b);
            end
        end
    end
```

GUI 代码：

```
    unction varargout = untitled(varargin)
    % UNTITLED MATLAB code for untitled.fig
    %       UNTITLED, by itself, creates a new UNTITLED or raises the existing
    %       singleton*.
    %
    %       H = UNTITLED returns the handle to a new UNTITLED or the handle
to
    %       the existing singleton*.
    %
    %       UNTITLED('CALLBACK',hObject,eventData,handles,...) calls the
local
    %       function named CALLBACK in UNTITLED.M with the given input
arguments.
    %
    %       UNTITLED('Property','Value',...) creates a new UNTITLED or raises
the
```

```
%       existing singleton*.   Starting from the left, property value pairs are
%       applied to the GUI before untitled_OpeningFcn gets called.   An
%         unrecognized property name or invalid value makes property
application
%       stop.   All inputs are passed to untitled_OpeningFcn via varargin.
%
%       *See GUI Options on GUIDE's Tools menu.   Choose "GUI allows
only one
%       instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help untitled

% Last Modified by GUIDE v2.5 18-Aug-2016 21:48:05

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @untitled_OpeningFcn, ...
                   'gui_OutputFcn',  @untitled_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before untitled is made visible.
function untitled_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to untitled (see VARARGIN)
```

```
% Choose default command line output for untitled
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes untitled wait for user response (see UIRESUME)
% uiwait(handles.figure1);
%set(handles.axes1,'visible','off')

% --- Outputs from this function are returned to the command line.
function varargout = untitled_OutputFcn(hObject, eventdata, handles)
% varargout   cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function edit2_Callback(hObject, eventdata, handles)
% hObject     handle to edit2 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double


% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject     handle to edit2 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%        str2double(get(hObject,'String')) returns contents of edit9 as a double


% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%        str2double(get(hObject,'String')) returns contents of edit10 as a
double


% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
%        str2double(get(hObject,'String')) returns contents of edit11 as a double


% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object creation, after setting all properties.
%function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes1
```

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%set(handles.axes1,'visible','on')

location.longitude=str2num(get(handles.edit2,'String'));%(经度)
location.latitude=str2num(get(handles.edit3,'String'));%(纬度)
location.altitude=str2num(get(handles.edit4,'String'));%(海拔)
  time.year =str2num(get(handles.edit12,'String'));
  time.month =str2num(get(handles.edit13,'String'));
  time.day = str2num(get(handles.edit14,'String'));
  time.hour = str2num(get(handles.edit15,'String'));
  time.min = str2num(get(handles.edit16,'String'));
  time.sec = str2num(get(handles.edit17,'String'));
  time.UTC=1;
sun = sun_position(time, location);

openfig('result.fig');
h=guihandles;
set(h.edit3,'String',sun.zenith );
set(h.edit4,'String',sun.azimuth);
set(h.edit1,'String',40*rand());
set(h.edit2,'String',40*rand());

axes(h.axes1);
filename='直射图.jpg';
pathname='C:\Users\lrw\Desktop\';
fpath=[pathname filename];%将文件名和目录名组合成一个完整的路径。
imshow(imread(fpath));%用 imread 读入图片，并用 imshow 在 axes_src 上显示。


% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
```

```
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listbox1 contents as cell
array
%         contents{get(hObject,'Value')} returns selected item from listbox1


% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to listbox1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if    ispc    &&    isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit12_Callback(hObject, eventdata, handles)
% hObject      handle to edit12 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
%         str2double(get(hObject,'String')) returns contents of edit12 as a
double


% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit12 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if    ispc    &&    isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end



function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
%        str2double(get(hObject,'String')) returns contents of edit13 as a double


% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit14_Callback(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit14 as text
%        str2double(get(hObject,'String')) returns contents of edit14 as a double


% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit15_Callback(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit15 as text
%        str2double(get(hObject,'String')) returns contents of edit15 as a double


% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit16_Callback(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit16 as text
%        str2double(get(hObject,'String')) returns contents of edit16 as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit17_Callback(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit17 as text
%        str2double(get(hObject,'String')) returns contents of edit17 as a double


% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit18 as text
%        str2double(get(hObject,'String')) returns contents of edit18 as a double


% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit19_Callback(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit19 as text
%        str2double(get(hObject,'String')) returns contents of edit19 as a double


% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in togglebutton1.
function togglebutton1_Callback(hObject, eventdata, handles)
% hObject     handle to togglebutton1 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton1



function varargout = result(varargin)
% RESULT MATLAB code for result.fig
%      RESULT, by itself, creates a new RESULT or raises the existing
%      singleton*.
%
%      H = RESULT returns the handle to a new RESULT or the handle to
%      the existing singleton*.
%
%      RESULT('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in RESULT.M with the given input
arguments.
%
%      RESULT('Property','Value',...) creates a new RESULT or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before result_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to result_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help result

% Last Modified by GUIDE v2.5 19-Aug-2016 16:50:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
```

```
                     'gui_OpeningFcn', @result_OpeningFcn, ...
                     'gui_OutputFcn',  @result_OutputFcn, ...
                     'gui_LayoutFcn',  [] , ...
                     'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before result is made visible.
function result_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to result (see VARARGIN)

% Choose default command line output for result
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes result wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = result_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
axes(handles.axes1);
```

```
filename='屋顶俯视图.jpg';
pathname='C:\Users\lrw\Desktop\';
fpath=[pathname filename];%将文件名和目录名组合成一个完整的路径。
imshow(imread(fpath));%用 imread 读入图片，并用 imshow 在 axes_src 上显示。

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double


% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```