

太阳影子定位的优化模型

摘要

太阳影子定位技术是通过分析视频中物体的太阳影子的变化来确定视频拍摄时间和地点的一种重要方法。本文通过几何分析,建立非线性最优化模型,利用粒子群算法进行求解,得到了精度较高且符合实际的结果。同时利用 Sketchup 软件对视频进行分析,提取数据对第四问及附加问进行求解并得到比较合理的结果。

对于问题一,根据几何关系得影长的决定因素为太阳高度角,建立空间直角坐标系,通过空间几何变换,得出太阳高度角和经纬度以及赤纬的关系,进而得到了天安门广场直杆的影子长度变化曲线。在北京时间 12 时 12 分影长达到最短为 3.6624m 。

对于问题二,本文将经纬度问题转化为一个非线性规划问题,采用粒子群算法对此优化问题进行求解。问题中杆长,当地经纬度均为未知参数,并且由于数据中给出的 y 轴方向未知,增加 y 轴与正南方向夹角为待求参数。利用美国国家可再生能源研究室提供的工具对太阳方位角、太阳高度角进行高精度计算,通过空间几何分析,以代入参数计算得到的影子各顶点坐标与附件 1 数据坐标距离和最小,建立非线性优化模型。考虑到参数对目标影响未知以及计算机使用最小二乘法初始搜索值难以确定,本文采用粒子群算法对解空间进行分区求解。得出东经 109.7792° ,北纬 18.4760° ,即海南省三亚市为最可能的测量地点,目标函数值 0.0063 结果非常精确。同时得出东经 126.0161° ,北纬 23.4722° (太平洋某点)虽误差较海南更大,但仍在可接受的范围内,为可能的地点。在此基础上,考虑左手系以及球的对称性,得出印尼近海某点,印尼萨纳纳岛某点也为可能的直杆坐标测量地点,具体结果见表 2。

对于问题三,由于粒子群算法不受参数维数的限制,本文在第二问基础上增加测量日期作为待求参数进行求解。得出附件 2 数据最可能的测量地点在东经 79.8989° ,北纬 41.9236° (新疆阿克苏附近)。其他可能地点分布在新疆和田、印度洋、孟加拉近海等区域。附件 3 数据最可能的测量地点在东经 111.8776° ,北纬 24.7517° (广西贺州市附近)。其他可能地点分布在湖南永州,西伯利亚,澳大利亚,印度洋等地,具体结果见表 8。

对于问题四,本文利用空间三维建模中的小孔成像相关原理,用 Sketchup 软件建立坐标系,按两分钟为步长,提取视频中的影长及与坐标轴夹角。由于测量误差去掉两个坏值。在问题二的基础上确定杆长进行求解。得到最可能的地点为湖南邵阳和南海西沙群岛附近。对于附加问,固定杆长将提取的数据使用问题三的模型进行求解得出最可能的地点在中沙群岛,发现与问题四的结果相差较大,对造成误差的原因进行分析。

关键词: 太阳高度角 空间几何 粒子群算法 分区求解 Sketchup 软件

一、问题重述

如何确定视频的拍摄地点和拍摄日期是视频数据分析的重要方面，太阳影子定位技术就是通过分析视频中物体的太阳影子变化，确定视频拍摄的地点和日期的一种方法。

1. 建立影子长度变化的数学模型，分析影子长度关于各个参数的变化规律，并应用你们建立的模型画出 2015 年 10 月 22 日北京时间 9:00-15:00 之间天安门广场（北纬 39 度 54 分 26 秒, 东经 116 度 23 分 29 秒）3 米高的直杆的太阳影子长度的变化曲线。

2. 根据某固定直杆在水平地面上的太阳影子顶点坐标数据，建立数学模型确定直杆所处的地点。将你们的模型应用于附件 1 的影子顶点坐标数据，给出若干个可能的地点。

3. 根据某固定直杆在水平地面上的太阳影子顶点坐标数据，建立数学模型确定直杆所处的地点和日期。将你们的模型分别应用于附件 2 和附件 3 的影子顶点坐标数据，给出若干个可能的地点与日期。

4. 附件 4 为一根直杆在太阳下的影子变化的视频，并且已通过某种方式估计出直杆的高度为 2 米。请建立确定视频拍摄地点的数学模型，并应用你们的模型给出若干个可能的拍摄地点。

如果拍摄日期未知，你能否根据视频确定出拍摄地点与日期？

二、问题分析

问题一分析几何关系得出影长直接决定于太阳高度角。为了确定太阳高度角，建立适当的空间直角坐标系。通过空间直角坐标系中太阳光线所在向量 S 和该地与地心连线的向量夹角，来确定太阳高度角的大小。

其中太阳光线所在向量通过太阳赤纬确定，该地与地心连线的向量用当地纬度和时角表示。这样影长用纬度，时角和赤纬表示。其中对于确定地点和确定日期，纬度和赤纬是一个常数。最后把时角用时间表示出来，这样影长就是时间的函数。画出影子长度随时间的变化情况。

具体思路如下图所示：

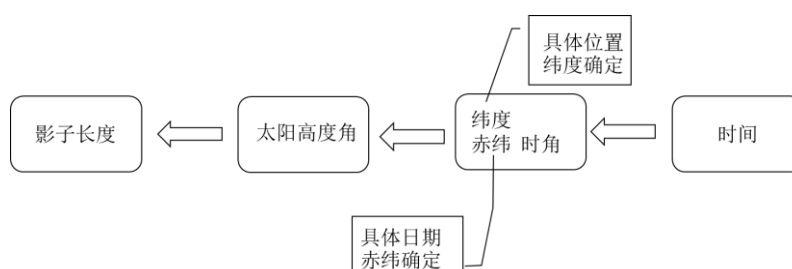


图 1 问题一思路图

问题二给出一组以地面为 xoy 面时影子顶点坐标数据， xy 轴的具体方向未知，因此设立参数为 y 轴正半轴与正南方向的夹角。问题一给出了太阳高度角的表达式，这只是粗略计算，用来定位会造成较大的误差。因此问题二在计算太阳高度角和方位角时引用美国国家利用可再生能源实验室的成果，把误差降到 $\pm 0.003^\circ$ ，大大提高定位精度。各设立目标函数使坐标的计算值与实际值距离最小，用粒子

群算法求解，得出各个参数，即可知直杆所处位置。

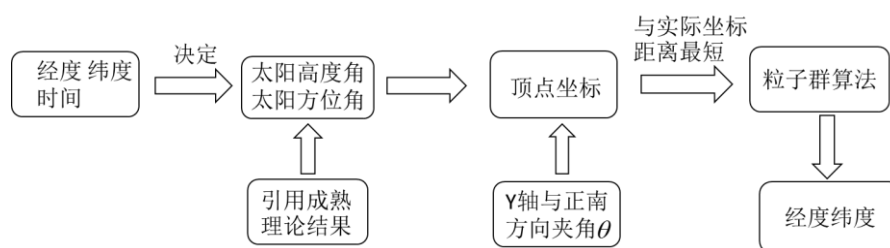


图 2 问题二思路图

问题三在问题二的基础上增加了参数 N , 利用问题二的模型进行求解。

问题四利用 Sketchup 软件，每两分钟对视频截一次屏，测量影子长度和影子与 x 轴夹角，进而得到影子顶端的坐标变化。在视频拍摄日期已知和杆高已知的条件下，应用问题二模型进行求解。附加问题中杆高已知，将其设为常量。利用问题三模型求解拍摄地点和日期，得出结果。

三、模型假设

1. 忽略赤纬在一天中的变化，一天中的赤纬不变
2. 地球是一个正球体
3. 本题所给数据真实可靠，具有客观性
4. 不考虑地球大气层对太阳高度角的影响

四、定义与符号说明

符号	定义
α	太阳高度角
δ	太阳赤纬
φ	纬度
E	经度
h	直杆长度
l	直杆影子长度
ω	时角
β	太阳方位角

五、问题（1）模型的建立与求解

5.1 名词解释

太阳高度角：对于地球上的某个地点，太阳高度角是指太阳光的入射方向和地平面之间的夹角，专业上讲太阳高度角是指某地太阳光线与通过

该地与地心相连的地表切线的夹角。

太阳赤纬：地球赤道平面与太阳和地球中心的连线之间的夹角。赤纬角以年为周期，在 $+23^{\circ}26'$ 与 $-23^{\circ}26'$ 的范围内移动，成为季节的标志。

时角：以当地正午时刻为 0° ，上午为负，每小时 -15° ，下午为正，每小时 $+15^{\circ}$ ，时角表示的是真太阳时，与时钟不同

真太阳时：北京时间一时差

5.2 北京时间转换成北京当地时间

北京时间并不是北京（东经 116.4° ）地方的时间，而是东经 120° 地方的地方时间。问题一给出的拍摄地点是北京天安门广场。在计算太阳高度角等问题时用的是当地时间。因此需要把北京时间换算成北京天安门广场的当地时间。北京天安门广场的经度为 $116^{\circ}23'29''$ 。利用换算公式把北京时间转换成北京当地时间。

$$t = T + \frac{116^{\circ}23'29'' - 120^{\circ}}{60} \times 4 \text{ (单位: 小时)} \quad (1)$$

5.3 太阳高度角的确定

不同太阳高度角对应不同的影长。分析影长和太阳光线角度 α 间关系，画出他们之间的几何关系图：如图所示：

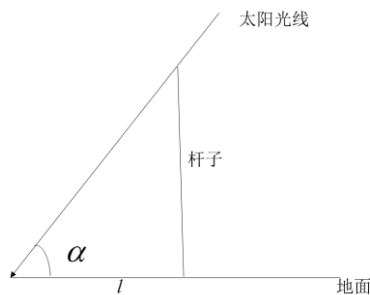


图 3 影长和太阳高度角几何关系

设太阳高度角为 α ，直杆长度为 h ，影子长度

$$l = h / \tan \alpha \quad (2)$$

由此可知确定直杆的影子长度关键在于确定的太阳高度角 α 。

为了确定太阳高度角的大小，分析太阳光线所在向量和当地位置间的关系，如下图所示：

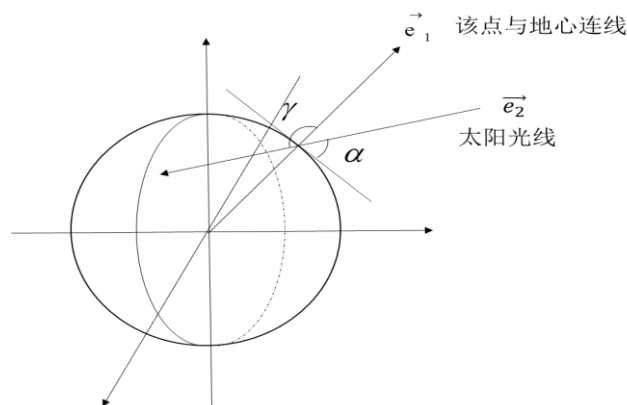


图 4 向量求解太阳高度角

太阳高度角由太阳光线所在向量和地心与该地连线的向量决定。设太阳光线所在向量为 \vec{e}_1 ，地心与该地连线为 \vec{e}_2 ， $\gamma = \langle \vec{e}_1, \vec{e}_2 \rangle$ 为两个向量夹角，太阳高度角为 α 则：

$$\gamma = \arccos \left(\frac{\vec{e}_1 \vec{e}_2}{|\vec{e}_1| |\vec{e}_2|} \right) \quad (3)$$

$$\alpha = \gamma - \frac{\pi}{2} = \arccos \left(\frac{\vec{e}_1 \vec{e}_2}{|\vec{e}_1| |\vec{e}_2|} \right) - \frac{\pi}{2} \quad (4)$$

为了确定该点与地心连线的向量和光线的向量，建立空间直角坐标系，方法如下：地球绕地轴自转，绕太阳做椭圆轨道运动。由于黄赤交角的存在，一年中的每一天太阳直射点所在纬度是不尽相同的。

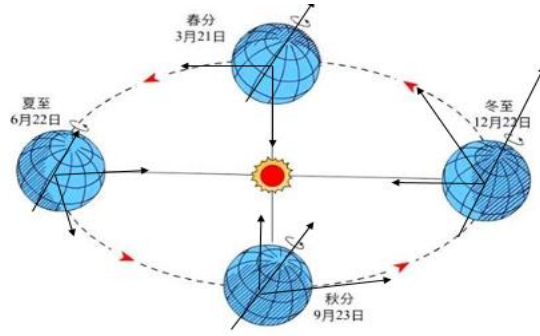


图 5 不同季节坐标轴位置

然而由基本假设，公转只影响不同日期下各个地点的不同时刻太阳高度角。即太阳直射点位置的不同。不妨假设地球位置固定，让地球绕地轴做自转运动，而太阳直射点在南北回归线之间摆动。

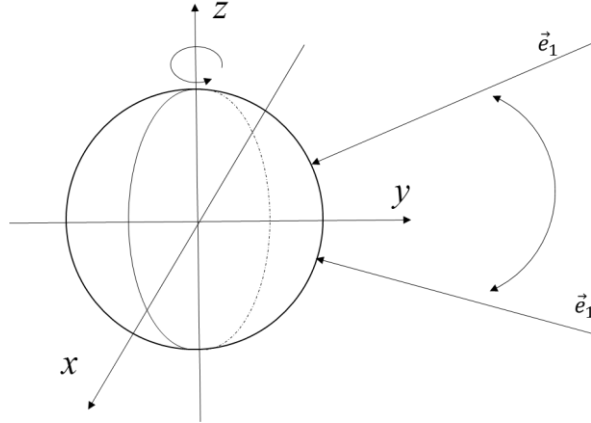


图 6 一年中太阳直射点的摆动

基于此考虑，以地轴为 z 轴，以赤道所在平面为 xoy 面，建立空间直角坐标系。使得太阳光线的照射方向始终与 yoz 面平行。

设 ω 为时角，当地纬度为 φ ，如下图所示：

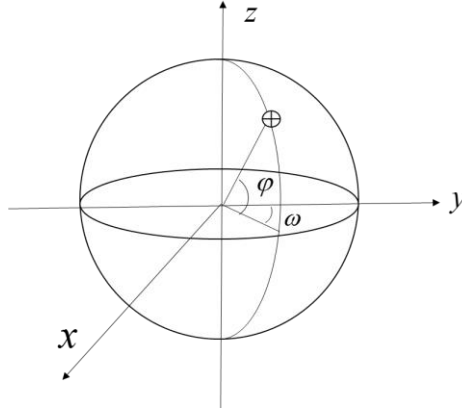


图 7 用太阳方位角时角表示坐标

则当地坐标为 (x, y, z) 为:

$$\begin{aligned} x &= R \cos \varphi \cos \omega \\ y &= R \cos \varphi \sin \omega \\ z &= R \sin \varphi \end{aligned} \quad (5)$$

该点与地心连线的向量为:

$$\vec{e}_1 = (x, y, z) = (R \cos \varphi \cos \omega, R \cos \varphi \sin \omega, R \sin \varphi) \quad (6)$$

确定太阳光线所在向量, 需要求得太阳赤纬的大小。任意一天的太阳赤纬 δ 的计算公式为:

$$\delta = (A - B \cos b + C \sin b - D \cos 2b + E \sin 2b - F \cos 3b + G \sin 3b)(180 / \pi) \quad (7)$$

其中,

$$b = 2\pi(N - 1) / 365 \quad (8)$$

N 为日数, 自每年 1 月 1 日开始计算

$A = 0.006918$, $B = 0.399912$, $C = 0.070257$, $D = 0.006758$, $E = 0.000907$
 $F = 0.002697$, $G = 0.00148$

根据太阳赤纬, 可以确定太阳光线的方向, 把光线方向分解到空间直角坐标系上, 单位化后得到太阳光线向量为:

$$\vec{e}_2 = (0, \cos \delta, \sin \delta) \quad (9)$$

综上, 可求得太阳高度角为:

$$\sin \alpha = \sin \delta \sin \varphi + \cos \delta \cos \varphi \cos \omega \quad (10)$$

5.4 模型结果

所以, 影长 l 的表达式为:

$$l = \frac{h}{\tan \left[\arcsin (\sin \delta \sin \varphi + \cos \delta \cos \varphi \cos \omega) \right]} \quad (11)$$

又因为

$$\omega = (t - 12) \times 15^\circ \quad (12)$$

影长又可以表示为：

$$l = h / \tan \left\{ \arcsin [\sin \delta \sin \varphi + \cos \delta \cos \varphi \cos (t - 12) \times 15^\circ] \right\} \quad (13)$$

这样求得任意时刻直杆影子长度，详情见附录[1]。画出得直杆影子长度随时间变化图为：

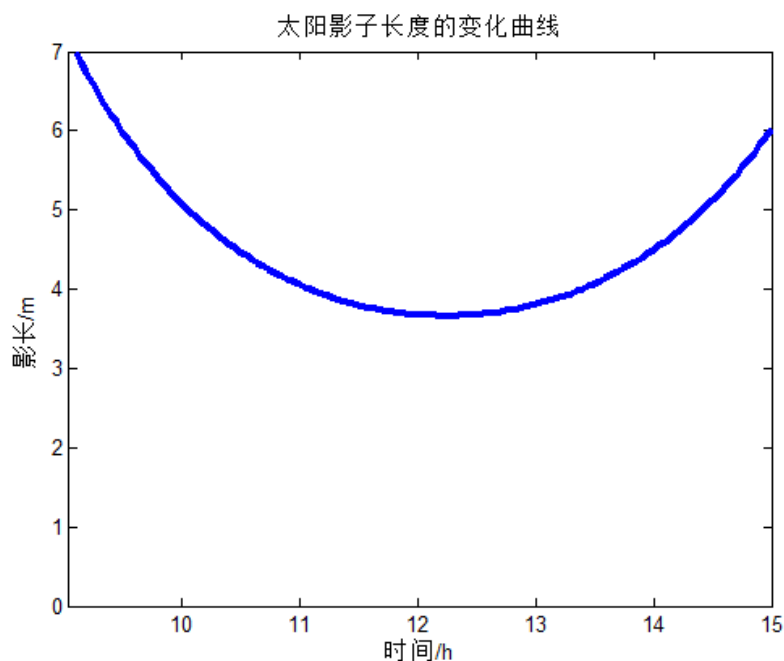


图 8 影长随时间变化图

求得北京时间 12 时 12 分影长最短，为 3.6624 m。

六、问题（2）模型建立与求解

6.1 xy 轴方向的确定参数 θ

附件二给出了 2014 年 4 月 18 日，以地面为 xoy 面，直杆底端为原点的直角坐标系中影子的顶点坐标。但是题目并没有给出 y 轴的具体指向，设 y 轴正方向与正南方向夹角为 θ 。定义如下图所示：

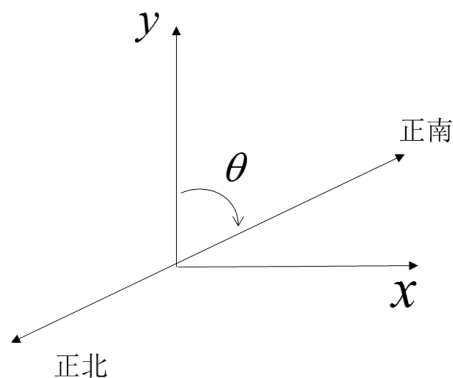


图 9 y 轴与正南方向夹角

设影长为 l ，再引入太阳方位角 β 来表示影子顶点坐标。太阳方位角指的是指太阳光线在地平面上的投影与当地经线的夹角，可近似地看作是竖立在地面上的直线在阳光下的阴影与正南方夹角。如下图所示：

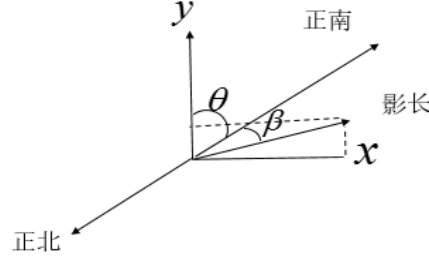


图 10

则影子顶点坐标为：

$$\begin{aligned} x &= l \sin(\beta + \theta) \\ y &= l \cos(\beta + \theta) \end{aligned} \quad (14)$$

6.2 太阳高度角和方向角的确定

查阅文献可知，目前对于太阳高度角和方向角的计算，误差最小的为 $\pm 0.01^\circ$ 。然而方向角的微小误差对太阳影子定位带来较大的影响。直接利用问题一求解出的太阳高度角的计算公式只能粗略地计算该测量点的位置。为了减少由于太阳高度角和方位角计算带来的误差，本文利用美国国家可再生能源实验室的研究成果，该成果能将太阳方位角和高度角的计算误差减少到 $\pm 0.003^\circ$ 。根据这一成果能大大提高定位精度。

利用研究成果，太阳高度角和方位角是纬度、精度和时间的函数，表示为 $\alpha(\varphi, E, t)$ ， $\beta(\varphi, E, t)$ 。这样太阳顶点的坐标又可以表示为：

$$\begin{aligned} x &= l \sin(\beta + \theta) = \frac{h}{\tan \alpha} \sin(\beta + \theta) \\ y &= l \cos(\beta + \theta) = \frac{h}{\tan \alpha} \cos(\beta + \theta) \end{aligned} \quad (15)$$

6.3 建立实际坐标和计算坐标距离最小的目标函数

根据以上分析，设影子顶点坐标为 $(x_i, y_i) i = 1, 2, \dots, n$ ，模型的计算结果和实际数据之间有一定的误差，设顶点实际坐标为 (x_i', y_i')

则计算值和实际值之间的差距为：

$$\sqrt{(x_i - x_i')^2 + (y_i - y_i')^2} \quad (16)$$

建立计算得出的顶点和实际顶点的距离和最小的目标函数：

$$\sum_{i=1}^n \sqrt{(x_i - x_i')^2 + (y_i - y_i')^2} = f(h, E, \varphi, \theta) \quad (17)$$

$$d = \min \sum_{i=1}^n \sqrt{(x_i - x_i')^2 + (y_i - y_i')^2} \quad (18)$$

6.4 利用粒子群算法求最优解

目标函数的求解为非线性最优化问题，求解使得计算得出的影子顶点位置与实际数据最为贴合的一组参数：杆高 h ，经度 E ，纬度 φ ， y 轴与正南方向夹角 θ 。

约束条件：

$$\begin{cases} \alpha > 0 \\ h > 0 \end{cases} \quad (18)$$

利用最小二乘法进行求解时必须给定一组参数初值以在周围进行搜寻。然而由于各个变量对于目标函数的影响尚不明确，人为给定一组初值可能导致算法陷入局部最优值，只能通过不断修改初始参数值来获得全局最优解。所以最小二乘法的求解存在很大的不足。

粒子群算法（particle swarm optimization, PSO）能够在解空间内进行全局搜索，当目标函数仅能提供极少搜索最优值的信息时，在其他算法无法辨别搜索方向的情况下，PSO 算法的粒子具有飞跃性的特点，使其能够跨过搜索平面上信息严重不足的障碍，求得全局最优目标值。因此本文采用 PSO 算法进行求解。

根据 PSO 算法的思想，人们决策过程中往往会综合两种重要的信息，第一种是他们自己的经验，即他们根据以前自己的尝试和经历，已经积累了一定的经验；第二种是他人的经验，即从周围人的行为获取知识，从中知道哪些选择是正确的，哪些选择是消极的。因此运用 PSO 求解时，每个粒子具有学习能力，粒子能够记住自己找到的最好位置，称之为局部最优。此外，还能记住其他个体所能找到的最好位置，称之为全局最优。

在一个 D 维的目标的搜索空间中，有 m 个粒子组成一个群体，其中第 i 个粒子位置表示为

$$S_i = (s_i^1, s_i^2, \dots, s_i^D) \quad (20)$$

记第 i 个粒子在 D 维搜索空间中的位置是 S_i 。每个粒子的位置就是一个潜在解，将 S_i 带入目标函数就可以计算出其适应值，根据适应值的大小衡量其优劣。粒子个体经历过的最好位置记为

$$P_i = (p_i^1, p_i^2, \dots, p_i^D) \quad (21)$$

整个群体所有粒子经历过的最好位置记为

$$P_g = (p_g^1, p_g^2, \dots, p_g^D) \quad (22)$$

粒子 i 的速度记为：

$$V_i = (v_i^1, v_i^2, \dots, v_i^D) \quad (23)$$

粒子群算法采用下列公式对粒子所在的位置不断更新（单位时间 1）：

$$v_i^d = \varepsilon v_i^d + c_1 r_1 (p_i^d - s_i^d) + c_2 r_2 (p_g^d - s_i^d) \quad (24)$$

$$s_i^d = s_i^d + \sigma v_i^d \quad (25)$$

其中， $i=1,2,\dots,m$ ； $d=1,2,\dots,D$ ； ε 非负，称为惯性因子；加速度 c_1 和 c_2 是非负场数， r_1 和 r_2 是 $[0,1]$ 范围内变换的随机数， σ 是约束因子，目的是控制速度的权重。

此外，记粒子的飞行速度 $v_i^d \in [-v_{\max}, v_{\max}]$ ，即粒子的飞翔速度被一个最大速度所限制。如果当前时刻粒子在 D 维的速度更新后超过该维的最大飞行速度，则将当前时刻该维的速度限制在 v_{\max} 。

PSO 算法本身适用于无约束最优化，然而实际问题中太阳高度角和杆高都必须大于 0。因此本文规定，当待求参数不符合实际，即当 $\alpha \leq 0$ 或 $h \leq 0$ 时将目标函数值设为无穷大，此时其他粒子不会向其靠拢，以排除不符合实际的解。

根据文献^[4]，模型系统参数的取值如下：粒子数 $m=100$ ，惯性因子 $\varepsilon=0.6$ ， $c_1=2.8$ ， $c_2=1.3$ 。

迭代终止条件根据问题设定，一般达到预定最大迭代次数或粒子群目前为止搜索到的最优位置满足目标函数的最小容许误差。

算法流程图如下：

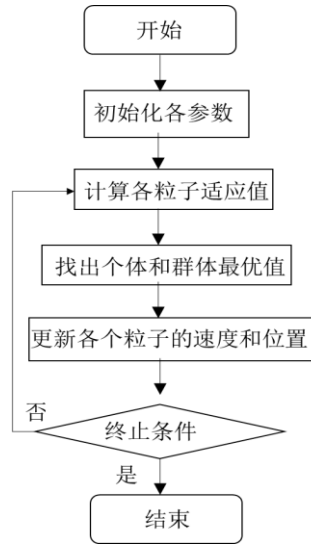


图 11 粒子群算法流程图

因为粒子群算法局部搜索能力较差，本文先将经度与纬度在其定义域各自分为 4 等份，共组成 16 个区域，再在各个区域分别用粒子群算法求解。

导入附件一数据，求得模型结果见表 1。

表 1 分区搜索的结果

搜索区间		f	h	E	φ	θ
经度区间	纬度区间					
[-180,-90]	[-90,-45]	∞	9.4427	150.7904	68.8302	212.0300
[-180,-90]	[-45,0]	∞	6.2595	-95.5969	-19.6798	160.8970
[-180,-90]	[0,45]	∞	3.2013	-110.0718	17.8350	159.9550
[-180,-90]	[45,90]	1.1300	0.0590	-137.5000	79.9743	96.6763
[-90,0]	[-90,-45]	0.0063	1.9646	109.7792	18.4760	524.8648
[-90,0]	[-45,0]	0.1658	1.1493	125.7500	23.1585	522.4860
[-90,0]	[0,45]	0.1655	1.1291	126.2168	23.4721	162.4919
[-90,0]	[45,90]	0.1655	1.1291	126.2500	23.4720	162.4919
[0,90]	[-90,-45]	0.1687	1.2216	123.8047	22.4489	162.7227
[0,90]	[-45,0]	0.1656	1.1392	125.7830	23.2918	162.4713
[0,90]	[0,45]	0.0063	1.9646	109.7688	18.4760	164.8648
[0,90]	[45,90]	0.1655	1.1291	126.0098	23.4721	162.4917
[90,180]	[-90,-45]	0.2511	0.9187	131.7500	26.6096	161.9140
[90,180]	[-45,0]	0.0063	1.9646	109.7541	18.4761	164.8648
[90,180]	[0,45]	0.1655	1.1291	126.0161	23.4722	162.4918
[90,180]	[45,90]	0.1655	1.1291	126.0073	23.4722	162.4918

6.5 误差分析与结果选取

附件中给出 21 个顶点数据坐标，以米为单位保留到小数点后四位，最后一位为估读位，数据的精度为 $1mm$ 。假设数据测量准确，则数据的误差出现在估读位。估读位的最大误差为 $0.0009m$ 。则计算顶点位置与实际顶点位置的距离和最大为：

$$\max d = 21 \times \sqrt{0.0009^2 + 0.0009^2} = 0.0267 \quad (27)$$

当给出的解与实际点间的距离和小于 0.0267 时，认为得到的解非常合理，具有很大的可信度。

当解与实际点的距离和大于 0.0267 时，距离越大可信度越低。但是由于模型本身存在的误差，精度要在 0.0267 内比较困难。本文认为误差在 0.2 以内时，解都具有一定的参考价值。

根据以上分析，本文选取误差在 0.2 以内的几组数据，有些数据指向的位置接近，选取误差最小的一点作为代表。结果如下表所示：

表 2 可能测量点的位置

f	h	E	φ	θ	Location
0.0063	1.9646	109.7792	18.4760	164.8648	海南三亚
0.1655	1.1291	126.0161	23.4722	162.4918	太平洋

以上是右手系下得出的结果，由于球的对称性，以当天赤纬为对称轴，如下图所示：

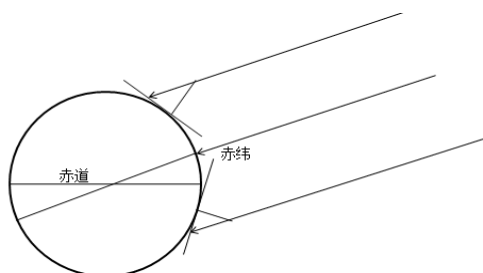


图 12 太阳光线关于赤纬对称图

得到上述结果的关于赤纬对称的另外一组可行解。

表 3 可能的测量点位置

f	h	E	φ	θ	Location
0.0063	1.9646	109.7792	2.6152	164.8648	印尼近海
0.1655	1.1291	126.0161	-2.3810	162.4918	印尼萨纳纳岛

根据以上结果，直杆位置在海南省三亚市的解精度最高，和实际顶点的拟合情况非常好， $d = 0.0063 < 0.0267$ ，解的可信度最高，因此直杆定位于海南省三亚市的可能性最大。由于合理化的图形和数据非常贴合，之后不再给出拟合图。图像如下：

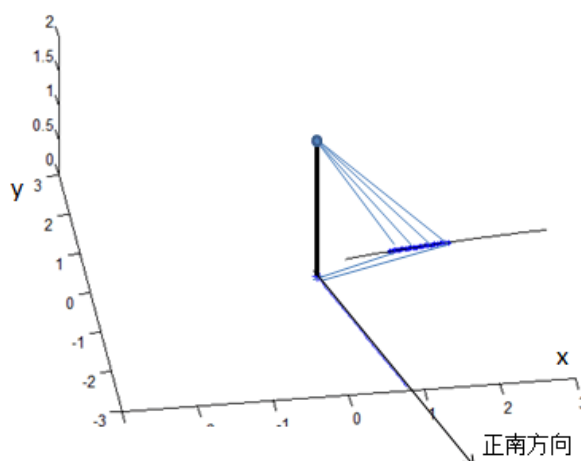


图 13 实际值和计算值的拟合效果图

七、问题三模型与求解

问题三要求计算直杆拍摄的地点和日期，在问题二的基础上增加了一个待求参数 N ， N 为日数，自每年 1 月 1 日开始计算。求解问题二给出的目标函数，利用粒子群算法进行分区求解。

附件 2 的求解结果：

表 4 分区求得的测量点位置

f	h	E	φ	θ	N
4.3794	0.4418	-162.3372	-90.0000	24.5442	350
4.3786	0.3906	-178.1828	-90.0000	10.9716	329
2.3534	0.4417	-178.8655	90.0000	29.6584	165
2.3528	0.4296	-179.8612	90.0000	30.2557	157
3.6745	0.1983	-77.2500	-78.3945	108.8630	334
∞	3.5303	-31.0871	-8.5440	124.2235	338
0.0063	2.0534	79.8989	41.9236	161.2652	170
0.9723	0.1622	-31.2558	69.9734	245.8931	151
0.0560	1.8266	80.1376	15.4995	149.6037	283
0.0604	1.8514	87.4418	12.4763	148.1514	53
0.0374	1.8099	80.7502	25.1861	153.1645	257
0.0053	2.0264	79.0435	41.0575	160.8119	153
0.0553	1.8138	80.4743	17.2795	149.5347	279
0.0819	2.0375	83.2854	-1.6985	144.9466	341
2.3547	0.4335	146.9986	90.0000	424.8976	184
0.0155	1.9665	81.7639	37.9729	158.9464	211

按照问题 2 的误差分析进行结果筛选。

图 5 可能的测量点位置

f	h	E	φ	N	Location
0.0063	2.0534	79.8989	41.9236	170	新疆阿克苏
0.0155	1.9665	81.7639	37.9729	211	新疆和田
0.0604	1.8514	87.4418	12.4763	53	印度洋
0.0819	2.0375	83.2854	-1.6985	341	印度洋
0.0063	2.0534	79.8989	4.9250	170	孟加拉近海
0.0155	1.9665	81.7639	-0.5531	211	印度洋
0.0604	1.8514	87.4418	-33.4397	53	印度洋
0.0819	2.0375	83.2854	-43.3475	341	印度洋

附件 3 的求解结果：

表 6 分区求得的测量点位置

f	h	E	φ	θ	N
11.1670	0.3739	-94.5113	-72.2678	198.1242	0
11.1638	0.3852	-98.5133	-72.3327	196.7078	345
0.9700	1.5594	-180.0000	78.7344	120.2665	194
0.9465	0.1411	-127.8328	88.0546	72.5189	87
11.1755	0.5866	-107.2500	-74.3911	187.9285	353
11.9630	1.6052	24.6654	-90.0000	-43.0602	362
0.2496	3.1660	130.7884	67.3734	163.2217	158
3.0960	1.6170	10.9205	90.0000	293.2332	175
11.2558	0.5587	105.7454	-57.6840	37.6726	172
11.2558	0.5489	105.4148	-57.8510	397.4813	171

0.0174	3.1552	108.2447	24.7517	190.7068	345
0.1592	2.4116	121.6649	59.7451	177.4159	94
11.2558	0.5489	105.4127	-57.8511	37.4813	171
0.0119	3.0812	111.8776	26.6037	190.4926	10
0.0186	3.1797	108.6995	24.2590	190.8071	350

按照问题 2 的误差分析进行结果筛选。

表 7 可能的测量点位置

f	h	E	φ	N	Location
0.0174	3.1552	108.2447	24.7517	345	广西
0.0119	3.0812	111.8776	26.6037	10	湖南永州
0.2496	3.1660	130.7884	67.3734	158	西伯利亚
0.15924	2.41155	121.6649	59.74514	94	西伯利亚
0.0174	3.1552	108.2447	-70.6149	345	南极洲
0.0119	3.0812	111.8776	-70.7523	10	南极洲
0.2496	3.1660	130.7884	-22.0100	158	澳大利亚
0.15924	2.41155	121.6649	-48.9549	94	印度洋

八、问题（4）模型与求解

8.1 数据准备

附件 4 视频给出了 8:54~9:47 直杆影子长度和位置的变化。问题 4 核心在于从视频中读取各个时刻的影长。在视频图片中以某一参照物建立三维空间直角坐标系，测量影子与 x 轴夹角的大小。将 40 分钟的视频每 2 分钟截屏一次，利用 Sketchup 设计软件，测量视频中影子长度和影子与坐标轴夹角。

该软件利用小孔成像原理进行三维空间建模，根据图像中直杆的长度计算影子长度和影子与坐标轴夹角，具体原理件文献[6]。Sketchup 软件的具体使用方法见支撑材料视频。

影子与坐标系的位置如图所示：



图 14 在地面建立坐标轴

测量出的 20 组影长和影子与坐标轴夹角如下表所示：

表 7 影长与夹角测量数据结果

时间	影长	与 x 轴夹角	时间	影长	与 x 轴夹角
8:56	2.3865	12.7	9:16	2.1032	12.3
8:58	2.3396	13.1	9:18	2.0672	11.9
9:00	2.4219	15.0	9:20	2.0258	12.1
9:02	2.4561	16.7	9:22	2.0127	12.1
9:04	2.3500	14.9	9:24	1.9839	11.7
9:06	2.3267	15.1	9:26	1.9462	11.6
9:08	2.2597	14.1	9:28	1.9200	11.7
9:10	2.2006	12.8	9:30	1.8905	11.6
9:12	2.2027	13.9	9:32	1.8507	11.0
9:14	2.1369	12.7	9:34	1.8162	10.9

8.2 模型建立与求解

利用测量所得影长和夹角，可以得到影子顶点的坐标。本问在问题二的基础上已知杆的长度，将问题二中的杆长作为已知参数，利用问题二模型代入数据求解即可。模型结果如下：

表 8 分区求得的测量点位置

f	E	φ	θ
4.8050	-179.9999	-20.6761	101.0624
4.8050	-180.0000	-20.6762	101.0624
3.5976	-179.7500	66.8542	234.4877
3.5976	179.7500	66.8542	234.4877
∞	-81.0160	-86.8556	199.2132
∞	-68.2496	-2.1990	359.4015
3.5976	-179.7500	66.8542	234.4877
51.4612	-47.0000	90.0000	106.6799
0.5548	111.2500	27.3979	352.3003
0.5548	111.2500	27.3979	-7.6997
0.5548	111.2500	27.3979	352.3003
0.5548	111.2500	27.3979	352.3003
0.5548	111.2500	27.3979	-7.6997
0.5548	111.2500	27.3979	-7.6997
0.5548	111.2500	27.3979	-7.6997
0.5548	111.2500	27.3979	352.3003

从以上结果中筛选在合理误差范围内的解，可行的解如下表所示：

表 8 可能的测量点位置

f	E	φ	θ	Location
0.5548	111.2500	27.3979	352.3003	湖南邵阳
0.5548	111.2500	16.5171	352.3003	南海西沙群岛附近

如果拍摄日期未知，在问题三的基础上已知杆的长度，将问题三中的杆长作为已知参数，利用问题三模型代入数据求解即可。模型最有可能的结果如下：

表 9 可能的测量点位置

f	E	φ	θ	N	Location
0.4508	114.6584	16.1530	333.5262	261	中沙群岛

当拍摄日期已知的情况下结果为湖南省与南沙群岛两个可行地点，而对于同一数据在拍摄日期未知的情况下结果为中沙群岛。显然这两问计算出的结果明显不同。然而，对视频中的数据进行测量时，会产生测量误差，而产生的测量误差对于最后给出的结果有很大的影响。因此，得出的结果有明显差异，是属于正常情况。

九、 模型评价与推广

本文解决的是利用太阳影子定位问题，利用几何知识建立解物理模型，结合智能算法解决实际问题。考虑多方面因素，联系实际，但也存在着不足之处。

9.1 模型优点：

1. 本文的模型，在实际的物理模型的基础上，利用空间几何关系进行推导，结合数学算法进行计算，得出结果。
2. 问题二的模型，在求太阳方位角和高度角的过程中，引用官方研究成果，相当精确的计算出太阳高度角和太阳方位角，在很大程度上提升了模型的精度。
3. 本文的模型，利用 PSO 算法进行求解，将复杂的不定初值的非线性目标优化问题，转化为全局最优问题，搜索计算，得出结果,降低了计算的复杂度。
4. 使用 PSO 算法的同时，在计算全局最优的基础上，对各个维度进行分区计算，得到局部最优解，在满足精度的范围内选点，考虑全面。

9.2 模型缺点：

1. 对于不同的年份，地球和太阳的位置关系不是一成不变的，本文在考虑计算日期时候，没有对具体的哪一年进行分析。
2. 在利用视频分析影长时会产生一定的误差，模型精度降低。

本文的模型，在研究太阳影子定位问题上，具有明显的适用性。可以利用模型，设计出具体太阳影子定位仪，让大众受益。

十、 参考文献

- [1] 贺晓雷，于贺军，李建英.太阳方位角的公式求解，太阳能学报，2008.1，卷期号：起止页码，出版年。
- [2] 百度百科，太阳高度角，<http://baike.baidu.com/view/86609.htm>，2015 年 9 月 11
- [3] 百度百科，太阳方位角，www.baidu.com/baidu?&ie=utf-8&word=太阳方位角，2015 年 9 月 11 日
- [4] 卓金武，魏永生，《MATLAB》在数学建模中的应用，北京航空航天大学出版社，2011
- [5] Afshin Andreas, Ibrahim Reda, Solar Position Algorithm for Solar Radiation Applications, 2008

[6]360 个人图书馆，如何测算图片中物体的实际尺寸，
http://www.360doc.com/content/14/0809/21/14106735_400656838.shtml，
 2015 年 9 月 12 日

十一、附件

附录 1：问题一中影长随时间

影长	时间	影长	时间	影长	时间
7.176572	8.809211	4.027663	10.80921	3.828114	12.80921
7.020312	8.859211	3.9967	10.85921	3.849576	12.85921
6.871017	8.909211	3.967232	10.90921	3.872415	12.90921
6.728258	8.959211	3.939233	10.95921	3.89665	12.95921
6.591645	9.009211	3.912682	11.00921	3.922298	13.00921
6.460817	9.059211	3.887557	11.05921	3.949379	13.05921
6.335446	9.109211	3.863839	11.10921	3.977917	13.10921
6.215227	9.159211	3.841509	11.15921	4.007933	13.15921
6.099882	9.209211	3.82055	11.20921	4.039453	13.20921
5.989153	9.259211	3.800948	11.25921	4.072505	13.25921
5.882801	9.309211	3.782687	11.30921	4.107115	13.30921
5.780607	9.359211	3.765754	11.35921	4.143317	13.35921
5.682365	9.409211	3.750139	11.40921	4.181142	13.40921
5.587888	9.459211	3.735829	11.45921	4.220626	13.45921
5.496999	9.509211	3.722814	11.50921	4.261807	13.50921
5.409536	9.559211	3.711087	11.55921	4.304725	13.55921
5.325347	9.609211	3.700639	11.60921	4.349423	13.60921
5.244291	9.659211	3.691463	11.65921	4.395947	13.65921
5.166237	9.709211	3.683553	11.70921	4.444347	13.70921
5.091061	9.759211	3.676903	11.75921	4.494674	13.75921
5.018651	9.809211	3.67151	11.80921	4.546986	13.80921
4.948899	9.859211	3.66737	11.85921	4.601341	13.85921
4.881706	9.909211	3.664481	11.90921	4.657804	13.90921
4.816979	9.959211	3.66284	11.95921	4.716443	13.95921
4.754631	10.00921	3.662446	12.00921	4.777331	14.00921
4.69458	10.05921	3.663299	12.05921	4.840546	14.05921
4.636751	10.10921	3.6654	12.10921	4.906172	14.10921
4.581073	10.15921	3.66875	12.15921	4.974296	14.15921
4.527478	10.20921	3.673351	12.20921	5.045016	14.20921
4.475904	10.25921	3.679207	12.25921	5.118432	14.25921
4.426293	10.30921	3.68632	12.30921	5.194654	14.30921
4.37859	10.35921	3.694696	12.35921	5.2738	14.35921
4.332744	10.40921	3.70434	12.40921	5.355994	14.40921

4.288707	10.45921	3.715259	12.45921	5.441373	14.45921
4.246434	10.50921	3.727459	12.50921	5.53008	14.50921
4.205883	10.55921	3.74095	12.55921	5.622271	14.55921
4.167014	10.60921	3.75574	12.60921	5.718114	14.60921
4.129792	10.65921	3.771839	12.65921	5.817789	14.65921
4.09418	10.70921	3.78926	12.70921	5.921491	14.70921
4.060147	10.75921	3.808013	12.75921	6.029429	14.75921

附录 2: 代码

代码均为 matlab 代码。以黑体字标示文件开头。

问题 1 代码

```
clear;clc;
%时间
set(0,'defaultfigurecolor','w');
date = datenum([2015 10 22]) - datenum([2015 1 1]) + 1;
%纬度角度值
fai = 39 + 54/60 + 26/100/60;
%3 分钟/60 分钟（将 60 制转换为 100 制）
dert = 3/60;
%时间序列，当地时间
E = 116 + 23/60 + 29/60/100;
t = 9 + dert:dert:15;
t = t + 4.*(E - 120)/60;
%太阳高度角弧度序列
arg = gaodujiao(fai,date,t);
%竿高度
H = 3;
%影子长度序列
yingzi = H./tan(arg)
plot(t,yingzi,'linewidth',3);
axis([t(1) t(end) 0 7])
set(gca,'xtick',9:1:15);
xlabel('时间/h');
ylabel('影长/m');
title('太阳影子长度的变化曲线');
[yingzi,t]
```

```
function [ arg ] = gaodujiao( fai,date,t )
%GAODUJIAO 参数：纬度，日期，时间
% 求出太阳高度角，弧度角
derta = chiwei(date);
omiga = 15.*(t- 12)*pi/180;
fai = fai*pi/180;
```

```

sinarg = sin(derta).*sin(fai) + cos(derta).*cos(fai).*cos(omiga);
arg = asin(sinarg);
end

```

```

function [ arg ] = chiwei( N )
%CHIWEI 根据 N（天）求得赤纬
%      每年的一月一号为 1,给出弧度角
    b = 2*pi*(N - 1)/365;
    d1 = 0.399912*cos(b);
    d2 = 0.070257*sin(b);
    d3 = 0.006758*cos(2*b);
    d4 = 0.000907*sin(2*b);
    d5 = 0.002697*cos(3*b);
    d6 = 0.00148*sin(3*b);
    arg = (0.006918 - d1 + d2 - d3 + d4 - d5 + d6);
end

```

问题 2 代码

%采用分区策略求解，结果在命令行中给出

%杆长，经度，纬度，y 轴方向角

```
global xlim;
```

```
xlim = [ 0 10
```

```
        -180 180
```

```
        -90 90
```

```
        0 360
```

```
];
```

```
Eindex = [-180 -90 0 90 180];
```

```
faiindex = [-90 -45 0 45 90];
```

```
for iii=1:4
```

```
    for jjj=1:4
```

```
        xlim = [
```

```
            0 10
```

```
            Eindex(iii) Eindex(iii + 1)
```

```
            faiindex(jjj) faiindex(jjj + 1)
```

```
            0 360
```

```
        ]
```

```
    pso
```

```
    end
```

```
end
```

function pso()

%PSO 算法求最小值

```
global num;
```

```
[num,str,cells] = xlsread('附件 1-3.xls',1,'A4:C24');
```

```

num(:,1)=linspace( 14 + 42/60,15+42/60,length(num(:,1)));
tic;                                %程序运行计时
E0=0.001;                          %允许误差
MaxNum=500;                        %粒子最大迭代次数
narvs=4;                          %目标函数的自变量个数
particlesize=100;                 %粒子群规模
c1=2.8;                          %每个粒子的个体学习因子，也称为加速
常数
c2=1.3;                          %每个粒子的社会学习因子，也称为加速
常数
w=0.6;                          %惯性因子
                                %粒子的最大飞翔速度（由各纬度的定义域
确定）
                                %粒子所在的位置

x = zeros(particlesize,narvs);
%x: 竿高，经度，纬度,夹角
global xlim;
for i=1:length(x)
    for j=1:narvs
        x(i,j) = (xlim(j,2) - xlim(j,1))*rand + xlim(j,1);
    end
end

                                %粒子的飞翔速度

for i=1:length(x)
    for j=1:narvs
        v(i,j) = (xlim(j,2) - xlim(j,1))*rand*0.2;
    end
end
vmax = (xlim(:,2) - xlim(:,1))*0.2;

%确定初始值
for i=1:particlesize
    f(i)=fitness(x(i,:));
end
personalbest_x=x;%各个粒子的最优时的 x
personalbest_faval=f;%各个粒子的最优函数值
[globalbest_faval,i]=min(personalbest_faval);%全局最优值， 对应索引
globalbest_x=personalbest_x(i,:);%全局最优值的 x
k=1;
while k<=MaxNum%迭代次数
    for i=1:particlesize%粒子群规模

```

```

        f(i)=fitness(x(i,:));
        if f(i)<personalbest_faval(i) %判断当前位置是否是历史上最佳位置
            personalbest_faval(i)=f(i);
            personalbest_x(i,:)=x(i,:);
        end
    end%改变各个粒子的历史最优值，点
    [globalbest_faval i]=min(personalbest_faval);
    globalbest_x=personalbest_x(i,:);
    for i=1:particlesize %更新粒子群里每个个体的最新位置
        v(i,:)=w*v(i,:)+c1*rand*(personalbest_x(i,:)-x(i,:))...
            +c2*rand*(globalbest_x-x(i,:));
        for j=1:narvs %判断粒子的飞翔速度是否超过了最大飞翔速度
            if v(i,j)>vmax(j);
                v(i,j)=vmax(j);
            elseif v(i,j)<-vmax(j);
                v(i,j)=-vmax(j);
            end
        end
        x(i,:)=x(i,:)+v(i,:);
    end
    if abs(globalbest_faval)<E0,break,end
    k=k+1;
end
Value1=globalbest_faval; Value1=num2str(Value1);
% strcat 指令可以实现字符的组合输出
disp(strcat('the minimum value','=',Value1));
%输出最大值所在的横坐标位置
Value2=globalbest_x; Value2=num2str(Value2);
disp(strcat('the corresponding coordinate','=',Value2));

```

%给出结果，绘图

```

%x: 竿高，经度，纬度,夹角
[num,str,cells] = xlsread('附件 1-3.xls',1,'A4:C24');
num(:,1)=linspace( 14 + 42/60,15+42/60,length(num(:,1)));
set(0,'defaultfigurecolor','w');
x =[1.982112      109.5      18.42842      164.9186];
h = x(1);
E = x(2);
fai = x(3);
sita = x(4)*pi/180;
if(E < 0)
    E = E + 360;
end

```

```

t = num(:,1) + 4.*(E - 120)/60;
t = 13:0.05:16;
date = [2015 4 18 0 0];
arfa = zeros(size(t));
beta = zeros(size(t));
for i=1:length(arfa)%提取两个角
    if(t(i)<0)%时间处理
        t(i) = t(i) + 24;
        date(3) = 17;
    elseif(t(i) > 24)
        t(i) = t(i) - 24;
        date(3) = 19;
    end%数据组成
    hour = floor(t(i));
    mint = floor(mod(t(i),1)*60);
    date(4) = hour;
    date(5) = mint;%计算
    [arfat,betat] = sun_osition_ch(E,fai,date);%e,fai,date
    arfa(i) = arfat*pi/180;
    beta(i) = betat*pi/180;
end

```

```

xline = (0:0.1:3)*sin(x(end)*pi/180);
yline = (0:0.1:3)*cos(x(end)*pi/180);
lall = h./tan(arfa);
xall = lall.*sin(sita + beta);
yall = lall.*cos(sita + beta);
hold on;
plot(0,0,'*');
plot(num(:,2),num(:,3),'.','linewidth',5);
plot(xall,yall,'black','linewidth',1);
plot(xline,yline);
gh = 0:0.1:2;
x0 = zeros(size(gh));
y0 = zeros(size(gh));
plot3(x0,y0,gh,'black','linewidth',3)
axis([-3 3 -3 3 0 2])
hold off

```

function [y] = fitness(x)

%FITNESS 方程与实际数据的距离，作为粒子群算法的目标函数

%输入参数为题目待求参数

%第一问，参数为纬度 fai,经度 E,竿长 h

```

        y = moxing2fangchengzu3( x );
end

```

function [date] = datetrans(date,E)

%DATETRANS 将当地时间转换为格林尼治时间

% 此处显示详细说明

```

    plus = -E*4/60;
    %年 月 日 时 分
    hour = date(4) + date(5) / 60;
    hour = hour + plus;
    if(hour >= 24)
        hour = hour -24;
        date(4) =floor( hour);
        date(5) = mod(hour,1)*60;
        date(3) = date(3) + 1;
        if(date(3) > 31)
            date(3) = 1;
            date(2) = date(2) + 1;
            if(date(2) > 12)
                date(2) = 1;
                date(1) = date(1) + 1;
            end
        end
    end
else
    date(4) =floor( hour);
    date(5) = mod(hour,1)*60;
end

```

end

function [y] = moxing2fangchengzu3(x)

%MOXING2 方程组 3 使用正规太阳位置确定函数进行求解

%

%x: 竿高, 经度, 纬度,坐标轴偏差,全部接受角度

```

    h = x(1);
    E = x(2);
    fai = x(3);
    sita = x(4)*pi/180;
    date = [2015 4 18 0 0];
    if(E>180 || E<-180 || fai>90 || fai<-90 || h<0)
        y = inf;
        return;
    end

```

```

global num;
if(E < 0)
    E = E + 360;
end
t = num(:,1) + 4.*(E - 120)/60;%由北京时间转换为当地时间
%改动只是高度角和方位角 t:小时为小数格式，得转换成分钟，秒等
%t 也可能是大于 24 和小于 0，都得考虑
arfa = zeros(size(t));
beta = zeros(size(t));
for i=1:length(arfa)
    if(t(i)<0)%时间处理
        t(i) = t(i) + 24;
        date(3) = 17;
    elseif(t(i) > 24)
        t(i) = t(i) - 24;
        date(3) = 19;
    end%数据组成
    hour = floor(t(i));
    mint = floor(mod(t(i),1)*60);
    date(4) = hour;
    date(5) = mint;%计算
    [arfat,betat] = sun_osition_ch(E,fai,date);%e,fai,date
    arfa(i) = arfat*pi/180;
    beta(i) = betat*pi/180;
end
lall = h./tan(arfa);
xall = lall.*sin(sita + beta);%计算值
yall = lall.*cos(sita + beta);
if(min(arfa) < 0)
    y = inf;
    return;
end
xd = xall - num(:,2);%实际值
yd = yall - num(:,3);
y = sum(sqrt(xd.^2 + yd.^2));
end

```

function [h,dir] = sun_osition_ch(E,fai,date)

%DIAOYONG 使得调用方便，只用给出当地时间即可调用 sun_position

date = datetrans(date,E);%转换成格林尼治时间

location.longitude =E; %经度

location.latitude = fai;


```

location.altitude = 0;
time.year = date(1);
time.month = date(2);
time.day = date(3);
time.hour = date(4);
time.min = date(5);
time.sec = 0;
time.UTC = 0;
sun = sun_position(time, location);
h = 90 - sun.zenith;%高度角
dir = sun.azimuth;%方位角
end

```

问题 3

%采用分区策略求解，结果在命令行中给出
%杆长，经度，纬度，y 轴方向角

```

global xlim;
xlim = [ 0 10
        -180 180
        -90 90
        0 360
        0 365
];
Eindex = [-180 -90 0 90 180];
faiindex = [-90 -45 0 45 90];
for iii=1:4
    for jjj=1:4
        xlim = [
            0 10
            Eindex(iii) Eindex(iii + 1)
            faiindex(jjj) faiindex(jjj + 1)
            0 360
            0 365
        ]
        pso_q3
    end
end

```

function pso_q3()

%PSO 算法求最小值

global num;

[num,str,cells] = xlsread('附件 1-3.xls',2,'A4:C24');**%第三问改动 xlsread 第 2 个输入参数**

%即可对两个附件

进行求解

```
num(:,1)=linspace(12+41/60,13+41/60,length(num(:,1)));
```

```
tic; %程序运行计时
```

```
E0=0.001; %允许误差
```

```
MaxNum=500; %粒子最大迭代次数
```

```
narvs=5; %目标函数的自变量个数
```

```
particlesize=100; %粒子群规模
```

```
c1=2.8; %每个粒子的个体学习因子，也称为加速
```

```
常数
```

```
c2=1.3; %每个粒子的社会学习因子，也称为加速
```

```
常数
```

```
w=0.6; %惯性因子
```

```
%粒子的最大飞翔速度（由各纬度的定义域
```

```
确定）
```

```
%粒子所在的位置
```

```
x=zeros(particlesize,narvs);
```

```
%x: 竿高，经度，纬度,夹角
```

```
global xlim;
```

```
for i=1:length(x)
```

```
    for j=1:narvs
```

```
        x(i,j)=(xlim(j,2)-xlim(j,1))*rand+xlim(j,1);
```

```
    end
```

```
end
```

```
%粒子的飞翔速度
```

```
for i=1:length(x)
```

```
    for j=1:narvs
```

```
        v(i,j)=(xlim(j,2)-xlim(j,1))*rand*0.2;
```

```
    end
```

```
end
```

```
vmax=(xlim(:,2)-xlim(:,1))*0.2;
```

```
%确定初始值
```

```
for i=1:particlesize
```

```
    x(i,end)=floor(x(i,end));
```

```
    f(i)=fitness_q3(x(i,:));
```

```
end
```

```
personalbest_x=x;%各个粒子的最优时的 x
```

```
personalbest_faval=f;%各个粒子的最优函数值
```

```
[globalbest_faval,i]=min(personalbest_faval);%全局最优值，对应索引
```

```
globalbest_x=personalbest_x(i,:);%全局最优值的 x
```

```

k=1;
while k<=MaxNum%迭代次数
    for i=1:particleSize%粒子群规模
        x(i,end) = floor(x(i,end));
        f(i)=fitness_q3(x(i,:));
        if f(i)<personalbest_faval(i) %判断当前位置是否是历史上最佳位置
            personalbest_faval(i)=f(i);
            personalbest_x(i,:)=x(i,:);
        end
    end%改变各个粒子的历史最优值，点
    [globalbest_faval i]=min(personalbest_faval);
    globalbest_x=personalbest_x(i,:);
    for i=1:particleSize %更新粒子群里每个个体的最新位置
        v(i,:)=w*v(i,:)+c1*rand*(personalbest_x(i,:)-x(i,:))...
            +c2*rand*(globalbest_x-x(i,:));
        for j=1:narvs %判断粒子的飞翔速度是否超过了最大飞翔速度
            if v(i,j)>vmax(j);
                v(i,j)=vmax(j);
            elseif v(i,j)<-vmax(j);
                v(i,j)=-vmax(j);
            end
        end
        x(i,:)=x(i,:)+v(i,:);
    end
    if abs(globalbest_faval)<E0,break,end
    k=k+1;
end
Value1=globalbest_faval; Value1=num2str(Value1);
% strcat 指令可以实现字符的组合输出
disp(strcat('the minimum value','=',Value1));
%输出最大值所在的横坐标位置
Value2=globalbest_x; Value2=num2str(Value2);
disp(strcat('the corresponding coordinate','=',Value2));

```

问题 4

%采用分区策略求解，结果在命令行中给出

%杆长，经度，纬度，y 轴方向角

global xlim;

```

xlim = [
        -180 180
        -90 90
        0 360

```

```

];

```

```

        Eindex = [-180 -90 0 90 180];
        faiindex = [-90 -45 0 45 90];
    for iii=1:4
        for jjj=1:4
            xlim = [
                Eindex(iii) Eindex(iii + 1)
                faiindex(jjj) faiindex(jjj + 1)
                0 360
            ];
            pso_q4
        end
    end
end

```

dataconfirm

%PSO 算法求最小值

```
tic;
```

```
E0=0.001;
```

```
MaxNum=500;
```

```
narvs=3;
```

夹角

```
particlesize=100;
```

```
c1=2.8;
```

常数

```
c2=1.3;
```

常数

```
w=0.6;
```

确定)

```
x = zeros(particlesize,narvs);
```

%x: 竿高, 经度, 纬度

```
global xlim;
```

```
for i=1:length(x)
```

```
    for j=1:narvs
```

```
        x(i,j) = (xlim(j,2) - xlim(j,1))*rand + xlim(j,1);
```

```
    end
```

```
end
```

%程序运行计时

%允许误差

%粒子最大迭代次数

%目标函数的自变量个数,3 个,经度,纬度,

%粒子群规模

%每个粒子的个体学习因子, 也称为加速

%每个粒子的社会学习因子, 也称为加速

%惯性因子

%粒子的最大飞翔速度 (由各纬度的定义域

%粒子所在的位置

%粒子的飞翔速度

```
for i=1:length(x)
```

```
    for j=1:narvs
```

```
        v(i,j) = (xlim(j,2) - xlim(j,1))*rand*0.2;
```

```

        end
    end
    vmax = (xlim(:,2) - xlim(:,1))*0.2;

    %确定初始值
    for i=1:particlesize
        f(i)=fitness_q4(x(i,:));
    end
    personalbest_x=x;%各个粒子的最优时的 x
    personalbest_faval=f;%各个粒子的最优函数值
    [globalbest_faval,i]=min(personalbest_faval);%全局最优值， 对应索引
    globalbest_x=personalbest_x(i,:);%全局最优值的 x
    k=1;
    while k<=MaxNum%迭代次数
        for i=1:particlesize%粒子群规模
            f(i)=fitness_q4(x(i,:));
            if f(i)<personalbest_faval(i) %判断当前位置是否是历史上最佳位置
                personalbest_faval(i)=f(i);
                personalbest_x(i,:)=x(i,:);
            end
        end%改变各个粒子的历史最优值， 点
        [globalbest_faval i]=min(personalbest_faval);
        globalbest_x=personalbest_x(i,:);
        for i=1:particlesize %更新粒子群里每个个体的最新位置
            v(i,:)=w*v(i,:)+c1*rand*(personalbest_x(i,:)-x(i,:))...
                +c2*rand*(globalbest_x-x(i,:));
            for j=1:narvs %判断粒子的飞翔速度是否超过了最大飞翔速度
                if v(i,j)>vmax(j);
                    v(i,j)=vmax(j);
                elseif v(i,j)<-vmax(j);
                    v(i,j)=-vmax(j);
                end
            end
        end
        x(i,:)=x(i,:)+v(i,:);
    end
    if abs(globalbest_faval)<E0,break,end
    k=k+1;
end
Value1=globalbest_faval; Value1=num2str(Value1);
% strcat 指令可以实现字符的组合输出
disp(strcat('the minimum value','=',Value1));
%输出最大值所在的横坐标位置

```

```
Value2=globalbest_x; Value2=num2str(Value2);
disp(strcat('the corresponding coordinate','=',Value2));
```

```
function [ y ] = fitness_q4( x)
```

```
%FITNESS 方程与实际数据的距离，作为粒子群算法的目标函数
```

```
%输入参数为题目待求参数
```

```
%第一问，参数为纬度 fai,经度 E,竿长 h
```

```
y = moxing2fangchengzu4( x );
```

```
end
```

```
function [ y ] = moxing2fangchengzu4( x )
```

```
%MOXING2 方程组 3 使用正规太阳位置确定函数进行求解
```

```
%
```

```
%x: 经度, 纬度,坐标轴偏差,全部接受角度
```

```
h = 2;
```

```
E = x(1);
```

```
fai = x(2);
```

```
sita = x(3)*pi/180;
```

```
date = [2015 7 13 0 0];
```

```
if(E>180 || E<-180 || fai>90 || fai<-90)
```

```
    y = inf;
```

```
    return;
```

```
end
```

```
global num;
```

```
if(E < 0)%日界线修正
```

```
    E = E + 360;
```

```
end
```

```
t = num(:,1) + 4.*(E - 120)/60;%由北京时间转换为当地时间
```

```
%改动只是高度角和方位角 t:小时为小数格式，得转换成分钟，秒等
```

```
%t 也可能是大于 24 和小于 0，都得考虑
```

```
arfa = zeros(size(t));
```

```
beta = zeros(size(t));
```

```
for i=1:length(arfa)
```

```
    if(t(i)<0)%时间处理
```

```
        t(i) = t(i) + 24;
```

```
        date(3) = 17;
```

```
    elseif(t(i) > 24)
```

```
        t(i) = t(i) - 24;
```

```
        date(3) = 19;
```

```
    end%数据组成
```

```
    hour = floor(t(i));
```

```
    mint = floor(mod(t(i),1)*60);
```

```
    date(4) = hour;
```

```

        date(5) = mint;%计算
        [arfat,betat] = sun_osition_ch(E,fai,date);%e,fai,date
        arfa(i) = arfat*pi/180;
        beta(i) = betat*pi/180;
    end
    lall = h./tan(arfa);
    xall = lall.*sin(sita + beta);%计算值
    yall = lall.*cos(sita + beta);
    if(min(arfa) < 0)
        y = inf;
        return;
    end
    xd = xall - num(:,2);%实际值
    yd = yall - num(:,3);
    y = sum(sqrt(xd.^2 + yd.^2));
end

```

附加问

%采用分区策略求解，结果在命令行中给出

%杆长，经度，纬度，y轴方向角

```
clear;clc;
```

```
global xlim;
```

```
dataconfirm
```

```
xlim = [
```

```
    -180 180
```

```
    -90 90
```

```
    0 360
```

```
    0 365
```

```
];
```

```
Eindex = [-180 -90 0 90 180];
```

```
faiindex = [-90 -45 0 45 90];
```

```
for iii=1:4
```

```
    for jjj=1:4
```

```
        xlim = [
```

```
            Eindex(iii) Eindex(iii + 1)
```

```
            faiindex(jjj) faiindex(jjj + 1)
```

```
            0 360
```

```
            0 365
```

```
        ];
```

```
        pso_q42
```

```
    end
```

```
end
```

%PSO 算法求最小值

```
tic; %程序运行计时
E0=0.001; %允许误差
MaxNum=500; %粒子最大迭代次数
narvs=4; %目标函数的自变量个数
particlesize=100; %粒子群规模
c1=2.8; %每个粒子的个体学习因子，也称为加速常数
c2=1.3; %每个粒子的社会学习因子，也称为加速常数
w=0.6; %惯性因子
%粒子的最大飞翔速度（由各纬度的定义域确定）
%粒子所在的位置

x = zeros(particlesize,narvs);
%x: 竿高，经度，纬度,夹角
global xlim;
xlim
for i=1:length(x)
    for j=1:narvs
        x(i,j) = (xlim(j,2) - xlim(j,1))*rand + xlim(j,1);
    end
end

%粒子的飞翔速度

for i=1:length(x)
    for j=1:narvs
        v(i,j) = (xlim(j,2) - xlim(j,1))*rand*0.2;
    end
end
vmax = (xlim(:,2) - xlim(:,1))*0.2;

%确定初始值
for i=1:particlesize
    x(i,end) = floor(x(i,end));
    f(i)=fitness_q42(x(i,:));
end
personalbest_x=x;%各个粒子的最优时的 x
personalbest_faval=f;%各个粒子的最优函数值
[globalbest_faval,i]=min(personalbest_faval);%全局最优值，对应索引
globalbest_x=personalbest_x(i,:);%全局最优值的 x
k=1;
```



```

while k<=MaxNum%迭代次数
    for i=1:particleSize%粒子群规模
        x(i,end) = floor(x(i,end));
        f(i)=fitness_q42(x(i,:));
        if f(i)<personalbest_faval(i) %判断当前位置是否是历史上最佳位置
            personalbest_faval(i)=f(i);
            personalbest_x(i,:)=x(i,:);
        end
    end%改变各个粒子的历史最优值，点
    [globalbest_faval i]=min(personalbest_faval);
    globalbest_x=personalbest_x(i,:);
    for i=1:particleSize %更新粒子群里每个个体的最新位置
        v(i,:)=w*v(i,:)+c1*rand*(personalbest_x(i,:)-x(i,:))...
            +c2*rand*(globalbest_x-x(i,:));
        for j=1:narvs %判断粒子的飞翔速度是否超过了最大飞翔速度
            if v(i,j)>vmax(j);
                v(i,j)=vmax(j);
            elseif v(i,j)<-vmax(j);
                v(i,j)=-vmax(j);
            end
        end
        x(i,:)=x(i,:)+v(i,:);
    end
    if abs(globalbest_faval)<E0,break,end
    k=k+1;
end
Value1=globalbest_faval; Value1=num2str(Value1);
% strcat 指令可以实现字符的组合输出
disp(strcat('the minimum value','=',Value1));
%输出最大值所在的横坐标位置
Value2=globalbest_x; Value2=num2str(Value2);
disp(strcat('the corresponding coordinate','=',Value2));

```

function [y] = fitness_q42(x)

```

%FITNESS 方程与实际数据的距离，作为粒子群算法的目标函数
%输入参数为题目待求参数
%第一问，参数为纬度 fai,经度 E,竿长 h
y = moxing2fangchengzu3_q42( x );
end

```

function [y] = moxing2fangchengzu3_q42(x)

```

%MOXING2 方程组 3 使用正规太阳位置确定函数进行求解
%

```

%x: 竿高, 经度, 纬度, 坐标轴偏差, 全部接受角度, 对于第三问, 参数增加到 5 个

```
h = 2;
E = x(1);
fai = x(2);
sita = x(3)*pi/180;
date = x(4);
number = datenum([2015 1 1]) + date;
if(E>180 || E<-180 || fai>90 || fai<-90 || date<0 || date>365)
    y = inf;
    return;
end
date = datevec(number);
%这里需要将参数转换为时间数组以供 sun_position.m 使用
global num;
if(E < 0)
    E = E + 360;
end
t = num(:,1) + 4.*(E - 120)/60;%由北京时间转换为当地时间
%改动只是高度角和方位角 t:小时为小数格式, 得转换成分钟, 秒等
%t 也可能是大于 24 和小于 0, 都得考虑
arfa = zeros(size(t));
beta = zeros(size(t));
for i=1:length(arfa)
    if(t(i)<0)%时间处理
        t(i) = t(i) + 24;
        date(3) = 17;
    elseif(t(i) > 24)
        t(i) = t(i) - 24;
        date(3) = 19;
    end%数据组成
    hour = floor(t(i));
    mint = floor(mod(t(i),1)*60);
    date(4) = hour;
    date(5) = mint;%计算
    [arfat,betat] = sun_posation_ch(E,fai,date);%e,fai,date
    arfa(i) = arfat*pi/180;
    beta(i) = betat*pi/180;
end
lall = h./tan(arfa);
xall = lall.*sin(sita + beta);%计算值
yall = lall.*cos(sita + beta);
if(min(arfa) < 0)
    y = inf;
```

```
        return;  
    end  
    xd = xall - num(:,2);%实际值  
    yd = yall - num(:,3);  
    y = sum(sqrt(xd.^2 + yd.^2));  
end
```