

# 太阳影子定位

## 摘要

在视频的数据分析中，确定视频的拍摄日期及地点非常重要，通过太阳影子的变化确定上述指标是一个有效方法。

针对问题一，我们以直杆影长为因变量，以太阳高度角、地理纬度、太阳赤纬和时角为自变量，建立影长的变化模型。采用控制变量法分析某一参量对影长的影响，得到影长关于各参量的变化规律如图 1、2、3 所示。再将题目信息代入，得出结论：直杆影子长度在 9:00~12:00 间逐渐减小，在 12:00~15:00 间逐渐增大，具体变化曲线见图 4，影长在 12:00 取到最小值 3.8978 米，在 9:00 和 15:00 时取到最大值 7.0688 米。

针对问题二，我们建立了基于**多目标规划**的影子地点模型，第一个目标函数为一个小时内太阳方位角变化量与影子实际角度变化量的差值最小，第二个目标函数为理论测量影长与实际影长之差的和最小；约束条件为经纬度范围、时角范围与杆长范围。利用**遍历法**和**粒子群算法**，求得直杆最佳长度为 2 米，根据杆长计算满足条件的可行解，得到最优解： $2^{\circ}3'40.32''\text{S}$ ， $73^{\circ}19'57.326''\text{W}$ 。

针对问题三，该题在问题二基础上增加了日期求解，因此我们仍建立**多目标规划**模型，求解影子地点及日期。目标函数与问题二一致，约束条件需在问题二基础上增加日期范围；我们采用**遍历法**和**粒子群算法**，求得直杆最佳长度为 2 米，接着得出结论：附件 2 的最优解为  $39^{\circ}16'12''\text{N}$ ， $89^{\circ}43'12''\text{W}$ ，日期为 2015 年 6 月 20 日；附件 3 最优解为  $36^{\circ}44'24''\text{S}$ ， $61^{\circ}55'48''\text{W}$ ，日期为 2015 年 6 月 28 日。

针对问题四，我们建立了**透视投影模型**，应用透视投影的相关知识将视频图像中的二维画面转化成三维显示，建立合理坐标系并求解出相应变换矩阵。利用视频中的第一帧及最后一帧图像，通过 MATLAB 软件模拟得到三维坐标系下影子的顶点坐标，求解出影子的变化角度及变化长度。最后利用问题二中的模型对视频的拍摄地点进行求解，得出直杆所处的可能位置有： $26^{\circ}10'48''\text{S}$ ， $48^{\circ}26'24''\text{W}$ ； $44^{\circ}6'\text{N}$ ， $57^{\circ}20'48''\text{W}$ 。

**关键词：**多目标规划 遍历法 粒子群算法 透视投影模型

## 一、问题重述

我们可通过太阳影子定位技术，即分析一段视频中某个物体的太阳影子变化过程，确定该视频拍摄的日期及时间。根据附件解决下列问题。

- 1.建立关于影子的长度变化模型，试分析影子长度与各参数间的变化规律。应用该模型绘制出 2015.10.22 北京时间 9:00~15:00 时天安门广场（39°54'26"N，116°23'29"E）上一根三米高直杆的太阳影子的长度变化曲线。
- 2.根据水平地上某一固定直杆的影子顶部坐标，建立模型确定其所处地点，并将该模型应用于附件 1，求出若干个可能的地点。
- 3.根据水平地上某一固定直杆的影子顶部坐标，建立模型确定其所处日期及地点，并将该模型应用于附件 2、3，求出若干个可能的日期及地点。
- 4.利用附件 4 中的视频，已知视频中直杆高 2 米，建立模型确定该视频的拍摄地点，并利用该模型求解出几个有可能的拍摄地。

若未知拍摄日期，能否根据附件视频求得拍摄的日期及地点？

## 二、模型假设

- 1.假设 2015.10.22 北京时间 9:00~15:00 内天安门广场上直杆的太阳影子一直存在。
- 2.假设 10 月 22 日在 9.23（秋分）至 12.22（冬至）这段时期内的  $\frac{1}{3}$  处。
- 3.假设附件一中每隔三分种的影子变化是每两个点间的影子变化。
- 4.假设人眼到成像面的距离为  $D=0.1$  米。

## 三、符号说明

纬度：北半球为正，南半球为负

经度：东半球为正，西半球为负

$\beta$	太阳直射角	单位：度
$\delta$	太阳赤纬	单位：度、分、秒
$\varphi$	地理纬度	单位：度、分、秒；弧度
$t$	时角	单位：度、分、秒
$N$	日数	单位：天；

$Y$	天数	单位：天
$t_0$	北京时角	单位：度、分、秒
$\alpha$	每两个时间点的直杆影子角度变化	单位：度
$\alpha_{i\varphi}^1, \alpha_{i\varphi}^{21}$	在经纬度为 $i\varphi$ 时第 1、21 个时间点直杆影子太阳方位角	单位：度
$L_{i\varphi}^n$	表示在 $i\varphi$ 经纬度时， $n$ 个时间点上直杆影子的实际长度	单位：米
$L^n$	在第 $n$ 个时间点下，直杆影子的理论测量长度	单位：米
$h$	表示直杆长度	单位：米
$D$	人眼到二维成像点的距离	单位：米

## 四、问题分析

### 4.1 问题一的分析

针对问题一，题目需要我们利用影响影子长度的各参数间的关系建立数学模型。由题目信息可知，模型中因变量为影子长度，自变量为地理纬度、日期、时间段等，然后对这些参数进行逐一分析，可得到影长关于各参数的变化规律，再将直杆的这几项参数代入模型中，求解出直杆太阳影子的长度变化曲线。

### 4.2 问题二的分析

针对问题二，为了确定直杆所在地，我们需对直杆的经纬度进行求解。经纬度会对直杆的影子长度造成影响，因此我们可通过求解直杆的影子长度来确定它的所处位置。利用问题一中得到的影响因素和模型，在满足经纬度范围等条件下对其进行求解。

### 4.3 问题三的分析

针对问题三，除了直杆方位外，还需求解出直杆所处的日期，与问题二相似度较高，因此，我们只需在问题二的模型基础上增加一个对日期的限制条件，再利用相同的方法求解即可。

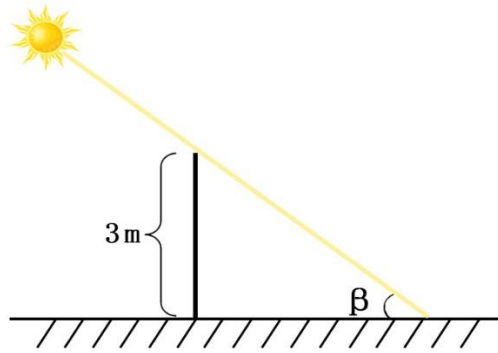
### 4.4 问题四的分析

针对问题四，由于视频图像为二维画面，为了确定视频的拍摄地点，应将二维图像转化为三维平面图，利用视频中影子的长度变化及角度变化，可得影子在坐标系中的顶点坐标变化，再利用问题二中的方法进行求解。

## 五、模型的建立及求解

### 5.1 问题一模型的建立及求解

#### 5.1.1 模型建立



直杆的太阳影子长度可由直杆长度和太阳高度角  $\beta$  决定。根据百度百科资料可知，太阳高度角与地理纬度  $\varphi$ 、太阳赤纬  $\delta$  和时角  $t$  有关。

一般地，我们认为9月23日为秋分，此时太阳直射在赤道，12月22日为冬至，此时太阳直射在 $-23.5^\circ$ ，所以我们假设10月22日在秋分到冬至这段时期内的 $\frac{1}{3}$ 处，因此该日的太阳赤纬在 $-\frac{23.5^\circ}{3}$ 处，由于；我们需求解该日9:00~15:00的直杆太阳影子变化，时间的推移会对直杆影子产生影响，该影响因素即为时角，时角随着地球的自转发生变化，且与观测地所处的子午圈有关。一天中的0~24h对应着 $0\sim 360^\circ$ ，即 $1h=15^\circ$ ， $1min=15'$ 。

令太阳影子长度为 $L$ ，根据三角关系，建立得到下列模型：

$$\begin{cases} L = 3 \times \cot \beta \\ \sin \beta = \sin \varphi \sin \delta + \cos \varphi \cos \delta \cos t \end{cases}$$

其中， $\beta = 90^\circ - |\varphi - \delta|$ ，即 $\sin \beta = \cos(\varphi - \delta)$ ； $\varphi = -\frac{23.5^\circ}{3}$ ；令时间为 $st$ ，因此 $t = 15^\circ \times st$ 。

#### 5.1.2 参数分析

直杆的太阳影子长度受地理纬度、太阳赤纬和时角的影响，利用上述数学

模型，采用控制变量法，对影长关于各参数的变换规律进行分析。由于问题一中需求解天安门广场上某个时段内一根 3 米长直杆的影长，故我们以该直杆为例分析，取地理纬度为天安门广场纬度：39°54'26"，太阳赤纬为 2015.10.22 对应的角度（即 $-\frac{23.5^\circ}{3}$ ），时间为北京时间上午九点。分析具体参数时控制其余变量不变。

### ①影子长度关于地理纬度的变化规律

分析地理纬度对影子长度的影响时，我们控制太阳赤纬和时角不变，仅改变地理纬度。利用 MATLAB 软件绘制出影子长度关于地理纬度的变化曲线如下，横坐标为地理纬度（-180° ~180° ），纵坐标为影子长度（单位：米），图中红线示影子长度为零的分界线：

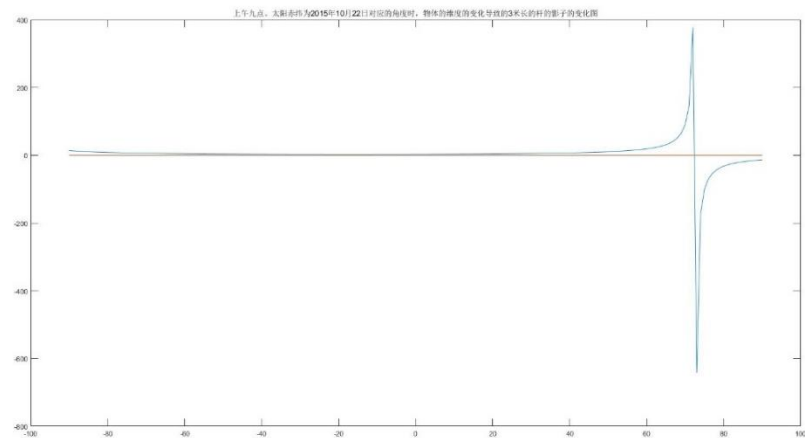


图 1：地理纬度对影长的影响

由上图可知，当太阳赤纬为 $-\frac{23.5^\circ}{3}$ ，时间为 2015.10.22 北京时间上午九点时，当纬度在（-90°， 30°）范围内，一根三米直杆的影子长度趋于零且无明显变化，当纬度在（30°， 72.5°）范围内，影子长度逐渐增大；当纬度在（72.5°， 90°）范围内，影子不存在。

### ②影子长度关于太阳赤纬的变化规律

分析太阳赤纬对影子长度的影响时，我们控制地理纬度和时角不变，仅改变太阳赤纬。由于太阳赤纬随着日期的变化而变化，通过查询资料可知，太阳赤纬与日期存在以下数学关系：

$$\delta = 0.006918 - 0.399912 \cos b + 0.070257 \sin b - 0.006758 \cos(2b) + 0.000907 \sin(2b) - 0.002697 \cos(3b) + 0.00148 \sin(3b)$$

上式中， $\delta$ 、 $b$  单位为弧度。其中：

$$b = \frac{2\pi \times (N - 1)}{Y}$$

$N$  为日数，以每年的 1 月 1 日为第一天； $Y$  为天数，若为平年，则天数为 365，若为闰年，天数等于 366。由于 2015 年为平年，故上式中  $Y=365$ 。

因此我们可用日期来刻画太阳赤纬。利用 MATLAB 软件绘制出影子长度关于太阳赤纬的变化曲线如下，横坐标为日期（单位：天），纵坐标为影子长度（单位：米）：

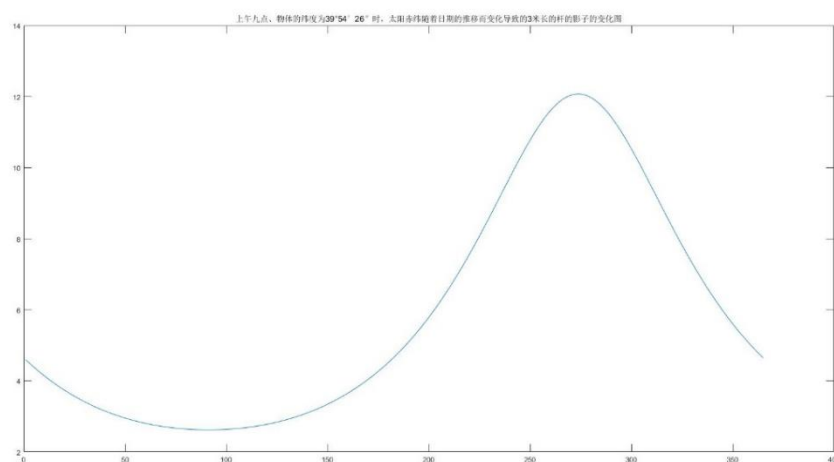


图 2：太阳赤纬对影长的影响

由上图可知，当地理纬度为  $39^{\circ}54'26''$ ，时间北京时间上午九点时，以每年的 1 月 1 日为第一天开始计数，当计数天数为  $(0, 91)$ ， $(274, 365)$  天时，直杆的影子长度逐渐减小；当计数天数为  $(91, 274)$  天时，直杆的太阳影子长度逐渐增大。

### ③影子长度关于时角的变化规律

分析时角对影子长度的影响时，我们控制地理纬度和太阳赤纬不变，仅改变时角。利用 MATLAB 软件绘制出影子长度关于时角的变化曲线如下，横坐标为日期，纵坐标为影子长度（单位：米），图中红线是影子长度为零的分界线：

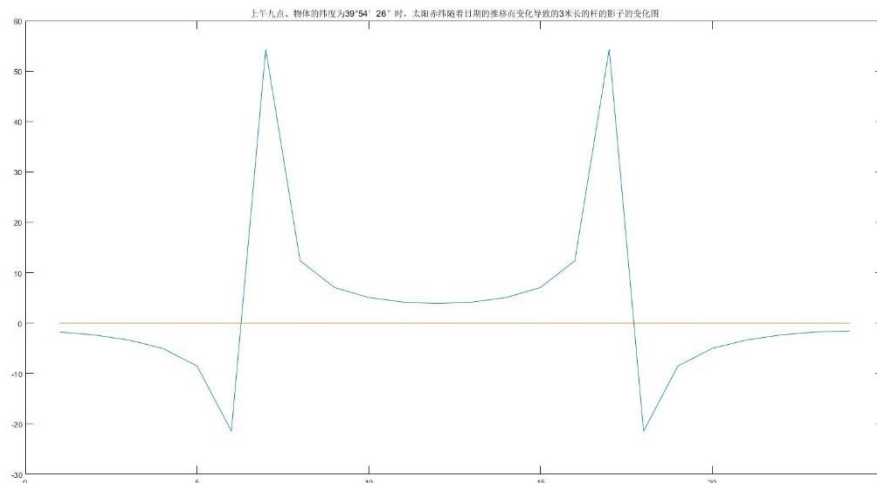


图 3：时角对影长的影响

由上图可知，当地理纬度为  $39^{\circ}54'26''$ ，太阳赤纬为  $-\frac{23.5^{\circ}}{3}$  时，当时间在 0:00~6:30, 17:30~24:00 范围内，影子不存在；当时间在 6:30~7:00, 12:00~17:00 范围内，影子长度逐渐增大；当时间在 7:00~12:00, 17:00~17:30 范围内，影子逐渐减小。

### 5.1.3 直杆影长变化曲线的求解

根据上述模型，利用 MATLAB 软件求解，绘制出 2015.10.22 北京时间上午 9:00~下午 15:00 间天安门广场上一根三米高直杆的影子长度变化曲线，横坐标为时间，纵坐标为影子长度（单位：米）：

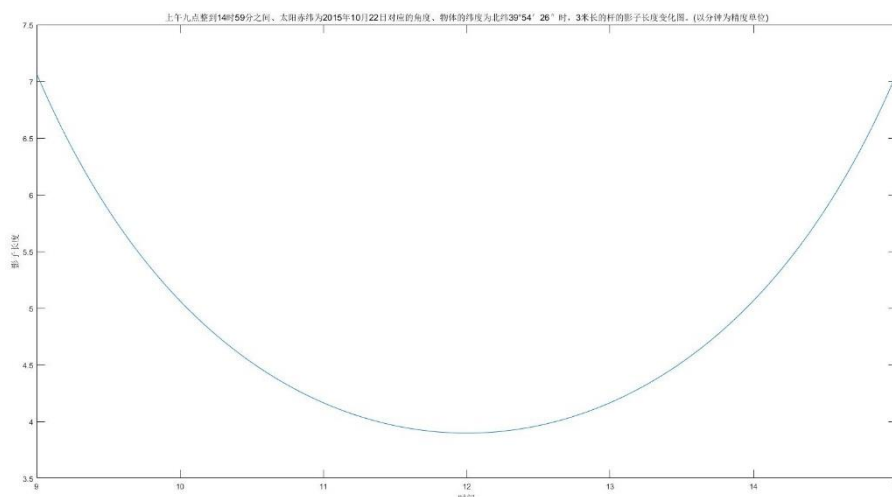


图 4：直杆影长在 9:00~15:00 的变化曲线

由上图可知，直杆影子长度在 9:00~12:00 间逐渐减小，在 12:00~15:00 间逐渐增大。直杆影长在中午 12:00 取到最小值 3.8978 米，在 9:00 和 15:00 时取到

最大值 7.0688 米。

#### 5.1.4 结果分析

由图 1 可知，当太阳赤纬为 $-\frac{23.5^\circ}{3}$ ，时间为 2015.10.22 北京时间上午九点时，当纬度在 $(-90^\circ, 30^\circ)$ 范围内，一根三米直杆的影子长度趋于零且无明显变化，当纬度在 $(30^\circ, 72.5^\circ)$ 范围内，影子长度逐渐增大；当纬度在 $(72.5^\circ, 90^\circ)$ 范围内，影子不存在。

由图 2 可知，当地理纬度为 $39^\circ 54' 26''$ ，时间为北京时间上午九点时，以每年的 1 月 1 日为第一天开始计数，当计数天数为 $(0, 91)$ ， $(274, 365)$ 天时，直杆的影子长度逐渐减小；当计数天数为 $(91, 274)$ 天时，直杆的太阳影子长度逐渐增大。

由图 3 可知，当地理纬度为 $39^\circ 54' 26''$ ，太阳赤纬为 $-\frac{23.5^\circ}{3}$ 时，当时间在 $0:00\sim 6:30, 17:30\sim 24:00$ 范围内，影子不存在；当时间在 $6:30\sim 7:00, 12:00\sim 17:00$ 范围内，影子长度逐渐增大；当时间在 $7:00\sim 12:00, 17:00\sim 17:30$ 范围内，影子逐渐减小。

根据我们建立的模型，在 2015 年 10 月 22 日，直杆影子长度在 $9:00\sim 12:00$ 间逐渐减小，在 $12:00\sim 15:00$ 间逐渐增大，具体变化曲线如图 4 所示，直杆影长在中午 12:00 取到最小值 3.8978 米，在 9:00 和 15:00 时取到最大值 7.0688 米。

### 5.2 问题二模型的建立及求解——基于多目标规划的影子地点模型

对于确定直杆的具体地点问题，需确定该影子所在地的经度与纬度。而地球上的经纬度有一个固定的范围，可作为约束条件来考虑；当直杆的理论影长和实际影长越接近且太阳方位角的变化值与实际角度变化值越接近时，越能更好地体现模型的正确性。因此，问题二是一个多目标规划问题。

#### 5.2.1 模型建立

##### ①时角的确定

问题一中我们利用北京时间来刻画时角，但在全球范围内，我们并不能采用北京时间。根据地理知识可知，全球各地时间由该地所处经度确定，故我们以直杆所在地的经度与北京经度（东经 $120^\circ$ ）的差值来刻画它的时角：



$$t = t_0 + (\text{物体所在地经度} - 120^\circ)$$

上式中,  $t_0$  表示北京时角。

## ②基本假设

- (1) 假设附件一中每隔三分钟的变化是每两个点间的变化。
- (2) 假设直杆的长度在 0~5 米间。

## ③模型新增参数

我们令  $\alpha$  为每两个时间点的直杆影子角度变化,  $\alpha_{i\varphi}^1$  与  $\alpha_{i\varphi}^{21}$  表示在经纬度为  $i\varphi$  时第 1、21 个时间点直杆影子太阳方位角,  $\alpha_{i\varphi}$  表示这两个太阳方位角之差;  $L_{i\varphi}^n$  表示在  $i\varphi$  经纬度时,  $n$  个时间点下直杆影子的实际长度,  $L^n$  表示在  $n$  个时间点下, 直杆影子的理论测量长度, 其中  $n=1,2,\dots,21$ ;  $t$  表示时角;  $h$  表示直杆长度。

## ④模型建立

根据以上讨论, 我们建立如下目标函数, 使一个小时内太阳方位角变化量与影子的实际角度变化量的差值最小, 并使 21 个时间点下的理论测量影长与实际影长之差的和最小:

$$\begin{cases} \min Z_1 = |\alpha_{i\varphi} - \alpha| \\ \min Z_2 = \sum_{n=1}^{21} |L_{i\varphi}^n - L^n| \end{cases}$$

上式中:

$$\alpha_{i\varphi} = \alpha_{i\varphi}^{21} - \alpha_{i\varphi}^1$$

根据太阳方位角的百度百科资料可知, 太阳方位角的求解公式为:

$$\cos \alpha_{i\varphi}^n = \frac{\sin \delta - \sin \theta \sin \varphi}{\cos \theta \cos \varphi}$$

我们以地球经纬度的范围、时角范围及直杆的长度范围 (现实生活中, 直杆长度一般小于 5 米) 作为约束条件:

$$s.t. \begin{cases} -180^\circ \leq i \leq 180^\circ \\ -90^\circ \leq \varphi \leq 90^\circ \\ 0^\circ \leq t \leq 360^\circ \\ 0 < h < 5m \end{cases}$$

### 5.2.2 求解过程

上述模型中，由于全球范围较大，不易直接确定直杆的地理位置，因此我们应用粒子群算法对直杆所处方位进行求解；使用该算法前，需先确定直杆长度，由于题中并未给出直杆长度的相关数据，因此我们需自行求解符合条件的最优值。我们将遍历法与粒子群算法结合，先确定一个直杆长度，再利用粒子群算法求解该长度对应的直杆方位及目标函数的解，遍历完成后，比对各长度所对应的目标函数的解，寻找最优解及其直杆方位。其中，直杆的实际影长和实际角度变化值可根据附件一中的数据，利用勾股定理及三角函数求出。

#### ①基于遍历和粒子群算法的直杆长度及方位求解

首先我们取长度为 1~5 米间的整数值，利用粒子群算法求解直杆的方位及目标函数的解，根据目标函数解的最小值缩小长度范围；在缩减后的范围内以 0.1 米为最小刻度，继续遍历直至结束。

#### I. 基于粒子群算法 (PSO)<sup>[1]</sup> 的直杆方位求解

在标准 PSO 中，由以下公式可确定出粒子搜索空间时的速度及位置：

$$V_{t+1} = WV_t + a_1 \text{rand}(S_t - x_t) + a_2 \text{rand}(B_t - x_t)$$
$$x_{t+1} = x_t + V_t$$

上述式子中， $a_1$ 、 $a_2$  表示加速常数， $V_t$  表示 t 时刻的速度， $x_t$  表示 t 时刻的位置，W 表示惯性权重， $S_t$  表示 t 时刻粒子的局部最优位置， $B_t$  表示 t 时刻粒子的全局最优位置。rand 表示满足约束条件且均匀分布的随机数。为了避免算法过程中跳过一些较优解，可为粒子飞行速度设置速度上限  $V_{max}$  及速度下限  $V_{min}$ 。

#### II. 算法步骤

**Step1:**将粒子群初始化，在满足模型约束条件的情况下，将 100 个粒子随机撒落在地理范围内，在满足约束条件的前提下，产生这些粒子的初始位置。

**Step2:**按照标准 PSO 中的公式，产生粒子的速度，并对其位置进行更新，若粒子速度超出  $[V_{min}, V_{max}]$ ，则取超出侧限值。

**Step3:**根据目标函数进行计算，并确定出这些粒子的局部最优位置，将这些局部最优位置再次代入目标函数进行计算，从而确定粒子的全局最优位置，得到最终收敛的解。

## ②算法结果

对直杆的五个整数长度进行遍历时，我们发现目标函数的最优解所对应的直杆长度在 0~3 米内。接着我们以 0.1 米为最小刻度，在 0.1~3 米内进行遍历，最终结果如下图所示，横坐标表示直杆长度（单位：0.1 米），纵坐标表示目标函数的解：

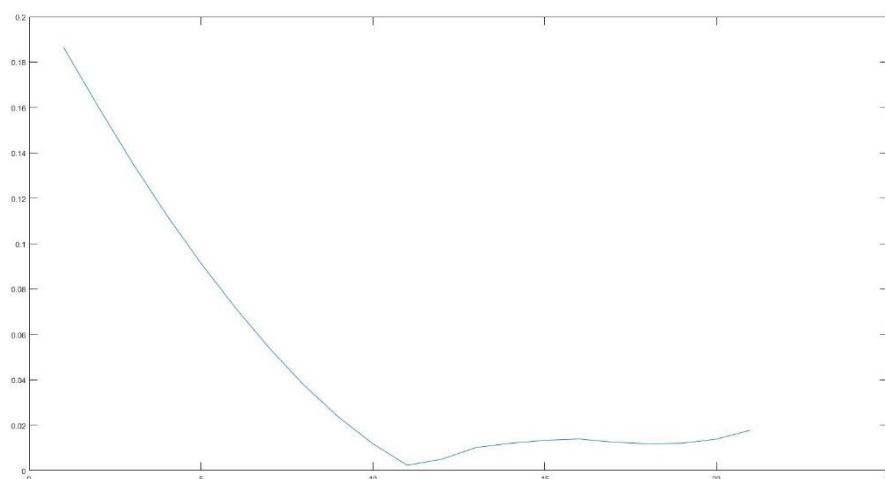


图 5：遍历结果

由上图可知，在直杆长度等于 2 米时，目标函数可取得最小值；利用该直杆长度，结合粒子群算法，得到直杆的经纬度，结合全球地图，确定直杆地点。

### 5.2.3 结果分析

直杆所在处的可能解为：南美洲哥伦比亚和秘鲁分界线处： $2^{\circ}3'40.32''\text{S}$ ， $73^{\circ}19'57.326''\text{W}$ ；加勒比海  $18^{\circ}0'36''\text{N}$ ， $70^{\circ}32'24''\text{W}$ 。



图 6：附件一直杆方位图

## 5.3 问题三模型的建立及求解——基于多目标规划的影子地点、日期模型

对于确定直杆的具体地点及日期问题，在问题二的基础上添加关于日期的约束条件即可。因此我们仍采用多目标规划模型对此问进行求解。

### 5.3.1 模型建立

我们建立如下目标函数，使一个小时内太阳方位角变化量与影子的实际角度变化量的差值最小，并使 21 个时间点下的测量影长与实际影长之差的和最小：

$$\begin{cases} \min Z_1 = |\alpha_{i\varphi} - \alpha| \\ \min Z_2 = \sum_{n=1}^{21} |L_{i\varphi}^n - L^n| \end{cases}$$

我们以地球经纬度范围、时角范围、直杆的长度范围及日期（*date*）作为约束条件：

$$s. t. \begin{cases} -180^\circ \leq i \leq 180^\circ \\ -90^\circ \leq \varphi \leq 90^\circ \\ 0^\circ \leq t \leq 360^\circ \\ 0 < h < 5m \\ 1 < date < 365 \end{cases}$$

### 5.3.2 求解过程

由于全球范围较大，不易直接确定直杆的地理位置，因此我们采用粒子群算法对直杆所处方位进行求解；使用该算法前，需先确定直杆长度，由于题中并未给出直杆长度的相关数据，因此我们需自行求解符合条件的最优值。我们将遍历法与粒子群算法结合，先确定一个直杆长度，再利用粒子群算法求解该长度对应的直杆方位及目标函数的解，遍历完成后，比对各长度所对应的目标函数的解，寻找最优解及该解所对应的直杆方位和日期。

#### ①基于遍历和粒子群算法的直杆长度及方位求解

我们以 0.1 米为最小刻度在 0~5 米范围内进行遍历，利用粒子群算法求解直杆的方位及模型的解，直至遍历结束。

#### I. 基于粒子群算法（PSO）的直杆方位求解

在标准 PSO 中，我们仍通过以下公式确定粒子搜索空间时的速度及位置：

$$V_{t+1} = WV_t + a_1 rand(S_t - x_t) + a_2 rand(B_t - x_t)$$

$$x_{t+1} = x_t + V_t$$

rand 表示满足约束条件且均匀分布的随机数。为了避免算法过程中跳过一些较优解，我们仍为粒子飞行速度设置速度上限  $V_{max}$  及速度下限  $V_{min}$ 。

#### II. 算法步骤

**Step1:** 将粒子群初始化，在满足模型约束条件的情况下，将 100 个粒子随机撒落

在地理范围内，在满足约束条件的前提下，产生这些粒子的初始位置。

**Step2:**按照标准 PSO 中的公式，产生粒子的速度，并对其位置进行更新，若粒子速度超出 $[V_{min}, V_{max}]$ ，则取超出侧限值。

**Step3:**根据目标函数进行计算，并确定出这些粒子的局部最优位置，将这些局部最优位置再次代入目标函数进行计算，从而确定粒子的全局最优位置，得到最终收敛的解。

### 5.3.3 结果分析

对于附件二，直杆所处的可能位置有：美国  $39^{\circ}16'12''N$ ， $89^{\circ}43'12''W$ ，日期为 2015 年 6 月 20 日；太平洋  $39^{\circ}2'24''S$ ， $89^{\circ}49'12''W$ ，日期为 2015 年 11 月 12 日。具体结果如下图所示：



图 7：附件二直杆方位图

对于附件三，直杆所处的可能位置有：阿根廷  $36^{\circ}44'24''S$ ， $61^{\circ}55'48''W$ ，日期为 2015 年 6 月 28 日；大西洋  $36^{\circ}32'24''N$ ， $61^{\circ}55'12''W$ ，日期为 2015 年 11 月 24 日。具体如下图所示：

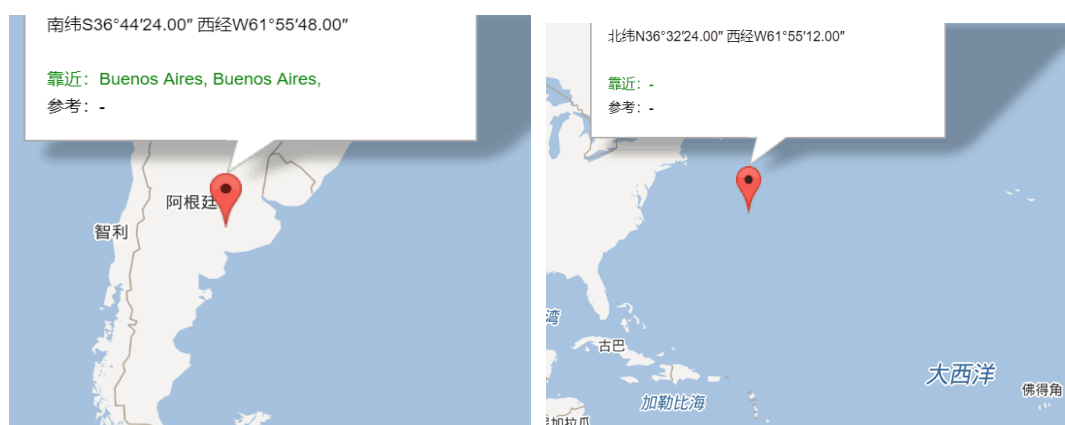


图 8：附件三直杆方位图

## 5.4 问题四模型的建立及求解——基于透视投影模型视频拍摄地求解

我们可利用视频中直杆的影子长度和角度变化，确定附件中视频的拍摄地点。直杆作为一个三维物体，投影变换后以平面图形的形式显示在摄像机中。

因此我们可采用透视投影模型<sup>[6]</sup>对此问进行求解。

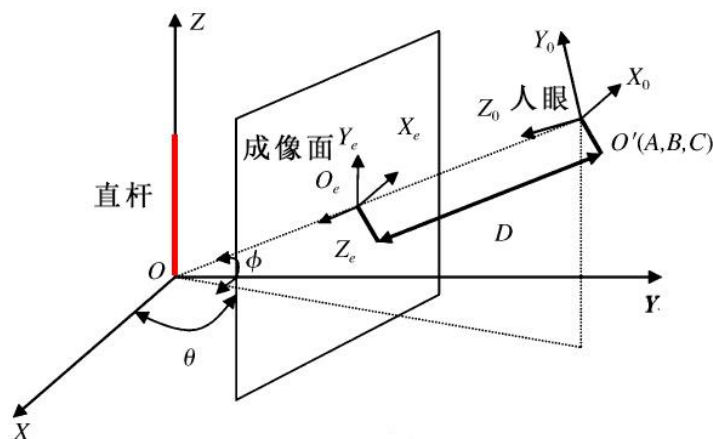
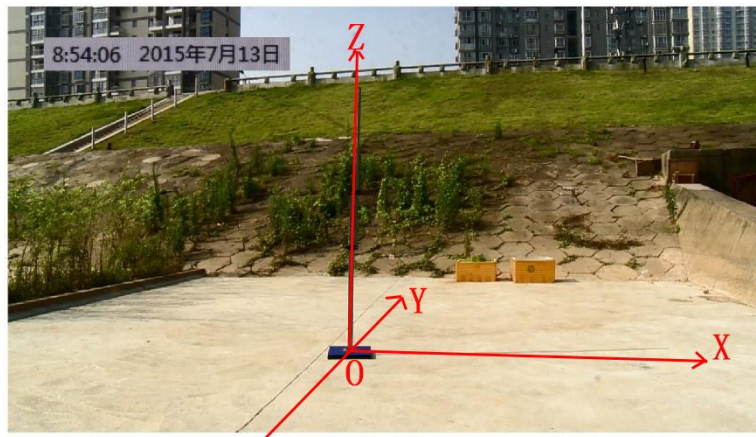
#### 5.4.1 模型建立

##### ①透视投影原理

透视投影可以模拟出人眼观察物体的过程，通过三点（透视中心、成像点、目标点）共线的条件，在改变原本的投影光线时，保持透视面的投影图形不变，在该变换下可使透视图的立体感更强。

##### ②坐标系的建立

将人眼所在的坐标系称为观察坐标系 $O'X_0Y_0Z_0$ （ $O'$ 为人眼所在点），摄像机的成像平面所在的坐标系称为成像面坐标系 $O_eX_eY_eZ_e$ （ $O_e$ 为人眼与 $O$ 点的连线在成像面的交点），视频截图中的直杆所在位置称为实际坐标系 $OXYZ$ （ $O$ 为直杆底端），具体如下图所示：



### ③透视投影矩阵求解

I.  $O$  点到  $O'$  的平移变换矩阵

将实际坐标系的原点  $O$  平移至  $O'(A, B, C)$ ，即可得到新的坐标系

$O'X_1Y_1Z_1$ ，该平移变换矩阵<sup>[11]</sup>为：

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -A & -B & -C & 1 \end{bmatrix}$$

由球坐标系与直角坐标系间的关系<sup>[12]</sup>可得：

$$\begin{cases} A = R \cos \theta \cos \Phi \\ B = R \sin \theta \cos \Phi \\ C = R \sin \Phi \end{cases}$$

II. 将坐标系  $O'X_1Y_1Z_1$  绕  $Z_1$  轴，作  $\theta - \frac{\pi}{2}$  角旋转，使  $Y_1$  负方向与  $Z$  轴相交。因此绕  $Z$  轴转动  $\alpha$  角后的变换矩阵为：

$$T_Z = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

接着我们将  $\alpha = \theta - \frac{\pi}{2}$  代入上式，得变换矩阵为：

$$T_2 = \begin{bmatrix} \sin \theta & \cos \theta & 0 & 0 \\ \cos \theta & \sin \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

III. 将坐标系  $O'X_1Y_1Z_1$  绕  $X_1$  轴，作  $\frac{\pi}{2} + \Phi$  角旋转，使  $Z_1$  轴方向指向原点  $O$ ，这一步的变换矩阵为：

$$T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\sin \Phi & -\cos \Phi & 0 \\ 0 & \cos \Phi & -\sin \Phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

IV. 由于观察坐标系  $O'X_1Y_1Z_1$  为左手坐标系，因此我们需对 III 中的右手坐标系进行转换，转换矩阵为：

$$T_3 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

V.综合变换矩阵  $T$  即为上述四个变换举证的连乘积:

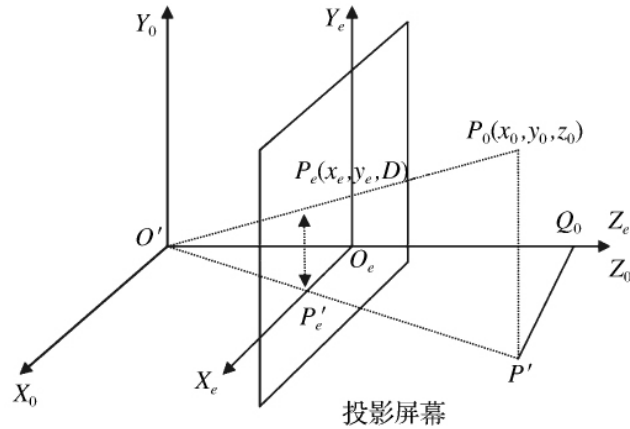
$$T = T_1 \cdot T_2 \cdot T_3 \cdot T_4$$

由此可得:

$$\begin{cases} x_0 = -x \sin \theta + y \cos \theta \\ y_0 = -x \cos \theta \sin \Phi - y \sin \theta \sin \Phi + z \cos \Phi \\ z_0 = -x \cos \theta \cos \Phi - y \sin \theta \cos \Phi - z \sin \Phi + R \end{cases}$$

#### ④成像面投影

观察坐标系  $O'X_0Y_0Z_0$ , 成像面坐标系  $O''X_eY_eZ_e$ ,  $P_0(x_0, y_0, z_0)$  是观察坐标系中的物体,  $P_e$  是视点  $P_0$  在成像面上的透视投影, 具体如下图所示:



由几何知识可得

$$\begin{cases} \frac{O_e P'_e}{Q_0 P'} = \frac{O' O_e}{O' Q_0}, \text{ 即 } \begin{cases} x_e = \frac{x_0}{z_0} \cdot D \\ y_e = \frac{y_0}{z_0} \cdot D \end{cases} \\ \frac{P_e P'}{P_0 P'} = \frac{O' O_e}{O' O_0} \end{cases}$$

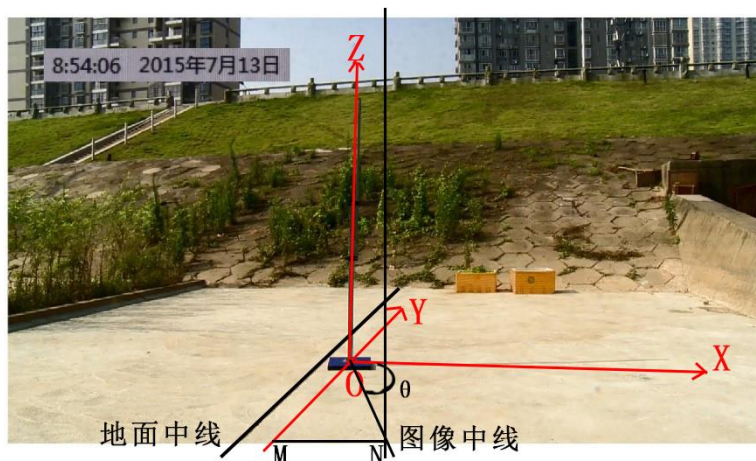
将上式代入 V 中得:

$$\begin{cases} x_e = \frac{-x \sin \theta + y \cos \theta}{-x \cos \theta \cos \Phi - y \sin \theta \cos \Phi - z \sin \Phi + R} \cdot D \\ y_e = \frac{-x \cos \theta \sin \Phi - y \sin \theta \sin \Phi + z \cos \Phi}{-x \cos \theta \cos \Phi - y \sin \theta \cos \Phi - z \sin \Phi + R} \cdot D \end{cases}$$

#### 5.4.2 过程求解

##### ①影子坐标求解





根据附件中视频的第一帧图像，分析得直杆共占用了 658 个像素点，由题已知直杆长度为 2 米，因此图像内一个像素点表示的三维空间内实际长度为  $\frac{2}{658}\text{m}$ 。根据常识可知，图像的中线处即为摄像点，我们计算得到中线与 y 轴的距离（即 MN）为 380 个像素点，因此在三维空间下，摄像点与 y 轴相隔 0.58 米。我们假设摄像头与 x 轴的距离为 3 米，因此  $\theta = -79^\circ$ 。我们假设摄像机一直水平放置，图像的正中间处应为摄像机的摄像中点，处理图片可知图像正中为直杆的中点，因此摄像机的摄像中点高为 1 米。

通过 MATLAB 软件在二维图片中模拟上述条件，不断改变 XOY 平面上影子的坐标来调整它的长度和夹角，直至二维图片中影子与杆之间的夹角等于视频图像中影子与杆之间的夹角，然后我们利用影子的像素点数（802 个像素点）及比例尺关系，模拟出影子的长度，最终确定出影子在二维图片中的顶点坐标（2.355,0.4626）。

我们利用视频的第一帧图像和最后一帧图像，计算得到直杆与影子的夹角变化及影子长度变化如下表所示：

表 1：视频始末图像直杆与影子的变化数据

图像	第一帧	最后一帧
直杆与影子的夹角/度	89.09	90.84
影子长度/米	2.4	2.01

我们利用遍历法，在上述已知条件的基础上在模拟二维图片中寻找最后一帧图像中影子的顶点坐标：（1.958,0.4077）。

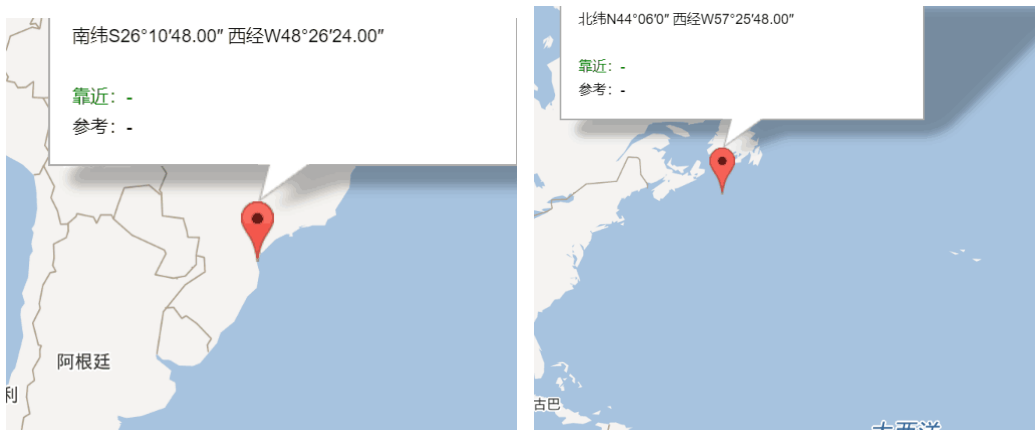
## ②直杆位置求解

由于我们将问题二和问题三中的 20 个 3 分钟间隔近似成 21 个时间点进行运算，故本题我们也将该视频的总历时（40 分 41 秒）平均分成 21 个时间点，将影子的变化角度及长度也作相同处理。

最后我们将这些数据代入问题二中的模型进行求解。

5.4.3 结果分析

直杆所处的可能位置有：阿根廷  $26^{\circ}10'48''S$ ， $48^{\circ}26'24''W$ ；大西洋  $44^{\circ}6'N$ ， $57^{\circ}20'48''W$ 。具体如下图所示：



六、模型评价

6.1 模型的优点

- ①针对问题二、三，将太阳方位角设定为目标函数，避免了坐标系带来的方向干扰问题，便于计算。
- ②针对问题四，利用透视投影模型确定实际太阳影子的长度，考虑到成像时图形的裁剪问题，计算方法较科学可信。
- ③针对问题二和问题三，我们采用粒子群算法求解，可避免在求解最优解的过程中陷入局部最优解而达不到全局最优解的缺陷，可以迅速收敛至全局最优解。

6.2 模型的缺点

- ①针对问题四，对人眼和成像点之间的距离进行假设，计算有一定误差。
- ②针对问题二和问题三，粒子群算法中对参数选取的不同会导致求解的结果出现偏差，使得求解出的结果可能是次优解。

## 七、参考文献

- [1]马丹祥, 初建宇, 王政, 陈灵利.基于多目标规划的防灾避难场所选址模型研究[J]. 自然灾害学报. 2015, 24(2):2-4
- [2]夏军营, 徐小泉, 熊九龙. 利用平行透视投影模型的位姿迭代估计[J]. 光学精密工程. 2012, 20(6):1343-1348
- [3]何援军. 透视和透视投影——论图形变换和投影的若干问题之三[J]. 计算机辅助设计与图形学学报. 2005, 17(4): 735-739
- [4]袁红星, 吴少群, 朱仁祥, 安鹏. 融合对象性和视觉显著度的单目图像 2D 转 3D[J]. 中国图像图形学报. 2013, 18(10): 1478-1485
- [5]李晋芳, 施法中. 从二维视图到三维几何模型转换中视图的包围盒分离法[J]. 计算机工程. 2004, 30(6): 139-140
- [6]王渊, 龙华, 邵玉斌, 李灵玥. 一种图形透视投影矩阵的推导及编程实现[J]. 甘肃科学学报. 2011, 23(4):128-129
- [7]吴迪, 黄文骞, 王莹. 图形变换中透视投影变换矩阵的推导[J]. 海洋测绘. 2003, 23(1):18-21
- [8]卓金武, MATLAB 在数学建模中的应用, 北京: 北京航空航天大学出版社, 2014. 9[M]
- [9]姜启源, 数学模型, 北京: 高等教育出版社, 2011. 1[M]
- [10]司守奎, 数学建模算法与应用, 北京: 国防工业出版社, 2011. 8[M]
- [11]Fletcher,Dunn,Ian Parberry. 3D数学基础: 图形与游戏开发[M]. 北京: 清华大学出版社, 2005.
- [12]同济大学数学系编.高等数学(下册)[M].第6版.北京: 高等教育出版社, 2007

附件

第一题

```
clear

solar_angle = 0; % 太阳赤纬
time = [9 0 0]; % 地方时间[时 分 秒]
direction = 1; % 北纬是 1, 南纬是-1
dimension = [39 54 26]; % 杆的纬度, 由[度 分 秒]组成
len_staff = 3; % 杆的长度(米)
dates = [2015 10 22]; % [年 月 日]

days = datenum(dates)-datenum([2015 3 21]); % 计算当天距离春分多
少天
solar_angle = ((23.50*sin(((2*pi)/365)*days))/(360)*(2*pi); % 由
日期计算太阳赤纬, 精确到小数点后两位, 北纬是正的, 南纬是负的, 弧度制
for k = 9:15
    time = [k 0 0];
    time_angle =
((15*time(1)+(15*time(2))/60+((15*time(3))/60)*0.1)/360)*(2*pi
)-pi; % 计算出时角, 精确到小数点后两位, 弧度制
    dimension_angle =
direction*(((dimension(1)+dimension(2))/60+(dimension(3)/60)*0.
1)/360)*(2*pi)); % 计算维度, 精确到小数点后两位, 北纬是正的, 南纬是
负的, 弧度制
    sin_altitude_angle =
sin(dimension_angle)*sin(solar_angle)+cos(dimension_angle)*cos
(solar_angle)*cos(time_angle); % 求太阳高度角的正弦值
    cos_altitude_angle = sqrt(1-sin_altitude_angle^2); % 求太阳
```

高度角的余弦值

```
len_shadow(k-8) =  
len_staff*(cos_altitude_angle/sin_altitude_angle); % 求杆的影子  
长度(米)  
end  
len_shadow  
plot(9:15, len_shadow)
```

第二题-影子长度规划

```
clear;  
tic;  
  
% 导入数据  
points = xlsread('附件 1-3.xls', '附件 1', 'B4:C24');  
l0 = transpose(sqrt(points(:,1).^2+points(:,2).^2));  
  
% 实际问题的参数设置  
h = 3; % 假设杆的高度为 3 米  
days = datenum([2015 4 18])-datenum([2015 3 21]); % 由题目已知计  
算  
solar = ((23.5*sin(((2*pi)/365)*days))/(360)*(2*pi); % 由题目已  
知计算  
t0 = [220.5 221.25 222 222.75 223.5 224.25 225 225.75 226.5  
227.25 228 228.75 229.5 230.25 231 231.75 232.5 233.25 234  
234.75 235.5]; % 时间角向量, 角度制  
rad_t0 = (t0/360)*(2*pi); % 时间角向量, 弧度制  
  
% 设置算法参数  
E0 = 0.05; % 允许误差  
maxNum = 10000; % 粒子最大迭代次数
```

```

narvs = 2; % 目标函数的自变量个数
particleSize = 100; % 粒子群规模
c1 = 2; % 每个粒子的个体学习因子，也称加速常数
c2 = 2; % 每个粒子的社会学习因子，也称加速常数
w = 0.6; % 惯性因子
vMax = 27; % 粒子的最大飞行速度，单位“度/次”

for v0 = 1:90
    % 初始化粒子群状态
    x = zeros(particleSize, narvs);
    x(:,1) = 180*(rand(particleSize,1)-0.5); % 粒子的所在维度，单位“度”
    x(:,2) = 360*(rand(particleSize,1)-0.5); % 粒子的所在经度，单位“度”
    v = v0*rand(particleSize,narvs); % 粒子的飞行速度，单位“度/次”

    for m = 1:particleSize
        rad_x = (x(m,1)./360).*(2*pi); % 将纬度角度制的 x 转换为弧度制的 rad_x
        t = t0 + (x(m,2)-120); % 计算各个维度所对应的实际的时间角
        rad_t = (t./360).*(2*pi); % 将 x 的经度对应的时间角转换为弧度制 rad_t
        sin_h =
sin(rad_x(1)).*sin(solar)+cos(rad_x(1)).*cos(solar).*cos(rad_t
);
        cos_h = sqrt(1-sin_h.^2);
        l = h.*(cos_h./sin_h);
        y(m) = sum(abs(l-l0));
    end
end

```

```

personalBest_x = x;
personalBest_faval = y;
[globalBest_faval m] = min(personalBest_faval);
globalBest_x = personalBest_x(m,:);
k = 1;
while k<=maxNum
    for m = 1:particleSize
        rad_x = (x(m,1)./360).*(2*pi); % 将纬度角度制的 x 转换
为弧度制的 rad_x
        t = t0 + (x(m,2)-120); % 计算各个维度所对应的实际的时间角
        rad_t = (t./360).*(2*pi); % 将 x 的经度对应的时间角转换为弧度制 rad_t
        sin_h =
sin(rad_x(1)).*sin(solar)+cos(rad_x(1)).*cos(solar).*cos(rad_t
);
        cos_h = sqrt(1-sin_h.^2);
        l = h.*(cos_h./sin_h);
        y(m) = sum(abs(l-l0));
        if y(m)<personalBest_faval % 判断当前位置是否是历史上
最佳位置
            personalBest_faval(m) = y(m);
            personalBest_x(m,:) = x(m,:);
        end
    end
    [globalBest_faval m] = min(personalBest_faval);
    globalBest_x = personalBest_x(m,:);
    for m = 1:particleSize % 更新粒子群里每个个体的最新位置
        v(m,:) = w*v(m,.)+c1*rand*(personalBest_x(m,.)-
x(m,.)+c2*rand*(globalBest_x-x(m,.)));

```

```

        for n = 1:narvs % 判断粒子的飞行速度是否超过最大飞行速度
            if v(m,n)>vMax
                v(m,n) = vMax;
            elseif v(m,n)<-vMax
                v(m,n) = -vMax;
            end
        end
        x(m,:) = x(m,:)+v(m,:);
    end
    if abs(globalBest_faval)<E0
        break
    end
    k = k+1;
end
globalBest_favals(v0) = globalBest_faval;
globalBest_xs(v0,:) = globalBest_x;
end
plot(1:v0, globalBest_favals, 'b')
hold on
plot(1:v0, globalBest_xs(:,1), 'm')
plot(1:v0, globalBest_xs(:,2), 'r')
hold off
toc;

```

## 第二题-角度规划

```

clear;
tic;

% 实际问题的参数设置

```



```

angle0 = 7.0679;
h = 3; % 假设杆的高度为 3 米
days = datenum([2015 4 18])-datenum([2015 3 21]); % 由题目已知计算
solar = ((23.5*sin(((2*pi)/365)*days))/360)*(2*pi); % 由题目已知计算
t0 = [220.5 235.5]; % 时间角向量，角度制
rad_t0 = (t0/360)*(2*pi); % 时间角向量，弧度制

% 设置算法参数
E0 = 0.05; % 允许误差
maxNum = 10000; % 粒子最大迭代次数
narvs = 2; % 目标函数的自变量个数
particleSize = 100; % 粒子群规模
c1 = 2; % 每个粒子的个体学习因子，也称加速常数
c2 = 2; % 每个粒子的社会学习因子，也称加速常数
w = 0.6; % 惯性因子
vMax = 27; % 粒子的最大飞行速度，单位“度/次”

for v0 = 1:90
    % 初始化粒子群状态
    x = zeros(particleSize, narvs);
    x(:,1) = 180*(rand(particleSize,1)-0.5); % 粒子的所在维度，单位“度”
    x(:,2) = 360*(rand(particleSize,1)-0.5); % 粒子的所在经度，单位“度”
    v = zeros(particleSize, narvs);
    v(:,1) = v0*rand(particleSize,1); % 粒子所在的维度的飞行速度，单位“度/次”

```

```

v(:,2) = 2*v0*rand(particleSize,1); % 粒子所在的经度的飞行速度，单位“度/次”

for m = 1:particleSize
    rad_x = (x(m,1)./360).*(2*pi); % 将纬度角度制的 x 转换为弧度制的 rad_x
    t = t0 + (x(m,2)-120); % 计算各个维度所对应的实际的时间角
    rad_t = (t./360).*(2*pi); % 将 x 的经度对应的时间角转换为弧度制 rad_t
    sin_h =
sin(rad_x(1)).*sin(solar)+cos(rad_x(1)).*cos(solar).*cos(rad_t
);
    cos_h = sqrt(1-sin_h.^2);
    cos_dir1 = (sin(solar)-
sin_h(1)*sin(x(m,1)))/(cos_h(1)*cos(x(m,1)));
    cos_dir2 = (sin(solar)-
sin_h(2)*sin(x(m,1)))/(cos_h(2)*cos(x(m,1)));
    y(m) = abs(abs((acos(cos_dir1)/(2*pi))*360-
(acos(cos_dir2)/(2*pi))*360)-angle0);
end
personalBest_x = x;
personalBest_faval = y;
[globalBest_faval m] = min(personalBest_faval);
globalBest_x = personalBest_x(m,:);
k = 1;
while k<=maxNum
    for m = 1:particleSize
        rad_x = (x(m,1)./360).*(2*pi); % 将纬度角度制的 x 转换为弧度制的 rad_x

```

```

        t = t0 + (x(m,2)-120); % 计算各个维度所对应的实际的时间角

        rad_t = (t./360).*(2*pi); % 将 x 的经度对应的时间角转换为弧度制 rad_t

        sin_h =
sin(rad_x(1)).*sin(solar)+cos(rad_x(1)).*cos(solar).*cos(rad_t
);

        cos_h = sqrt(1-sin_h.^2);
        cos_dir1 = (sin(solar)-
sin_h(1)*sin(x(m,1)))/(cos_h(1)*cos(x(m,1)));
        cos_dir2 = (sin(solar)-
sin_h(2)*sin(x(m,1)))/(cos_h(2)*cos(x(m,1)));
        y(m) = abs((acos(cos_dir1)/(2*pi))*360-
(acos(cos_dir2)/(2*pi))*360);

        if y(m)<personalBest_faval % 判断当前位置是否是历史上最佳位置

            personalBest_faval(m) = y(m);
            personalBest_x(m,:) = x(m,:);
        end
    end

    [globalBest_faval m] = min(personalBest_faval);
    globalBest_x = personalBest_x(m,:);

    for m = 1:particleSize % 更新粒子群里每个个体的最新位置
        v(m,:) = w*v(m,:)+c1*rand*(personalBest_x(m,:)-
x(m,:))+c2*rand*(globalBest_x-x(m,:));

        for n = 1:narvs % 判断粒子的飞行速度是否超过最大飞行速度

            if v(m,n)>vMax
                v(m,n) = vMax;
            elseif v(m,n)<-vMax

```

```

        v(m,n) = -vMax;
    end
end
x(m,:) = x(m,:)+v(m,:);
end
if abs(globalBest_faval)<E0
    break
end
k = k+1;
end
globalBest_favals(v0) = globalBest_faval;
globalBest_xs(v0,:) = globalBest_x;
end
plot(1:v0, globalBest_favals, 'b')
hold on
plot(1:v0, globalBest_xs(:,1), 'm')
plot(1:v0, globalBest_xs(:,2), 'r')
hold off
toc;

```

## 第二题-求最优高度

```

clear;
tic;

% 导入数据
points = xlsread('附件 1-3.xls', '附件 1', 'B4:C24');
l0 = transpose(sqrt(points(:,1).^2+points(:,2).^2));

% 实际问题的参数设置
h = 1; % 假设杆的高度为 3 米

```

```

days = datenum([2015 4 18])-datenum([2015 3 21]); % 由题目已知计算
solar = ((23.5*sin(((2*pi)/365)*days))/360)*(2*pi); % 由题目已知计算
t0 = [220.5 221.25 222 222.75 223.5 224.25 225 225.75 226.5
227.25 228 228.75 229.5 230.25 231 231.75 232.5 233.25 234
234.75 235.5]; % 时间角向量，角度制
rad_t0 = (t0/360)*(2*pi); % 时间角向量，弧度制

% 设置算法参数
E0 = 0; % 允许误差
maxNum = 1000; % 粒子最大迭代次数
narvs = 2; % 目标函数的自变量个数
particleSize = 100; % 粒子群规模
c1 = 2; % 每个粒子的个体学习因子，也称加速常数
c2 = 2; % 每个粒子的社会学习因子，也称加速常数
w = 0.6; % 惯性因子
vMax = 27; % 粒子的最大飞行速度，单位“度/次”
p = 0;
for h = 1:0.1:3
    for v0 = 1:90
        % 初始化粒子群状态
        x = zeros(particleSize, narvs);
        x(:,1) = 180*(rand(particleSize,1)-0.5); % 粒子的所在维度，单位“度”
        x(:,2) = 360*(rand(particleSize,1)-0.5); % 粒子的所在经度，单位“度”
        v = v0*rand(particleSize,narvs); % 粒子的飞行速度，单位“度/次”
    end
end

```

```

        for m = 1:particleSize
            rad_x = (x(m,1)./360).*(2*pi); % 将纬度角度制的 x 转换
为弧度制的 rad_x
            t = t0 + (x(m,2)-120); % 计算各个维度所对应的实际的时
间角
            rad_t = (t./360).*(2*pi); % 将 x 的经度对应的时间角转换
为弧度制 rad_t
            sin_h =
sin(rad_x(1)).*sin(solar)+cos(rad_x(1)).*cos(solar).*cos(rad_t
);
            cos_h = sqrt(1-sin_h.^2);
            l = h.*(cos_h./sin_h);
            y(m) = sum(abs(l-l0));
        end
        personalBest_x = x;
        personalBest_faval = y;
        [globalBest_faval m] = min(personalBest_faval);
        globalBest_x = personalBest_x(m,:);
        k = 1;
        while k<=maxNum
            for m = 1:particleSize
                rad_x = (x(m,1)./360).*(2*pi); % 将纬度角度制的 x
转换为弧度制的 rad_x
                t = t0 + (x(m,2)-120); % 计算各个维度所对应的实际
的时间角
                rad_t = (t./360).*(2*pi); % 将 x 的经度对应的时间角
转换为弧度制 rad_t
                sin_h =
sin(rad_x(1)).*sin(solar)+cos(rad_x(1)).*cos(solar).*cos(rad_t
);

```

```

        cos_h = sqrt(1-sin_h.^2);
        l = h.*(cos_h./sin_h);
        y(m) = sum(abs(l-l0));
        if y(m)<personalBest_faval % 判断当前位置是否是历史
置上最佳位置

            personalBest_faval(m) = y(m);
            personalBest_x(m,:) = x(m,:);
        end
    end
    [globalBest_faval m] = min(personalBest_faval);
    globalBest_x = personalBest_x(m,:);
    for m = 1:particleSize % 更新粒子群里每个个体的最新位
置
        v(m,:) = w*v(m,:)+c1*rand*(personalBest_x(m,:)-
x(m,:))+c2*rand*(globalBest_x-x(m,:));
        for n = 1:narvs % 判断粒子的飞行速度是否超过最大飞
行速度

            if v(m,n)>vMax
                v(m,n) = vMax;
            elseif v(m,n)<-vMax
                v(m,n) = -vMax;
            end
        end
        x(m,:) = x(m,:)+v(m,:);
    end
    if abs(globalBest_faval)<E0
        break
    end
    k = k+1;
end
end

```

```

        globalBest_favals(v0) = globalBest_faval;
        globalBest_xs(v0,:) = globalBest_x;
    end
    p = p+1;
    min_globalBest_favals(p) = min(globalBest_favals);
end
plot(1:p, min_globalBest_favals)
toc;

```

### 第三题-影子长度规划

```

clear;
tic;

% 导入数据
points = xlsread('附件 1-3.xls', '附件 2', 'B4:C24');
l0 = transpose(sqrt(points(:,1).^2+points(:,2).^2));

% 实际问题的参数设置
h = 3; % 假设杆的高度为 3 米
t0 = [190.25 191 191.75 192.5 193.25 194 194.75 195.5 196.25
197 197.75 198.5 199.25 200 200.75 201.5 202.25 203 203.75
204.5 205.25]; % 时间角向量，角度制
rad_t0 = (t0/360)*(2*pi); % 时间角向量，弧度制

% 设置算法参数
E0 = 0.05; % 允许误差
maxNum = 10000; % 粒子最大迭代次数
narvs = 3; % 目标函数的自变量个数
particleSize = 100; % 粒子群规模
c1 = 2; % 每个粒子的个体学习因子，也称加速常数

```



```

c2 = 2; % 每个粒子的社会学习因子，也称加速常数
w = 0.6; % 惯性因子
vMax = 30; % 粒子的最大飞行速度，单位“度/次”

for v0 = 1:90
    % 初始化粒子群状态
    x = zeros(particleSize, narvs);
    x(:,1) = 180*(rand(particleSize,1)-0.5); % 粒子的所在维度，单位“度”
    x(:,2) = 360*(rand(particleSize,1)-0.5); % 粒子的所在经度，单位“度”
    x(:,3) = 365*rand(particleSize,1); % 粒子的所在经度，单位“度”
    v = v0*rand(particleSize,narvs); % 粒子的飞行速度，单位“度/次”

    for m = 1:particleSize
        solar = ((23.5*sin(((2*pi)/365)*x(m,3)))/360)*(2*pi); % 由题目已知计算
        rad_x = (x(m,1)./360).*(2*pi); % 将纬度角度制的 x 转换为弧度制的 rad_x
        t = t0 + (x(m,2)-120); % 计算各个维度所对应的实际的时间角
        rad_t = (t./360).*(2*pi); % 将 x 的经度对应的时间角转换为弧度制 rad_t
        sin_h =
sin(rad_x(1)).*sin(solar)+cos(rad_x(1)).*cos(solar).*cos(rad_t
);
        cos_h = sqrt(1-sin_h.^2);
        l = h.*(cos_h./sin_h);
        y(m) = sum(abs(l-l0));
    end
end

```

```

personalBest_x = x;
personalBest_faval = y;
[globalBest_faval m] = min(personalBest_faval);
globalBest_x = personalBest_x(m,:);
k = 1;
while k<=maxNum
    for m = 1:particleSize
        solar =
((23.5*sin(((2*pi)/365)*x(m,3)))/360)*(2*pi); % 由题目已知计算
        rad_x = (x(m,1)./360).*(2*pi); % 将纬度角度制的 x 转换
为弧度制的 rad_x
        t = t0 + (x(m,2)-120); % 计算各个维度所对应的实际的时间角
        rad_t = (t./360).*(2*pi); % 将 x 的经度对应的时间角转换为弧度制 rad_t
        sin_h =
sin(rad_x(1)).*sin(solar)+cos(rad_x(1)).*cos(solar).*cos(rad_t
);
        cos_h = sqrt(1-sin_h.^2);
        l = h.*(cos_h./sin_h);
        y(m) = sum(abs(l-l0));
        if y(m)<personalBest_faval % 判断当前位置是否是历史上
最佳位置
            personalBest_faval(m) = y(m);
            personalBest_x(m,:) = x(m,:);
        end
    end
    [globalBest_faval m] = min(personalBest_faval);
    globalBest_x = personalBest_x(m,:);

```

```

        for m = 1:particleSize % 更新粒子群里每个个体的最新位置
            v(m,:) = w*v(m, :)+c1*rand*(personalBest_x(m,:)-
x(m,:))+c2*rand*(globalBest_x-x(m,:));
            for n = 1:narvs % 判断粒子的飞行速度是否超过最大飞行速
度
                if v(m,n)>vMax
                    v(m,n) = vMax;
                elseif v(m,n)<-vMax
                    v(m,n) = -vMax;
                end
            end
            x(m,:) = x(m,:)+v(m,:);
        end
        if abs(globalBest_faval)<E0
            break
        end
        k = k+1;
    end
    globalBest_favals(v0) = globalBest_faval;
    globalBest_xs(v0,:) = globalBest_x;
    v0
end
plot(1:v0, globalBest_favals, 'b')
hold on
plot(1:v0, globalBest_xs(:,1), 'm')
plot(1:v0, globalBest_xs(:,2), 'r')
plot(1:v0, globalBest_xs(:,3), 'g')
hold off
toc;

```

### 第三题-角度规划

```
clear;
tic;

% 实际问题的参数设置
angle0 = 23.8447;
h = 3; % 假设杆的高度为 3 米
t0 = [220.5 235.5]; % 时间角向量，角度制
rad_t0 = (t0/360)*(2*pi); % 时间角向量，弧度制

% 设置算法参数
E0 = 0; % 允许误差
maxNum = 5000; % 粒子最大迭代次数
narvs = 3; % 目标函数的自变量个数
particleSize = 100; % 粒子群规模
c1 = 2; % 每个粒子的个体学习因子，也称加速常数
c2 = 2; % 每个粒子的社会学习因子，也称加速常数
w = 0.6; % 惯性因子
vMax = 27; % 粒子的最大飞行速度，单位“度/次”

for v0 = 1:90
    % 初始化粒子群状态
    x = zeros(particleSize, narvs);
    x(:,1) = 180*(rand(particleSize,1)-0.5); % 粒子的所在维度，单位“度”
    x(:,2) = 360*(rand(particleSize,1)-0.5); % 粒子的所在经度，单位“度”
    x(:,3) = 365*rand(particleSize,1); % 粒子的所在经度，单位“度”
    v = zeros(particleSize, narvs);
```

```

    v(:,1) = v0*rand(particleSize,1); % 粒子所在的维度的飞行速度，单位“度/次”

    v(:,2) = 2*v0*rand(particleSize,1); % 粒子所在的经度的飞行速度，单位“度/次”

    for m = 1:particleSize
        solar = ((23.5*sin(((2*pi)/365)*x(m,3)))/360)*(2*pi); % 由题目已知计算
        rad_x = (x(m,1)./360).*(2*pi); % 将纬度角度制的 x 转换为弧度制的 rad_x
        t = t0 + (x(m,2)-120); % 计算各个维度所对应的实际的时间角
        rad_t = (t./360).*(2*pi); % 将 x 的经度对应的时间角转换为弧度制 rad_t
        sin_h =
sin(rad_x(1)).*sin(solar)+cos(rad_x(1)).*cos(solar).*cos(rad_t
);
        cos_h = sqrt(1-sin_h.^2);
        cos_dir1 = (sin(solar)-
sin_h(1)*sin(x(m,1)))/(cos_h(1)*cos(x(m,1)));
        cos_dir2 = (sin(solar)-
sin_h(2)*sin(x(m,1)))/(cos_h(2)*cos(x(m,1)));
        y(m) = abs(abs((acos(cos_dir1)/(2*pi))*360-
(acos(cos_dir2)/(2*pi))*360)-angle0);
    end
    personalBest_x = x;
    personalBest_faval = y;
    [globalBest_faval m] = min(personalBest_faval);
    globalBest_x = personalBest_x(m,:);
    k = 1;
    while k<=maxNum

```

```

        for m = 1:particleSize
            solar =
((23.5*sin(((2*pi)/365)*x(m,3)))/360)*(2*pi); % 由题目已知计算
            rad_x = (x(m,1)./360).*(2*pi); % 将纬度角度制的 x 转换
为弧度制的 rad_x
            t = t0 + (x(m,2)-120); % 计算各个维度所对应的实际的时间角
            rad_t = (t./360).*(2*pi); % 将 x 的经度对应的时间角转换为弧度制 rad_t
            sin_h =
sin(rad_x(1)).*sin(solar)+cos(rad_x(1)).*cos(solar).*cos(rad_t
);
            cos_h = sqrt(1-sin_h.^2);
            cos_dir1 = (sin(solar)-
sin_h(1)*sin(x(m,1)))/(cos_h(1)*cos(x(m,1)));
            cos_dir2 = (sin(solar)-
sin_h(2)*sin(x(m,1)))/(cos_h(2)*cos(x(m,1)));
            y(m) = abs((acos(cos_dir1)/(2*pi))*360-
(acos(cos_dir2)/(2*pi))*360);
            if y(m)<personalBest_faval % 判断当前位置是否是历史上
最佳位置
                personalBest_faval(m) = y(m);
                personalBest_x(m,:) = x(m,:);
            end
        end
        [globalBest_faval m] = min(personalBest_faval);
        globalBest_x = personalBest_x(m,:);
        for m = 1:particleSize % 更新粒子群里每个个体的最新位置
            v(m,:) = w*v(m,:)+c1*rand*(personalBest_x(m,:)-
x(m,:))+c2*rand*(globalBest_x-x(m,:));

```

```

        for n = 1:narvs % 判断粒子的飞行速度是否超过最大飞行速度
            if v(m,n)>vMax
                v(m,n) = vMax;
            elseif v(m,n)<-vMax
                v(m,n) = -vMax;
            end
        end
        x(m,:) = x(m,:)+v(m,:);
    end
    if abs(globalBest_faval)<E0
        break
    end
    k = k+1;
end
globalBest_favals(v0) = globalBest_faval;
globalBest_xs(v0,:) = globalBest_x;
v0
end
plot(1:v0, globalBest_favals, 'b')
hold on
plot(1:v0, globalBest_xs(:,1), 'm')
plot(1:v0, globalBest_xs(:,2), 'r')
plot(1:v0, globalBest_xs(:,3), 'g')
hold off
toc;

```

第四问-三维转换二维

```

clear;
tic;

```

```

% 设置杆的坐标位置
points1 = zeros(100, 3);
points1(:,1) = 0;
points1(:,2) = 0;
points1(:,3) = transpose(linspace(0,2));
% 设置影子的坐标位置
shadow_len = 2;
shadow_x = 1.958;
points2 = zeros(100, 3);
points2(:,1) = transpose(linspace(0,shadow_x));
points2(:,2) = transpose(linspace(0,sqrt(shadow_len^2-
shadow_x^2)));
points2(:,3) = 0;
% 设置中线的位置
points3 = zeros(100, 3);
points3(:,1) = 0;
points3(:,2) = transpose(linspace(0,-1.5));
points3(:,3) = 0;
R = 3.5;
D = 0.1;
angle1 = 0.3218; %  $\phi$ 
angle2 = -1.3798; %  $\theta$ 
T = [-sin(angle2) -cos(angle2)*sin(angle1) -
cos(angle2)*cos(angle1) 0; cos(angle2) -
sin(angle2)*sin(angle1) -sin(angle2)*cos(angle1) 0; 0
cos(angle1) -sin(angle1) 0; 0 0 R 1];

for k = 1:size(points1,1)
    xyzc = [points1(k,1) points1(k,2) points1(k,3) 1]*T;

```



```

    x0(k) = xyzc(1);
    y0(k) = xyzc(2);
    z0(k) = xyzc(3);
    c0(k) = xyzc(4);
end
xe = (x0./z0).*D;
ye = (y0./z0).*D;
plot(xe, ye, 'b')
hold on
for k = 1:size(points2,1)
    xyzc = [points2(k,1) points2(k,2) points2(k,3) 1]*T;
    x0(k) = xyzc(1);
    y0(k) = xyzc(2);
    z0(k) = xyzc(3);
    c0(k) = xyzc(4);
end
xe = (x0./z0).*D;
ye = (y0./z0).*D;
plot(xe, ye, 'k')
for k = 1:size(points3,1)
    xyzc = [points3(k,1) points3(k,2) points3(k,3) 1]*T;
    x0(k) = xyzc(1);
    y0(k) = xyzc(2);
    z0(k) = xyzc(3);
    c0(k) = xyzc(4);
end
xe = (x0./z0).*D;
ye = (y0./z0).*D;
plot(xe, ye, 'r')
hold off

```

```
toc;
```

#### 第四问-最终的规划

```
clear;
tic;

% 导入数据
points = xlsread('附件 1-4.xlsx', 1, 'B2:C22');
l0 = transpose(sqrt(points(:,1).^2+points(:,2).^2));

% 实际问题的参数设置
h = 2; % 假设杆的高度为 2 米
days = datenum([2015 4 18])-datenum([2015 3 21]); % 由题目已知计算
solar = ((23.5*sin(((2*pi)/365)*days))/(360)*(2*pi)); % 由题目已知计算
t0 = [133.5 133.75 134.25 134.75 135 135.5 136 136.25 136.75
137.25 137.75 138 138.5 139 139.25 139.75 140.25 140.5 141
141.5 142]; % 时间角向量，角度制
rad_t0 = (t0/360)*(2*pi); % 时间角向量，弧度制

% 设置算法参数
E0 = 0.05; % 允许误差
maxNum = 5000; % 粒子最大迭代次数
narvs = 2; % 目标函数的自变量个数
particleSize = 100; % 粒子群规模
c1 = 2; % 每个粒子的个体学习因子，也称加速常数
c2 = 2; % 每个粒子的社会学习因子，也称加速常数
w = 0.6; % 惯性因子
```

```

vMax = 27; % 粒子的最大飞行速度，单位“度/次”

for v0 = 1:90
    % 初始化粒子群状态
    x = zeros(particleSize, narvs);
    x(:,1) = 180*(rand(particleSize,1)-0.5); % 粒子的所在维度，单位“度”
    x(:,2) = 360*(rand(particleSize,1)-0.5); % 粒子的所在经度，单位“度”
    v = v0*rand(particleSize,narvs); % 粒子的飞行速度，单位“度/次”

    for m = 1:particleSize
        rad_x = (x(m,1)./360).*(2*pi); % 将纬度角度制的 x 转换为弧度制的 rad_x
        t = t0 + (x(m,2)-120); % 计算各个维度所对应的实际的时间角
        rad_t = (t./360).*(2*pi); % 将 x 的经度对应的时间角转换为弧度制 rad_t
        sin_h =
sin(rad_x(1)).*sin(solar)+cos(rad_x(1)).*cos(solar).*cos(rad_t
);
        cos_h = sqrt(1-sin_h.^2);
        l = h.*(cos_h./sin_h);
        y(m) = sum(abs(l-l0));
    end
    personalBest_x = x;
    personalBest_faval = y;
    [globalBest_faval m] = min(personalBest_faval);
    globalBest_x = personalBest_x(m,:);
    k = 1;

```

```

while k<=maxNum
    for m = 1:particleSize
        rad_x = (x(m,1)./360).*(2*pi); % 将纬度角度制的 x 转换为弧度制的 rad_x
        t = t0 + (x(m,2)-120); % 计算各个维度所对应的实际的时间角
        rad_t = (t./360).*(2*pi); % 将 x 的经度对应的时间角转换为弧度制 rad_t
        sin_h =
sin(rad_x(1)).*sin(solar)+cos(rad_x(1)).*cos(solar).*cos(rad_t
);
        cos_h = sqrt(1-sin_h.^2);
        l = h.*(cos_h./sin_h);
        y(m) = sum(abs(l-l0));
        if y(m)<personalBest_faval % 判断当前位置是否是历史上最佳位置
            personalBest_faval(m) = y(m);
            personalBest_x(m,:) = x(m,:);
        end
    end
    [globalBest_faval m] = min(personalBest_faval);
    globalBest_x = personalBest_x(m,:);
    for m = 1:particleSize % 更新粒子群里每个个体的最新位置
        v(m,:) = w*v(m,.)+c1*rand*(personalBest_x(m,.)-
x(m,))+c2*rand*(globalBest_x-x(m,));
        for n = 1:narvs % 判断粒子的飞行速度是否超过最大飞行速度
            if v(m,n)>vMax
                v(m,n) = vMax;
            elseif v(m,n)<-vMax

```

```

        v(m,n) = -vMax;
    end
end
x(m,:) = x(m,:)+v(m,:);
end
if abs(globalBest_faval)<E0
    break
end
k = k+1;
end
globalBest_favals(v0) = globalBest_faval;
globalBest_xs(v0,:) = globalBest_x;
v0
end
plot(1:v0, globalBest_favals, 'b')
hold on
plot(1:v0, globalBest_xs(:,1), 'm')
plot(1:v0, globalBest_xs(:,2), 'r')
hold off
toc;

```