



## 2017 湖南省研究生数学建模竞赛参赛承诺书

我们仔细阅读了湖南省研究生数学建模竞赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们授权湖南省研究生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号是（从组委会提供的试题中选择一项填写）：B

我们的队号为（填写完整的队号）：201718001001

所属学校（请填写完整的全名）：国防科学技术大学

参赛队员（打印并签名）：

1. 朱元昊

2. 陈青

3. 朱云飞

指导教师或指导教师组负责人(打印并签名)：王丹

日期：2017年4月25日

---

评阅编号（由组委会评阅前进行编号）：



# 湖南省第三届研究生数学建模竞赛

题目 **智慧考古探测的最优探测方式研究**

---

**摘 要：**

本文主要解决在考古探测的过程中确定最优探测方式的实际问题，待探测区域是两个相同的相距一定距离的细长矩形区域，区域内的待探测点均匀排列。

为了尽可能缩短探测所需要的总时间，在考虑探测车相互干扰作用，根据探测时的约束条件建立了计算机仿真模型，模型中将各时间节点下，探测车及待探测点的状态以及状态的动态变化进行定量描述，在各探测点均达到被探测的状态时计算得出累积的时间，通过随机生成进行大量的可行解并进行仿真试验并选定最优值。

为了进一步高效计算出完成任务所花的最少时间，我们通过试验和理论分析的方法确定了较为普适的一种探测方式，通过建立工作时序图的方式详细给出了探测车的初始布局，行走路线、扫描次序，并编写了计算程序，通过这种方式确定了最高效的探测车的台数。此外，为了方便计算机进行无线远程协调控制，还将的结果表达成计算机能够处理的数据组织方式。

关键词：最优探测 计算机仿真 协调控制 数据组织方式

# 一、问题重述

## 1.1 问题的背景

目前，在考古探测领域，人们已经摒弃了原有的探洞、探沟、探方等实验性挖掘方式，而采取非接触式的方式通过运用“探测车”，对地下建筑进行探测。探测车通过发射、接收并分析一种特殊的“波”来进行地下建筑情况和地质结构的探测。

## 1.2 问题的提出

探测的方式是将地面上的工作区域划分为若干矩形区域，探测车一个节点扫描时间为 12 秒。对于大面积的探测工作，需要多台探测车协同工作，为了避免互相干扰，扫描时间间隔根据它们之间的距离受到一定的限制（见图 1-1）。根据该区域节点选取的方式（如图 1-2）来看，其横向 100 米，纵向 50 米，所行走的一个节点时间分别为 24 秒和 12 秒。

## 1.3 待解决问题

(1) 分别计算 12 台、20 台、64 台、80 台探测车的初始布局，行走路线、扫描次序、完成任务所花的最少时间。

(2) 需要投入多少台的探测车可以使得完成任务的效率最高？给出其初始布局，行走路线，扫描次序，完成任务所花的最少时间。

(3) 为了方便计算机进行无线远程协调控制，请进一步将 (1)、(2) 的结果表达成计算机能够处理的数据组织方式，即：任意时刻探测车在什么位置？何时计算机向那个探测车发布什么指令？

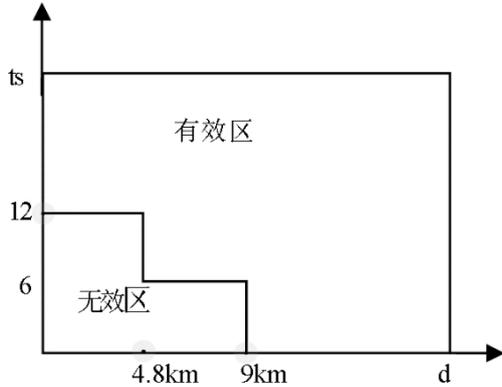


图 1-1 扫描限制图

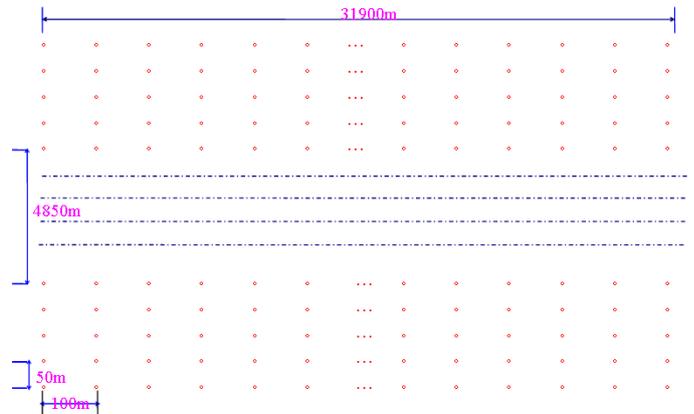


图 1-2 探测点分布

# 二、符号表示

$n$ : 均分后河上或河下的车台数

$T$ : 一个“L”型周期运动时间

$t_2$ : 最后一组间隔时间

$t_{纵}$ : 一个周期性纵向运动时间

$t_{扫}$ : 一个节点的扫描时间

$t_{纵1}$ : 50m 运动时间

$t_{横}$ ：一个横向过渡时间  
 $t_{横1}$ ：一个横向行走时间

### 三、问题假设

- 1.河岸两岸上下对称，且探测车的数量为偶数，探测车是无法过河的。
- 2.探测车在运行过程中没有受到其他自然因素和人为因素的干预。
- 3 探测车不必全部进行工作，即可以出现“废车”。

### 四、模型一的建立和求解

#### 4.1 问题分析

该问题是一道最优化问题，较为普遍的优化算法主要是遗传算法，模拟退火算法，粒子群算法，蒙特卡罗方法等，但是该问题的可行域巨大、约束条件复杂，通过智能寻优的方法会导致编码和寻优过程的困难，为此我们建立了一种计算仿真的模型，通过随机生成大量可行解，再比较各可行解的优劣，获得一种近似的最优解的方法。

#### 4.2 计算机仿真模型的建立

对该过程每个时间节点的状态以及状态的变化进行定量描述，首先根据特定的探测车的数量，随机生成一些初始位置，其次同样通过随机生成的方式，在一定约束条件下对其移动路径及探测次序进行仿真，最终在所有的探测点都已被探测过的状态下通过所经历的时间步长的个数，来计算出探测时间，然后进行大量的重复试验，比较探测时间，选出最优方案。其算法流程及过程细节如图 4-2。

初始时刻随机生成  $2n$  个探测车的位置，根据题目的要求  $2n=12$ ， $2n=20$ ， $2n=64$ ， $2n=80$ 。建立大小为  $10 \times 320$  的节点状态矩阵  $A$ ，初始时刻  $A$  矩阵各个元素的值均为零，在探测过程中，被探测过的节点状态更新为 1；建立大小  $2n \times 1$  的探测车状态矩阵  $B$ ，矩阵中元素初始状态为 0，正在扫描的探测车状态表示为，扫描完毕的探测车状态表示为 0；建立探测车位置矩阵  $C$ ，随着时间状态的更新而不断更新矩阵  $C$ 。整个过程中，时间以 6s 作为周期进行状态更新。节点状态更新示意图如下：

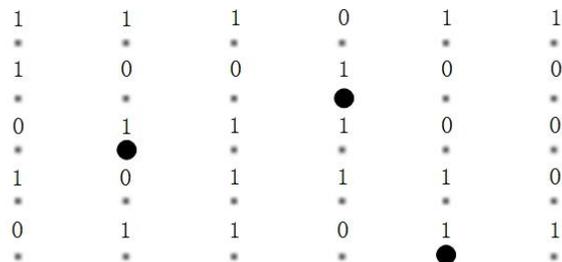


图 4-1 节点状态更新示意图

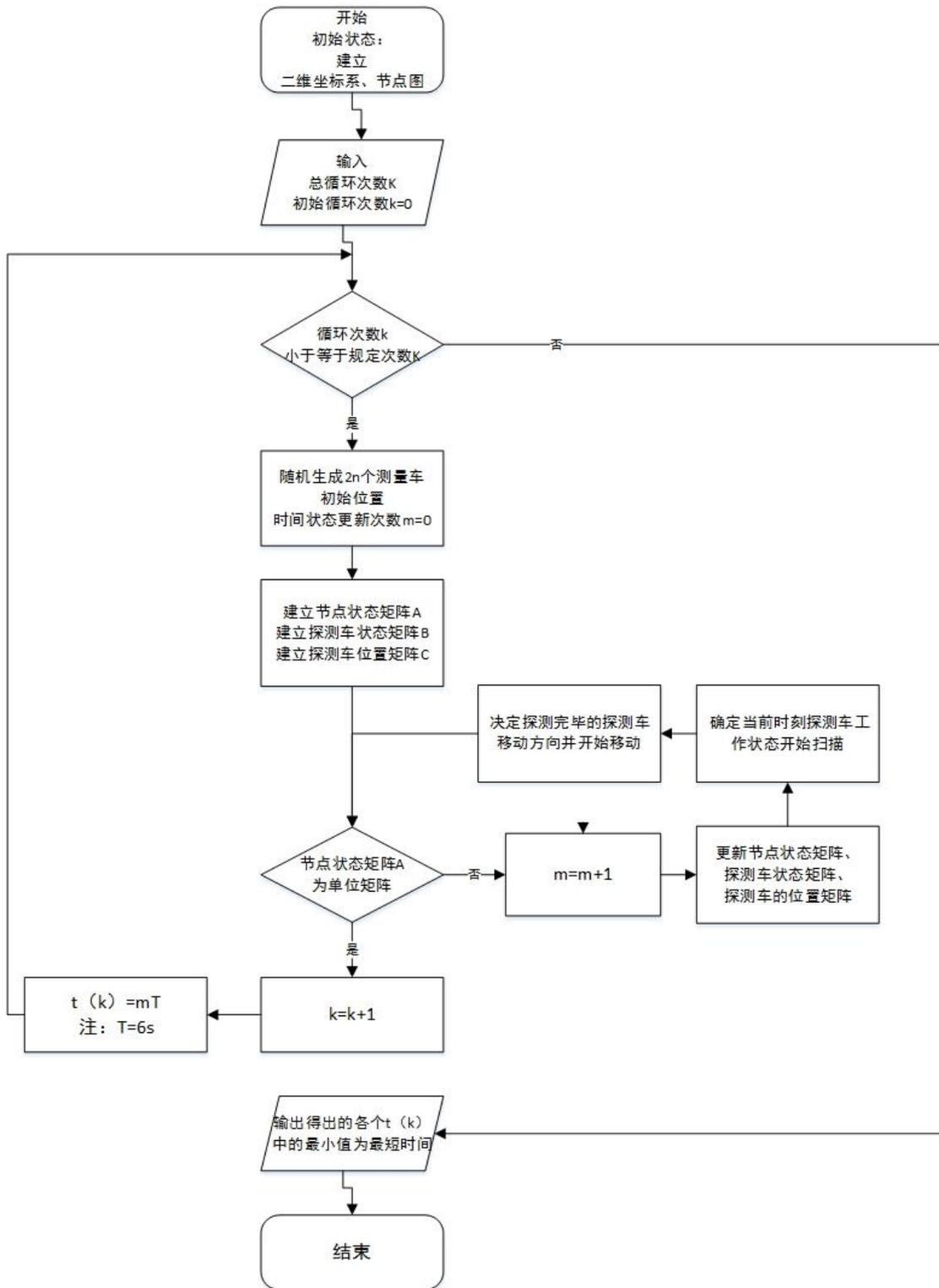


图 4-2 算法流程

默认处在空闲状态的最左上角的探测车当前时刻可以进行扫描，通过各个探测车之间的间距得出本时刻可同时进行扫描的全部探测车，并开始扫描。完成探测车的移动方向通过势场法进行判定探测车的移动方向。根据不同方向上未完成探测的节点数目作为引力，同时考虑到节点纵向距离比横向距离短一倍，因而探测车纵向运动的概率更大。通过以上条件约束形成引力，以概率  $P_{ij}$  ( $i$  表示探测车编号,  $j$  代表下一时刻运动方向) 生成下一时刻探测车的移动方向。如图所示，

依概率  $P_{ij}$  随机生成从 1 到 4 的正整数，决定不同的运动方向。

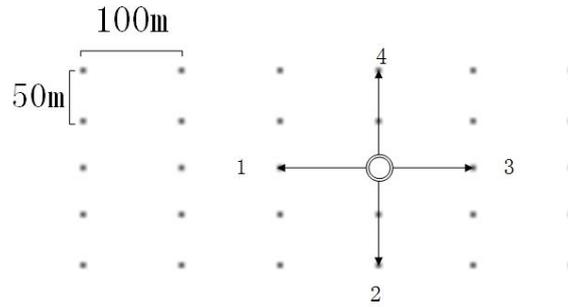


图 4-3 探测车移动方向示意图

探测车的运动存在一定的约束条件，条件约束如下：

- 1.任何探测车不能超过边界，如果下一时刻选择的运动方向超出边界，这个选择就是无效的，需要重新开始选择。
- 2.任何两个点下一时刻的前进方向不能是同一点，如果出现这种选择则令编号大的车改变前进方向。
- 3.任何探测车下一时刻的目标节点，在下一时刻不能存在另外一台车，若存在则重新生成运动方向。
- 4.考虑到探测车从一个节点运动到另一个节点所需要的时间等于或是二倍于扫描时长，为简化分析模型，所有探测车移动到未探测过的节点后一定要完成探测才能前往下一节点。

整个流程每循环一次会求解出所需时间，探测车位置状态矩阵的更新可以说明各台车所处位置和行走路线，通过不断循环求解出最优状态。

## 五、模型二的建立与求解

### 5.1 问题分析

首先考虑 1 台探测车在不受影响的前提下扫描一侧区域的情况，为使其走过所有的区域，可以采取一种纵向推进和横向推进的方式，如图所示。由于节点之间的纵向间距为 50m，横向间距为 100m，应该尽量减少横向的移动才能使在移动中消耗的时间尽量少。这就要求探测车尽可能多在纵向上运动而少在横向上运动。考虑一种极限的情况，即一种“弓”字形的移动轨迹。



图 5-1 纵向“弓”字形的移动轨迹

我们沿着上述移动方向依次对探测点进行编号。假设在河的一岸区域内，共有  $n$  台探测车，则一共布置了  $2n$  台探测车，每台探测车需要探测的节点数为

$\frac{320 \times 5}{n}$ ，但所得结果不为整数时则向上取整，在最后一个工作周期中编号靠后的

探测车停止工作。为避免探测位置重复探测我们设计初始位置的编号以此为上一台探测车需要探测点的下一号。并且所有探测车沿着探测点编号增加的顺序移动

并进行探测。这样的探测结果是最为高效，所需时间也是最少的。

5	6	15	16	25	26	35	36		
4	7	14	17	24	27	34	37		
3	8	13	18	23	28	33	38		
2	9	12	19	22	29	32	39		
1	10	11	20	21	30	31	40		

图 5-2 点阵标号

以  $n = 6$  为例，给定探测车的数量  $2n=12$ ，将整个的探测区域尽可能均匀地划分成 12 个区域，河岸两侧各有六个均分区域，在每个区域中小车的运动轨迹模式基本上都是一致的，所以我们只需研究其中的部分探测车的运动轨迹即可。其初始分布位置分别为 1, 268, 535, 802, 1069, 1335。之后以次进行探测，便可以计算出探测所需的总时间。

为了计算某一特定探测车台数探测全部探测点所需的最短时间，我们定义一个物理量  $N$  表示平均每一个时间间隔  $\Delta t$  中处于探测状态的探测车的数量，由于该问题所涉及的特征时间都是 6s 的倍数，我们取时间间隔  $\Delta t = 6s$ ，即每隔一个时间间隔  $\Delta t$ ，更新一次状态，初始时间节点为  $t_0$ ，之后一次为  $t_1, t_2, t_3, \dots$ 。易知：当所有结点全部探测完毕时所需的时间  $t$  满足  $N \times t = 3200$ ，为使  $t$  尽可能短，这需要使  $N$  尽可能大。为了描述探测的效率，我们定义另一个物理量  $Q$  表示平均一个时间步长中处于探测状态下的探测车占总探测车数量的比值。为了寻求最短的探测时间和最高效的探测车台数，我们着重研究  $N, Q$  的变化规律。为了便于计算出探测的最短时间  $t_{\min}$ ，我们希望所有的探测车能按照几种特定的工作时序，并且这样工作时序具有周期性，而且所有的探测车能够相互配合，减少相互影响，这样我们便能较为快捷的估计出探测的最短时间  $t_{\min}$ 。

为了考虑探测车相互影响的作用范围，我们采用了等比例绘图的方式确定一台探测车在进行探测时所能影响到的范围。并建立如图所示的坐标系分别假设探测车的位置位于  $(0, 0), (0, 50), (0, 100), (0, 150), (0, 200)$ ，并依次以这些位置为圆心以 4800m 和 9000m 为半径画圆。然后确定距离圆边界最近的点的坐标。从图中可以发现一台探测车在进行探测的时候，只能对其一侧的区域产生强干扰，对另一侧的一定范围内产生弱干扰。

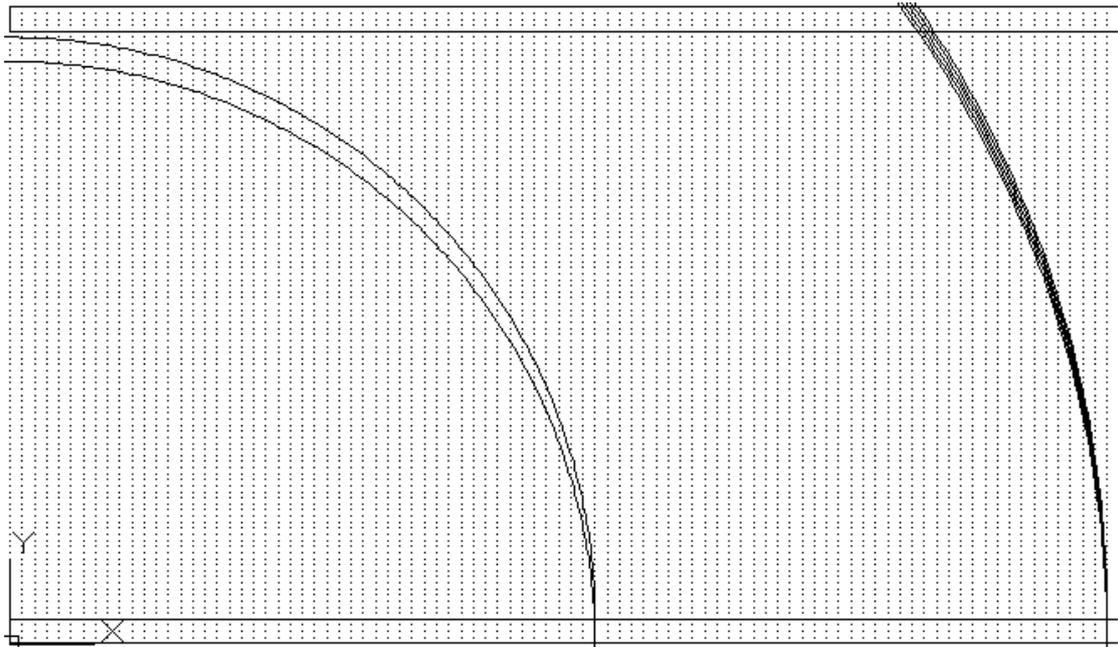


图 5-3 强干扰和弱干扰示意图

我们按照初始均匀分布和工作时序周期性的思想,以探测车台数为  $2n=12$  为例,将探测区域均匀分成 12 个区域,每个区域布置一台探测车,这样可以使探测车的距离最远,区域内探测车的初始位置都位于区域的左下角,让同一组区域内的探测车按照确定的时序进行探测,所以其相邻探测车的位置能够保持不变,我们能够通过简单计算得出同侧相邻两点初始位置的距离  $l = 31900 / n$ 。

1	2	3	4	5	6
7	8	9	10	11	12

图 5-4 12 台探测车的区域划分

当距离  $l > 9000$  时,1 区探测车在进行探测时,其右侧区域不会受其干扰,而下侧区域则会受到其干扰,必须延后 1 个  $\Delta t$  在进行探测,我们按照之前讨论的“弓”形移动方案。但是探测车在进行探测的过程中,还需要进行位置的移动,而位置移动的时间会导致扫描效率  $Q$  的下降,而探测车的数目比较少,则  $N$  比较小,当  $9000 > l > 4800$  时,虽然探测车会相互进行干扰,考虑到位置竖向移动的时间与扫描的相同,我们希望尽量将探测穿插进行工作。而一次竖向移动和扫描的总时间为 24s,而此时相互影响时间为 6s,这样可以将所有的探测车分成 4 组,4 组以此进行探测和移动,每组工作时序相差 6s,这样可以获得最大的扫描效率  $Q$ 。并且  $Q$  的值相比于  $l > 9000$  减小量较少,而  $N$  值却有了很大的提高。依旧以  $2n=12$  台探测车为例,将探测区域均匀分成 12 个区域,每个区域布置一台探测车,这样可以使探测车的距离最远,然后将 12 个区域分成 a、b、c、d 四组,a 组为 1、3、5 区,b 组为 2、4、6 区,c 组为 7、9、11 区,d 组为 8、10、12 区。并且每个区域内探测车都位于区域的左下角,让同一组区域内的探测车按照下表所示的确定的时序进行探测。

表 5-1 (a)

时间	1	2	3	4	5	6	7	8	9
a	探测	探测	移动	移动	探测	探测	移动	移动	探测
b	待机	探测	探测	移动	移动	探测	探测	移动	移动
c	待机	待机	探测	探测	移动	移动	探测	探测	探测
d	待机	待机	待机	探测	探测	移动	移动	探测	探测

表 5-1 (b)

时间	10	11	12	13	14	15	16	17	18
a	探测	移动	移动	探测	探测	移动	移动	探测	探测
b	探测	探测	移动	移动	探测	探测	移动	移动	探测
c	移动	探测	探测	移动	移动	探测	探测	移动	移动
d	移动	移动	探测	探测	移动	移动	探测	探测	移动

表 5-1 (c)

时间	19	20	21	22	23	24	25	26	27
a	移动	移动	移动	移动	探测	探测	移动	移动	探测
b	探测	移动	移动	移动	移动	探测	探测	移动	移动
c	探测	探测	移动	移动	移动	移动	探测	探测	移动
d	移动	探测	探测	移动	移动	移动	移动	探测	探测

表 5-1 车辆状态表

在以上的工作时序中我们 1 个组每隔 18 个时间间隔  $\Delta t$  恰好探测完毕一个“I”型的路线，每隔 22 个时间间隔  $\Delta t$  恰好探测完毕一个“L”型的路线，正好完成了 1 列探测点的扫描，下一组则较前一组延后 1 个时间间隔  $\Delta t$ ，在本例中 4 组相比较于 1 个组则会延后 3 个时间间隔  $\Delta t$ 。

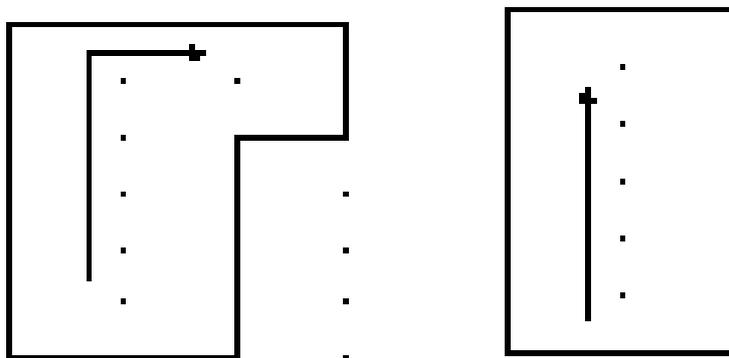


图 5-5 “L”与“I”周期图

问题分析已经得出，为扫过所有的点，我们采用纵向往复从左到右进行扫描运动。这是一种周期式的运动，每一个周期包含四段纵向扫描运动和一段横向扫描运动，呈现“L”型。图

每一段纵向运动的时间为

$$t_{纵总} = t_{扫} + t_{纵1} = 12s + 12s = 24s$$

每一段横向运动的时间为

$$t_{横} = t_{扫} + t_{横1} = 12s + 24s = 36s$$

每一个周期的运动时间为

$$T = 4t_{纵} + t_{横} = 4 \times 24 + 36 = 132s$$

当  $l < 4800$  时，我们首先考虑当  $n$  非常大时，并且  $n$  依然不断变大的过程， $N$  所受  $n$  变化的影响会越来越小而只会受到探测车之间的相互干扰，而这种相互影响作用的范围是确定的，所以可以想象，必然存在某一特定的  $n$  值  $n_{max}$ ，满足这样的条件，当  $n > n_{max}$  时，继续增加探测车数目，其  $N$  不会发生变化，而  $N$  与探测总时间  $t$  是一种反比的关系，所以我们想要确定这一  $n_{max}$  值，可以使时间最短，又不必使跟多的探测车处于完全无用的状态，从而控制  $Q$  值。此时可以认为是最高效的探测车台数。

实际上，4.1 的方法虽然在原理上能够解决这个问题，但是计算量庞大，对此我们进行了一些简化处理，建立新的模型进行运算。

## 5.2 12 台探测车的布局模型

### 5.2.1 初始布局

由问题分析得出小车之间的距离小于 9km 大于 4.8km 时扫描间隔大于 6 秒有效，小车之间的距离小于 4.8km 时扫描间隔大于 12 秒有效，为保证多台车能同时运行，将 12 台车编号并分为四组 A.B.C.D。

表 5-2 12 台车辆分组

组号	组员
A	1,3,5
B	2,4,6
C	7,9,11
D	8,10,12

并且每个区域内探测车都位于区域的左下角，河岸两侧各有六个均分区域。

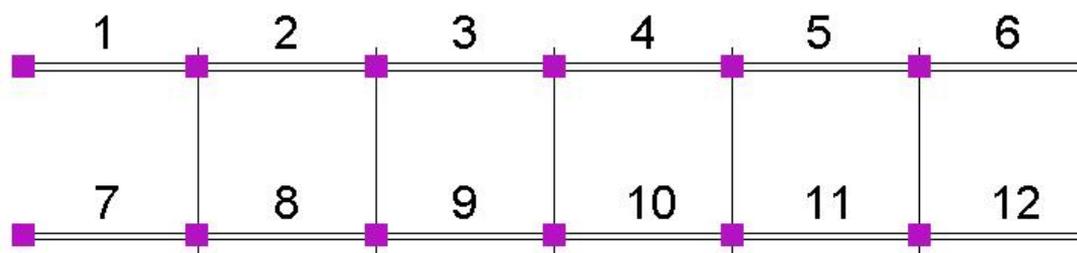


图 5-6 12 台探测车的初始位置分布

为使每组车辆能间隔 6 秒交替进行，1-4 区域宽  $d_1 = 5300m$ ，5、6 区域宽

为  $d_2=5400\text{m}$ 。区域高  $h=200\text{m}$ 。河上、河下探测车  $n_1, n_2$ ，距离

$$L = \sqrt{5050^2 + [(n_2 - 7 - n_1) \times 5300]^2}$$

### 5.2.2 扫描路线

在每个区域中，小车的运动轨迹模式应是一致的，这样可使小车间保持初始的时间间隔有序的运行下去，时间最优。所以我们只需研究其中的一辆探测车的运动轨迹即可。

由问题分析使用“弓”字形的移动轨迹（如图 5-1）。

### 5.2.3 扫描次序

A 组探测车先开始扫描，扫描时间为 0-12s，考虑到 50m 距离探测车要走 12s 时间，到达下一个节点。为使时间最优，相邻两组扫描间隔为 6s。为使 A 组探测车到达下一个节点没有停留时间立刻开始扫描，如此连续循环下去，故做出下图安排。

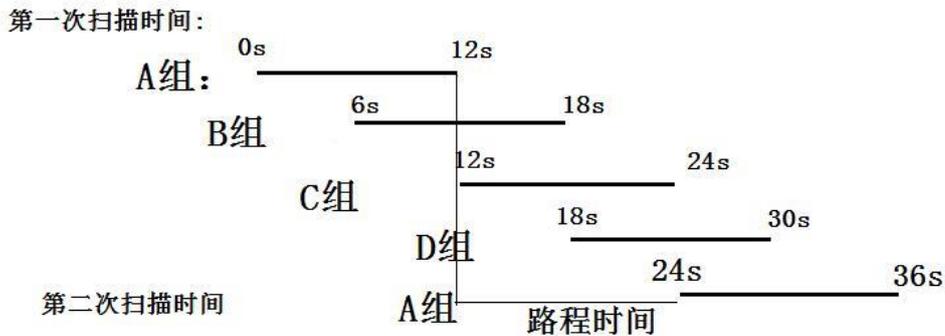


图 5-7 12 台车扫描时间图

以  $\Delta t = 6\text{s}$  为一个步长，每个时间节点下各组探测车的状态如表 5-1。

### 5.2.4 完成任务所花的最少时间

以 1 号探测车为例，在纵向运动中 0s-96s 内小车的运动服从周期性运动规律，每个周期为一个节点的扫描时间 12s 加上路程运动时间 12s。如图 4-3 所示：

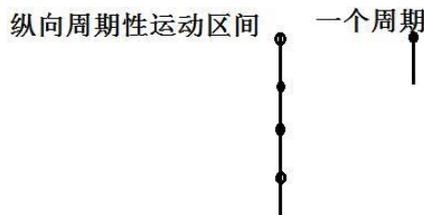


图 5-8 纵向周期图

定义一个周期性纵向运动的时间为

$$t_{\text{纵}} = t_{\text{扫}} + t_{\text{纵1}} = 12\text{s} + 12\text{s} = 24\text{s}$$

将“L”型运动分为 4 个纵向运动和一个横向运动，因 100m 运动时间为 24s

故每一段横向运动的时间为一个节点扫描时间加上一个横向运动时间：

$$t_{横} = t_{扫} + t_{横1} = 12s + 24s = 36s$$

故每一个“L”型周期的运动时间为

$$T = 4t_{纵} + t_{横} = 4 \times 24 + 36 = 132s$$

定义一号车坐标系  $x_1, y_1$

故  $x_1$  与  $y_1$  可利用周期性运动规律，根据时刻  $t$  与时间  $T, t_{纵}, t_{扫}, t_{纵1}, t_{横},$

$t_{横1}$  的关系得出。

我们在 12tai/ru1.m 文件中实现了此算法，其时序轨迹图如下：

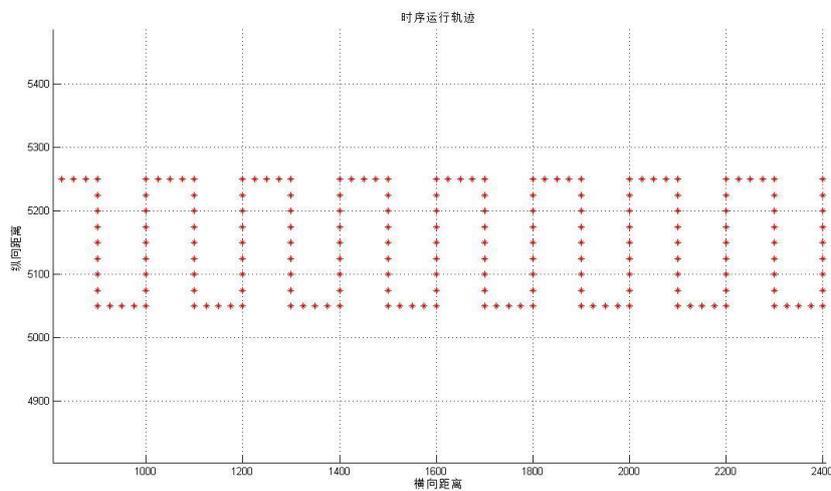


图 5-9 1 号探测车的时序轨迹图

图中显示的 1 号探测车的时序轨迹图，以 6 秒为周期不断更新探测车的位置信息，形成该辆探测车的时序运行轨迹。

在实际应用中可能出现两种情况，一是知道此时的时刻而想知道全局所有探测车的所在位置，二是知道某台车的实际位置，想知道此时的时刻和其他探测车辆的位置。为此，本模型中以时间和探测车位置作为两个并行的自变状态量，可以完成时间与位置的交互计算功能。

在文件 12tai/main.m 中，输入 1 号车探测车的最终所在位置坐标  $x_1 = 5200$ ， $y_1 = 5250$ ，可得 1 号车完成全部探测的时间  $t = 6978s$ 。

总时间

$$T_{总} = t + t_2 + T = 7128s$$

## 5.3 20 台探测车的布局模型

### 5.3.1 初始布局

仿照 5.2 中 12 台车的模型分析方法，因总节点数一定，为使同一时刻探测车辆尽量多，可给出两套模型方法：

**(1) 第一套布局:**

考虑 20 台探测车的情况为了尽量避免探测车的相互影响,我们只使用 18 台探测按照下图所示的划分区域其中 9 和 18 号区域的长度为 1900m,其余为 3750m,这样当所有探测车均位于其所分配区域的左下角时,1 号探测车与 12 号探测车距离  $L = \sqrt{5050^2 + 7600^2} > 9\text{km}$ ,恰好不能影响到 12 区的探测车,但是会对 2 区的探测车产生强影响,对 3、10、11 区的探测车产生弱影响,同理 12 区的探测车不会对 5 区的探测车产生影响,所以我们将 18 个区域分成 A、B、C、D 四组如下:

表 5-3 18 台车辆分组

组号	组员
A	1、12、5、16、9
B	10、3、14、7、18
C	2、13、6、17
D	11、4、15、8

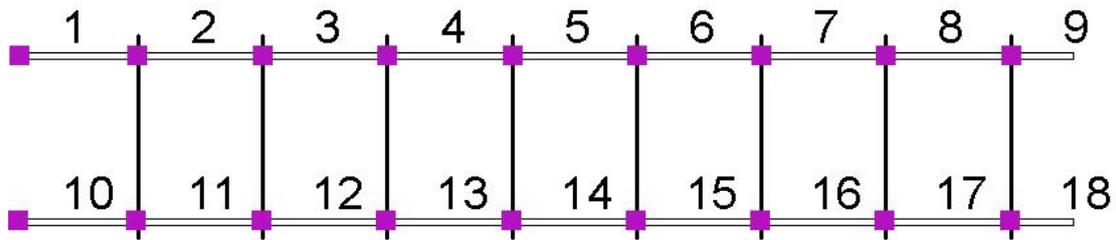


图 5-10 18 台探测车的初始位置分布

A、B 两组间隔一个时间步长 6s。考虑到 b 组进行探测时会对 C 组产生 <4.8km 的干扰,所以其工作时序必须间隔两个时间步长 12s,而我们依旧可以依次按照 A、B、C、D 组的顺序进行工作。此时一个工作周期所经历的时间如下:

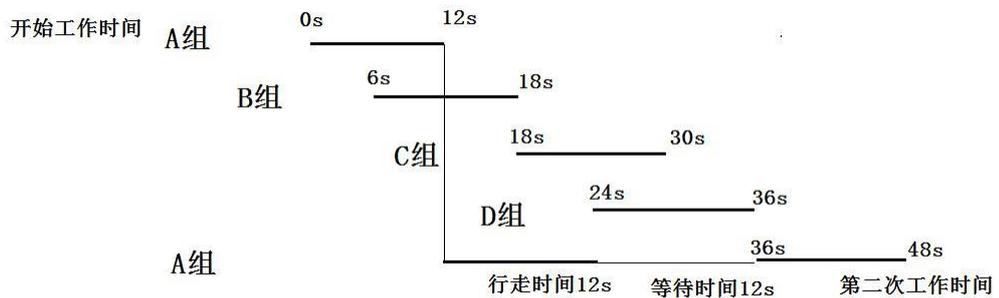


图 5-11 18 台车扫描时间图

**(2) 第二套布局**

考虑 20 台探测车的情况为了尽量避免探测车的相互影响,同时保证每组运行间隔都是一个步长 6s,我们只使用 16 台探测按照下图所示的划分区域其中 8 和 16 号区域的宽度为 300m,其余为 4500m 这样当所有探测车均位于其所分配区域的左下角时,1 号探测车与 3 号探测车距离  $L = 9\text{km}$ ,恰好不能影响到 3 区的探测车,3 区恰好不能影响到 5 区的探测车,5 区恰好不能影响到 7 区的探测车。

同理 9 区的探测车不会对 11 区的探测车产生影响，所以我们将 16 个区域分成 A、B、C、D 四组如下：

表 5-4 16 台车辆分组

组号	组员
A	1、3、5、7
B	9、11、13、15
C	2、4、6、8
D	10、12、14、16

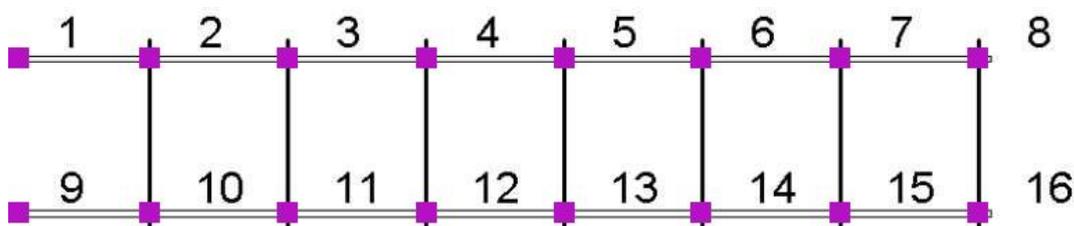


图 5-12 16 台探测车的初始位置分布

此方法我们依旧可以依次按照 A、B、C、D 组的顺序进行工作，每两组间隔一个步长 6s。此时一个工作周期所经历的时间如下：

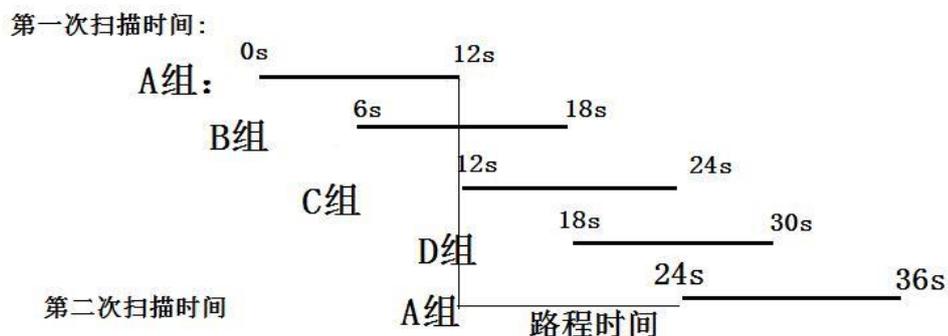


图 5-13 16 台探测车扫描时间图

### 总结

综合两个布局分析，同一时间段内，第一个布局扫描工作车辆最多，同样的区域，用时短，故选用第一个布局。

### 5.3.2 扫描路线

结合 12 台探测车的扫描路线分析，20 台车继续采用“弓”字形的移动轨迹。



D	移动	待机	待机	探测	探测	移动	探测	探测	移动
---	----	----	----	----	----	----	----	----	----

表 5-5 (d)

时间	162s	168s	174s	180s	186s	192s	198s	204s	210s
A	移动	移动	移动	探测	探测	移动	移动	待机	待机
B	移动	移动	移动	移动	探测	探测	移动	移动	待机
C	探测	移动	移动	移动	移动	探测	探测	移动	移动
D	探测	探测	移动	移动	移动	移动	探测	探测	移动

表 5-5 车辆状态表

在以上的工作时序中我们 1 个组每隔 26 个时间步长恰好探测完毕一个“I”型的路线，每隔 30 个时间步长恰好探测完毕一个“L”型的路线，正好完成了 1 列探测点的扫描，如图 5-5 所示。

我们继续采用纵向往复从左到右进行扫描运动。这是一种周期式的运动，每一个周期包含四段纵向扫描运动和一段横向扫描运动，呈现“L”型。

### 5.3.4 完成任务所花的最少时间

以 1 号探测车为例在纵向运动中 0s-144s 内小车的运动服从周期性运动规律，每个周期为一个节点的扫描时间 12s 加上路程运动时间 12s 加上等待时间 12s，如图 4-3 所示。

定义一个周期性纵向运动的时间为

$$t_{纵} = t_{扫} + t_{纵1} + t_{等} = 12s + 12s + 12s = 36s$$

将“L”型运动分为 4 个纵向运动和一个横向运动，因 100m 运动时间为 24s 故每一段横向运动的时间为一个节点扫描时间加上一个横向运动时间：

$$t_{横} = t_{扫} + t_{横1} = 12s + 24s = 36s$$

故每一个“L”型周期的运动时间为

$$T = 4t_{纵} + t_{横} = 4 \times 36 + 36 = 180s$$

定义一号车坐标系  $x_1, y_1$

故  $x_1, y_1$  可利用周期性运动规律，根据时刻  $t$  与时间  $T, t_{纵}, t_{扫}, t_{纵1}, t_{横}, t_{横1}, t_{等}$  的关系得出。

我们在 20=18tai/five1.m 文件中实现了此算法，其时序轨迹图如下：

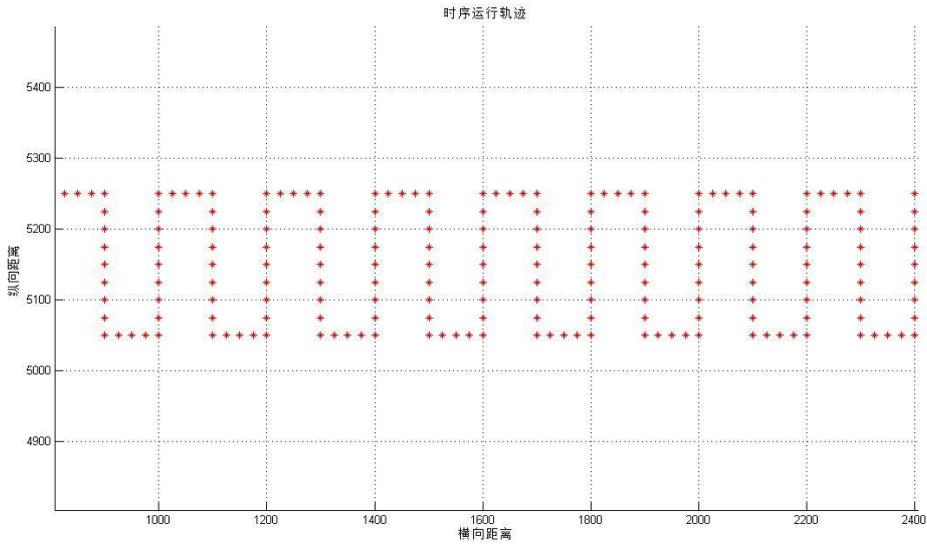


图 5-16 1 号探测车的时序轨迹图

图中显示的是 1 号探测车的时序轨迹图，以 6 秒为周期不断更新探测车的位置信息，形成该辆探测车的时序运行轨迹。

20 台探测车的情况下，时间计算方案与第一套模型相同，

在  $20=18tai/main.m$  中

输入 1 号车探测车的最终所在位置坐标  $x_1=3700$ ，  $y_1=5050$ ，

$$T_{总}=t+t_2$$

可得

$$T_{总}=t+24=6840s$$

## 5.4 最优探测车台数的布局模型

### 5.4.1 初始布局

首先我们假设:扫描时间  $t$  最小的情况下，当车辆最少时为效率最高。

通过多种模型下的扫描方式，我们发现只使用 16 台探测车时按照下图所示的划分区域 A1、B1、C1、D1 四组，可以交替进行下去不存在待机时间。

表 5-6 16 台车辆分组情况

组号	组员	间隔
A1	1、3、5、7	9km
B1	9、11、13、15	9km
C1	2、4、6、8	9km
D1	10、12、14、16	9km

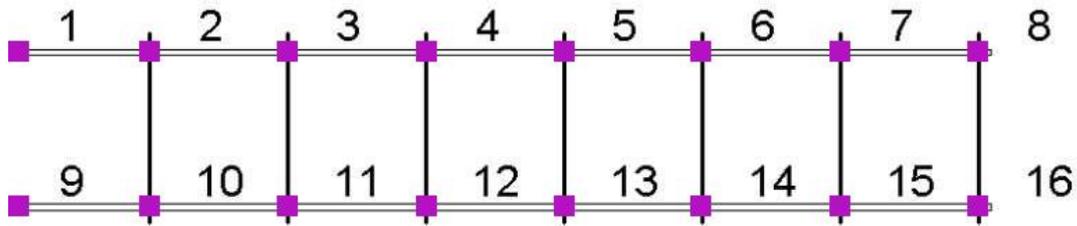


图 5-17 16 台探测车的初始位置分布

纵向周期性工作所经历的时间如下：

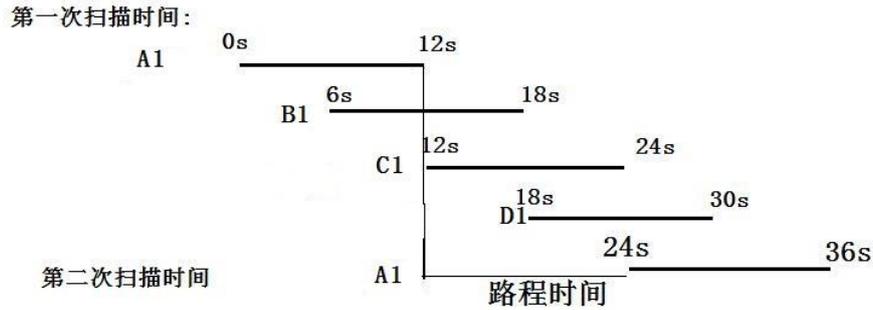


图 5-18 初始 16 台探测车扫描时间图

在“弓”型路线下，当探测车每进行一次横向运动时，间隔 24s 内多出 12s 不在工作，如图：

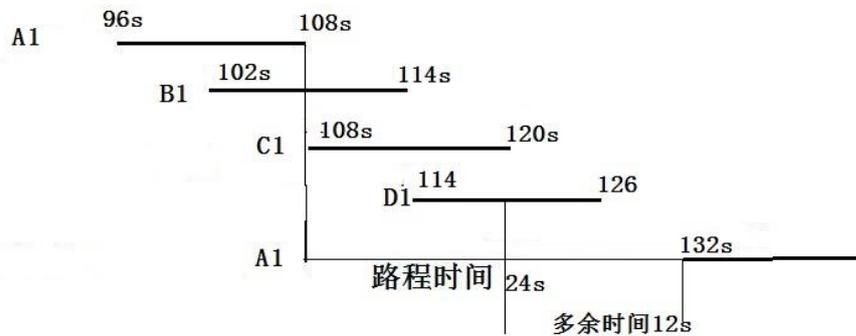


图 5-19 横向过渡 16 台探测车扫描时间图

在此多余时间内，通过分析，我们发现当在 A1、B1、C1、D1 横向内每两台车的中间加入一台车时，此时为 28 台，依次编号，新形成如下分配：

表 5-7 28 台车辆分组情况

组号	组员	间隔
A1	1、5、9、13	9.2km
B1	15、19、23、27	9.2km
C1	2、6、10、14	9.2km
D1	16、20、24、28	9.2km
A2	3、7、11	9.2km

B2	17、21、25	9.2km
C2	4、8、12	9.2km
D2	18、22、26	9.2km

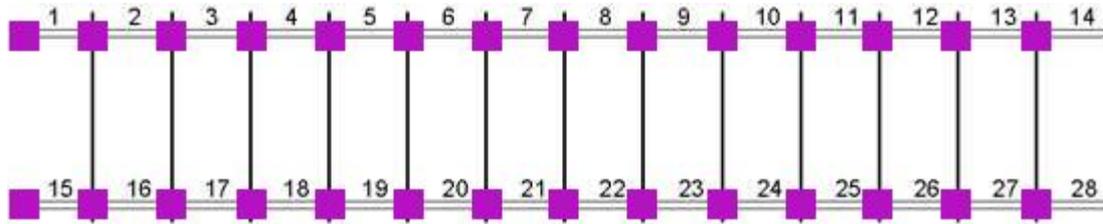


图 5-20 28 台探测车的初始位置分布

此时每组间隔 6s 扫描，交替进行，横向不存在多余时间。车辆达到饱和，为最优模型。

#### 5.4.2 扫描路线

结合 12 台探测车的扫描路线分析，28 台车继续采用“弓”字形的移动轨迹。

#### 5.4.3 扫描次序

A1 组探测车先开始扫描，扫描时间为 0-12s，每组扫描时间间隔为 12s。如此连续循环下去 A1、B1、C1、D1、A2、B2、C2、D2、A1、B1、C1……

#### 5.4.4 完成任务所花的最少时间

以 1 号探测车为例，在纵向运动中 0s-144s 内小车的运动服从周期性运动规律，每个周期为一个节点的扫描时间 12s 加上路程运动时间 12s 加上等待时间 12s，如图 4-3。定义一个周期性纵向运动的时间为

$$t_{纵} = t_{扫} + t_{纵1} + t_{等} = 12s + 12s + 24s = 48s$$

将“L”型运动分为 4 个纵向运动和一个横向运动，因 100m 运动时间为 24s 故每一段横向运动的时间为一个节点扫描时间加上一个横向运动时间加上一个等待时间：

$$t_{横} = t_{扫} + t_{横1} + t_{等} = 12s + 24s + 12s = 48s$$

故每一个“L”型周期的运动时间为

$$T = 4t_{纵} + t_{横} = 4 \times 48 + 48 = 240s$$

根据时刻 t 与时间 T,  $t_{纵}$ ,  $t_{扫}$ ,  $t_{纵1}$ ,  $t_{横}$ ,  $t_{横1}$ ,  $t_{等}$  的关系。

我们在 64,80 best=28tai/four1.m 文件中实现了此算法，其时序轨迹图如下：

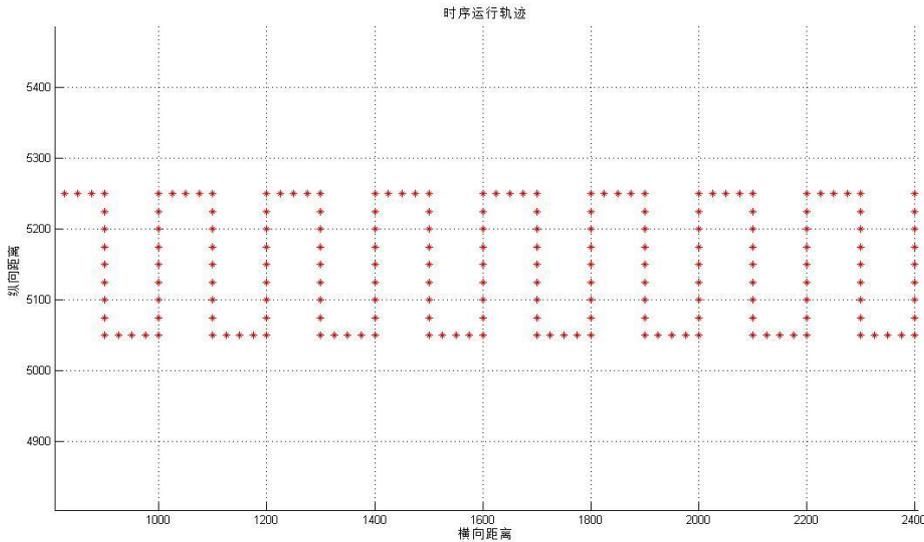


图 5-21 1 号探测车的时序轨迹图

图中显示的是 1 号探测车的时序轨迹图，以 6 秒为周期不断更新探测车的位置信息，形成该辆探测车的时序运行轨迹。

在 `64,80,best=28tai/main.m` 中输入 1 号车探测车的最终所在位置坐标

$x_1=2200$ ,  $y_1=5250$ ，可得 1 号车完成全部探测的时间  $t=5448s$ 。

故 28 台探测车的情况下，时间计算方案与第一套模型相同，故：

$$T_{\text{总}}=t+42=5490s$$

### 5.5 64、80 台探测车的布局模型

因需要投入 28 台探测车可以使完成任务的效率最高，故 64 与 80 台车的扫描过程都等同于 28 台。

## 五、模型评价与分析

模型一的优点是只要重复足够多的试验，就一定能获得最优解，并且所有的状态都能利用具体参数表示出来，便于计算机进行读入和分析，但是其计算所需的内存巨大，且计算效率低下，往往需要几万次的试验才能接近于真实最优解。

模型二通过从简单到复杂的条件，并按照一种较为简单又较为有效的移动方案，不断深入。通过分析平均一个时间间隔中处于探测状态的探测车的数量  $N$  和平均一个时间间隔中处于探测状态下的探测车占总探测车数量的比值  $Q$  两个参数随探测车总数  $2n$  变化的规律，来估计探测完成时所需的最短时间  $t_{\min}$ ，使所求解的问题获得了极大的简化，但是其方法只能对  $t_{\min}$  进行估计，难以获得真实的结果。

## 六、模型的改进

可以考虑将模型 1 和模型 2 方法综合起来，通过模型 2 的估计方法，对模型 1 的可行域进行简化，从而可以大大减少其仿真的试验数目。

## 七、结果总结

(1) 12 台探测车初始分布在 12 个均匀区域的左下角。行走路线为“弓”字型路线。均分四组扫描，每组间隔 6s。完成任务所需最少时间为 7128s。

20 台探测车只使用 18 台探测车，初始分布在 18 个区域的左下角使得河上和河下每相邻两台探测车间隔为 3.8km。行走路线依旧为“弓”字型路线。分四组扫描，前两组为每组 5 台，间隔 6s 扫描，后两组为每组 4 台，间隔 6s 扫描，前两组与后两组间隔 12s 扫描。完成任务所需最少时间为 6840s。

64 与 80 台车的扫描过程都等同于 28 台。28 台车初始分布在 28 个区域的左下角，使得河上和河下每相邻两台探测车间隔为 2.3km。行走路线同上为“弓”字型路线。分 8 组扫描，每组间隔 6s 依次进行扫描。完成任务所需最少时间为 5490s。

(2) 需要投入 28 台探测车可以使完成任务的效率最高。

(3) 为了方便计算机进行无线远程协调控制，将每台车的位置用矩阵形式表示，在程序 12tai;20=18tai;64,80,best=28 文件夹的 main.m 程序中得以实现，通过输入时刻 t 可以输出每台车的位置。计算机每 6s 发送行动指令，在每个子程序中已体现。

例：输入 t=1000,12 台车显示位置坐标如下：

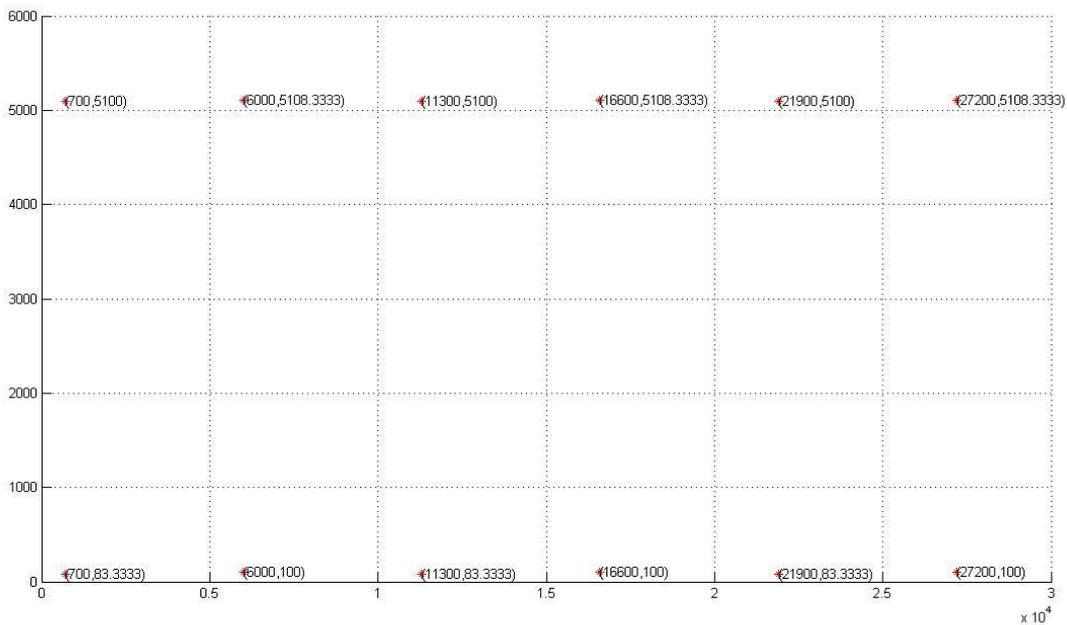


图 8-1 12 台探测车的位置坐标图

## 八、参考文献

- [1] 姜启源, 数学模型, 北京: 高等教育出版社, 2005
- [2] 龚纯, 精通 MATLAB 最优化计算, 北京: 电子工业出版社, 2009
- [3] 郭齐胜, 计算机仿真, 北京: 国防工业出版社, 2011

## 附录：

MATLAB 程序如下：

### 一. 12 台探测车

由位置求解当前时刻的主程序：

```
clc;
clear all;
t=0;
x(1)=0;
y(1)=0;
while
~(x(1)==5200&&y(1)==5250)==1
    t=t+6;
if t>=18
[x1,y1]=ru1(t);
[x2,y2]=ru2(t);
[x3,y3]=rd1(t);
[x4,y4]=rd2(t);
x=x1+x2+x3+x4;
y=y1+y2+y3+y4;
end
if t<12
    for n=1:6;
        x(n)=5300*(n-1);
        y(n)=5050;
    end
    for n=7:12;
        x(n)=5300*(n-7);
        y(n)=0;
    end
end
if t>=12 && t<18
    for n=1:2:5;
        x(n)=5300*(n-1);
        y(n)=(t-12)*50/12+5050;
    end
```

```
    for n=2:2:6;
        x(n)=5300*(n-1);
        y(n)=5050;
    end
    for n=7:12
        x(n)=5300*(n-7);
        y(n)=0;
    end
end
hold on
for i=1:12
    plot(x(i),y(i),'r-.*');
end
grid on
end
t=t+150;
xlabel('横向距离');
ylabel('纵向距离');
title('时序运行轨迹');
任意时刻探测车的位置:
clc;
clear all;
t=300;
if t>=18
[x1,y1]=ru1(t);
[x2,y2]=ru2(t);
[x3,y3]=rd1(t);
[x4,y4]=rd2(t);
x=x1+x2+x3+x4;
y=y1+y2+y3+y4;
end
if t<12
    for n=1:6;
        x(n)=5300*(n-1);
        y(n)=5050;
```

```

end
for n=7:12;
    x(n)=5300*(n-7);
    y(n)=0;
end
end
if t>=12 && t<18
    for n=1:2:5;
        x(n)=5300*(n-1);
        y(n)=(t-12)*50/12+5050;
    end
    for n=2:2:6;
        x(n)=5300*(n-1);
        y(n)=5050;
    end
    for n=7:12;
        x(n)=5300*(n-7);
        y(n)=0;
    end
end
hold on
for i=1:12
    plot(x(i),y(i),'r-.*');

text(x(i),y(i),['(',num2str(x(i)
),',',num2str(y(i)),')']);
end
grid on

rd1:
function [x,y]=rd1(t)
t2=t-12;
a=floor(t2/132);
b=rem(t2,132);
if b<=108;
    x(7)=a*100;
end
    if b>108;
        x(7)=a*100+100*(b-108)/24;
    end
c=floor(b/24);
d=rem(b,24);
e=rem(a,2);
if b<=96 && e==0 && d<=12;
    y(7)=c*50;
end
if b<=96 && e==0 && d>12;
    y(7)=c*50+(d-12)*50/12;
end
if b<=96 && e~=0 && d<=12;
    y(7)=200-c*50;
end
if b<=96 && e~=0 && d>12;
    y(7)=200-c*50-(d-
12)*50/12;
end
if b>96 && e==0;
    y(7)=200;
end
if b>96 && e~=0;
    y(7)=0;
end
for n=1:6;
    x(n)=0 ;
    y(n)=0;
end
for n=12;
    x(n)=0;
    y(n)=0;
end
end

```

```

for n=7:2:11;
    x(n)=x(7)+5300*(n-7);
    y(n)=y(7);
end

rd2:
function [x,y]=rd2(t)
t2=t-18;
a=floor(t2/132);
b=rem(t2,132);
if b<=108;
    x(8)=a*100+5300;
end
    if b>108;
        x(8)=a*100+100*(b-
108)/24+5300;
    end
c=floor(b/24);
d=rem(b,24);
e=rem(a,2);
if b<=96 && e==0 && d<=12;
    y(8)=c*50;
end
if b<=96 && e==0 && d>12;
    y(8)=c*50+(d-12)*50/12;
end
    if b<=96 && e~=0 && d<=12;
        y(8)=200-c*50;
    end
    if b<=96 && e~=0 && d>12;
        y(8)=200-c*50-(d-
12)*50/12;
    end
    if b>96 && e==0;
        y(8)=200;
    end
    if b>96 && e~=0;
        y(8)=0;
    end
for n=1:7;
    x(n)=0;
    y(n)=0;
end
for n=8:2:12;
    x(n)=x(8)+5300*(n-8);
    y(n)=y(8);
end

rul:
function [x y]=rul(t)
a=floor(t/132);
b=rem(t,132);
if b<=108;
    x(1)=a*100;
end
    if b>108;
        x(1)=a*100+100*(b-108)/24;
    end
c=floor(b/24);
d=rem(b,24);
e=rem(a,2);
if b<=96 && e==0 && d<=12;
    y(1)=c*50+5050;
end
    if b<=96 && e==0 && d>12;
        y(1)=c*50+(d-12)*50/12+5050;
    end
    if b<=96 && e~=0 && d<=12;
        y(1)=200-c*50+5050;
    end

```

```

end
if b<=96 && e~=0 && d>12;
    y(1)=200-c*50-(d-
12)*50/12+5050;
end
if b>96 && e==0;
    y(1)=200+5050;
end
if b>96 && e~=0;
    y(1)=0+5050;
end
for n=1:2:5;
    x(n)=x(1)+5300*(n-1);
    y(n)=y(1);
end
for n=6:12;
    x(n)=0;
    y(n)=0;
end

```

ru2:

```

function [x, y]=ru2(t)
t2=t-6;
a=floor(t2/132);
b=rem(t2, 132);
if b<=96;
    x(2)=a*100+5300;
end
if b>96;
    x(2)=a*100+100*(b-
108)/24+5300;
end
c=floor(b/24);
d=rem(b, 24);

```

```

e=rem(a, 2);
if b<=96 && e==0 && d<=12;
    y(2)=c*50+5050;
end
if b<=96 && e==0 && d>12;
    y(2)=c*50+(d-12)*50/12+5050;
end
if b<=96 && e~=0 && d<=12;
    y(2)=200-c*50+5050;
end
if b<=96 && e~=0 && d>12;
    y(2)=200-c*50-(d-
12)*50/12+5050;
end
if b>96 && e==0;
    y(2)=200+5050;
end
if b>96 && e~=0;
    y(2)=0+5050;
end
for n=2:2:6;
    x(n)=x(2)+5300*(n-2);
    y(n)=y(2);
end
for n=7:12;
    x(n)=0 ;
    y(n)=0;
end

```

二. 20 台探测车

由位置求解当前时刻的主程序:

```

clc;
clear all;
t=0;

```

```

x(1)=0;
y(1)=0;
while ~ (x(1)==3700&& y(1)==5050) ==1
    t=t+6;
    if t>=24
        [x1,y1]=five1(t);
        [x2,y2]=five2(t);
        [x3,y3]=four1(t);
        [x4,y4]=four2(t);
        x=x1+x2+x3+x4;
        y=y1+y2+y3+y4;
    end
    if t<12
        for n=1:9;
            x(n)=3800*(n-1);
            y(n)=5050;
        end
        for n=10:18;
            x(n)=3800*(n-10);
            y(n)=0;
        end
    end
    if t>=12 && t<18
        for n=1:4:9;
            x(n)=3800*(n-1);
            y(n)=(t-12)*50/12+5050;
        end
        for n=12:4:16;
            x(n)=3800*(n-10);
            y(n)=(t-12)*50/12;
        end
        for n=3:4:7;
            x(n)=3800*(n-1);
            y(n)=(t-18)*50/12+5050;
        end
        for n=10:4:18;
            x(n)=3800*(n-10);
            y(n)=(t-18)*50/12;
        end
    end
    if t>=12 && t<18
        for n=1:4:9;
            x(n)=3800*(n-1);
            y(n)=(t-12)*50/12+5050;
        end
        for n=12:4:16;
            x(n)=3800*(n-10);
            y(n)=(t-12)*50/12;
        end
        for n=[2,3,4,6,7,8];
            x(n)=3800*(n-1);
            y(n)=5050;
        end
    end
end
n=[10,11,13,14,15,17,18];
x(n)=3800*(n-10);
y(n)=0;
end
if t>=18 && t<24
    for n=1:4:9;
        x(n)=3800*(n-1);
        y(n)=(t-12)*50/12+5050;
    end
    for n=12:4:16;
        x(n)=3800*(n-10);
        y(n)=(t-12)*50/12;
    end
    for n=3:4:7;
        x(n)=3800*(n-1);
        y(n)=(t-18)*50/12+5050;
    end
    for n=10:4:18;
        x(n)=3800*(n-10);
        y(n)=(t-18)*50/12;
    end
end
for n=[2,4,6,8];
    x(n)=3800*(n-1);
    y(n)=5050;
end
for n=[11,13,15,17];
    x(n)=3800*(n-10);
    y(n)=0;
end
end
hold on
for i=1:12

```

```

        plot(x(i),y(i),'r-.*');
end
grid on
end
t=t+24;
xlabel(' 横向距离' ) ;
ylabel(' 纵向距离' ) ;
title(' 时序运动轨迹' ) ;

任意时刻探测车的位置:
clc;
clear all;
t=20;
if t>=24
[x1,y1]=five1(t);
[x2,y2]=five2(t);
[x3,y3]=four1(t);
[x4,y4]=four2(t);
x=x1+x2+x3+x4;
y=y1+y2+y3+y4;
end
if t<12
    for n=1:9;
        x(n)=3800*(n-1);
        y(n)=5050;
    end
    for n=10:18;
        x(n)=3800*(n-10);
        y(n)=0;
    end
end
if t>=12 && t<18
    for n=1:4:9;
        x(n)=3800*(n-1);
        y(n)=(t-12)*50/12+5050;
    end
    for n=12:4:16;
        x(n)=3800*(n-10);
        y(n)=(t-12)*50/12;
    end
    for n=[2, 3, 4, 6, 7, 8];
        x(n)=3800*(n-1);
        y(n)=5050;
    end
    for
n=[10, 11, 13, 14, 15, 17, 18];
        x(n)=3800*(n-10);
        y(n)=0;
    end
    if t>=18 && t<24
        for n=1:4:9;
            x(n)=3800*(n-1);
            y(n)=(t-12)*50/12+5050;
        end
        for n=12:4:16;
            x(n)=3800*(n-10);
            y(n)=(t-12)*50/12;
        end
        for n=3:4:7;
            x(n)=3800*(n-1);
            y(n)=(t-18)*50/12+5050;
        end
        for n=10:4:18;
            x(n)=3800*(n-10);
            y(n)=(t-18)*50/12;
        end
        for n=[2, 4, 6, 8];
            x(n)=3800*(n-1);

```

```

        y(n)=5050;
    end
    for n=[11, 13, 15, 17];
        x(n)=3800*(n-10);
        y(n)=0;
    end
end
hold on
for i=1:12
    plot(x(i),y(i), 'r-.*');

text(x(i),y(i), ['(', num2str(x(i)), ', ', num2str(y(i)), ')']);
end
grid on

function [x y]=fivel(t)
a=floor(t/180);
b=rem(t, 180);
if b<=156;
    x(1)=a*100;
end
    if b>156;
        x(1)=a*100+100*(b-156)/24;
    end
c=floor(b/36);
d=rem(b, 36);
e=rem(a, 2);
if b<=156 && e==0 && d<=12;
    y(1)=c*50+5050;
end
if b<=156 && e==0 && d>12 &&
d<=24;
        y(1)=5050;
    end
    if b<=156 && e==0 && d>24;
        y(1)=c*50+5100;
    end
    if b<=156 && e~0 && d<=12;
        y(1)=200-c*50+5050;
    end
    if b<=156 && e~0 && d>12 &&
d<=24;
        y(1)=200-c*50-(d-
12)*50/12+5050;
    end
    if b<=156 && e~0 && d>24;
        y(1)=150-c*50+5050;
    end
    if b>156 && e==0;
        y(1)=200+5050;
    end
    if b>156 && e~0;
        y(1)=0+5050;
    end
    for n=1:4:9;
        x(n)=x(1)+3800*(n-1);
        y(n)=y(1);
    end
    for n=12:4:16;
        x(n)=x(1)+3800*(n-10);
        y(n)=y(1)-5050;
    end
    for n=17:18;
        x(n)=0;
        y(n)=0;
    end
end

```

```

function [x, y]=five2(t)
t2=t-6;
a=floor(t2/180);
b=rem(t2, 180);
if b<=156;
    x(3)=a*100+7600;
end
    if b>156;
        x(3)=a*100+100*(b-
156)/24+7600;
    end
c=floor(b/36);
d=rem(b, 36);
e=rem(a, 2);
if b<=156 && e==0 && d<=12;
    y(3)=c*50+5050;
end
if b<=156 && e==0 && d>12 &&
d<=24;
    y(3)=c*50+(d-12)*50/12+5050;
end
if b<=156 && e==0 && d>24;
    y(3)=c*50+5100;
end
    if b<=156 && e~0 && d<=12;
        y(3)=200-c*50+5050;
    end
    if b<=156 && e~0 && d>12 &&
d<=24;
        y(3)=200-c*50-(d-
12)*50/12+5050;
    end
    if b<=156 && e~0 && d>24;
        y(3)=150-c*50+5050;
    end
end
    if b>156 && e==0;
        y(3)=200+5050;
    end
    if b>156 && e~0;
        y(3)=0+5050;
    end
for n=3:4:7;
    x(n)=x(3)+3800*(n-3);
    y(n)=y(3);
end
for n=10:4:18;
    x(n)=x(3)+3800*(n-12) ;
    y(n)=y(3)-5050;
end

function [x, y]=four1(t)
t2=t-18;
a=floor(t2/180);
b=rem(t2, 180);
if b<=156;
    x(2)=a*100+3800;
end
    if b>156;
        x(2)=a*100+100*(b-
156)/24+3800;
    end
c=floor(b/36);
d=rem(b, 36);
e=rem(a, 2);
if b<=156 && e==0 && d<=12;
    y(2)=c*50+5050;
end
if b<=156 && e==0 && d>12 &&

```

```

d<=24;
    y(2)=c*50+(d-12)*50/12+5050;
end
if b<=156 && e==0 && d>24;
    y(2)=c*50+5100;
end
if b<=156 && e~0 && d<=12;
    y(2)=200-c*50+5050;
end
if b<=156 && e~0 && d>12 &&
d<=24;
    y(2)=200-c*50-(d-
12)*50/12+5050;
end
if b<=156 && e~0 && d>24;
    y(2)=150-c*50+5050;
end
if b>156 && e==0;
    y(2)=200+5050;
end
if b>156 && e~0;
    y(2)=0+5050;
end
for n=2:4:6;
    x(n)=x(2)+3800*(n-2);
    y(n)=y(2);
end
for n=13:4:17;
    x(n)=x(2)+3800*(n-11);
    y(n)=y(2)-5050;
end
for n=18;
    x(n)=0;
    y(n)=0;
end

function [x, y]=four2(t)
t2=t-24;
a=floor(t2/180);
b=rem(t2, 180);
if b<=156;
    x(4)=a*100+11400;
end
if b>132;
    x(4)=a*100+100*(b-
132)/24+11400;
end
c=floor(b/36);
d=rem(b, 36);
e=rem(a, 2);
if b<=156 && e==0 && d<=12;
    y(4)=c*50+5050;
end
if b<=156 && e==0 && d>12 &&
d<=24;
    y(4)=c*50+(d-12)*50/12+5050;
end
if b<=156 && e==0 && d>24;
    y(4)=c*50+5100;
end
if b<=156 && e~0 && d<=12;
    y(4)=200-c*50+5050;
end
if b<=156 && e~0 && d>12 &&
d<=24;
    y(4)=200-c*50-(d-
12)*50/12+5050;
end
if b<=156 && e~0 && d>24;

```

```

        y(4)=150-c*50+5050;
    end
    if b>156 && e==0;
        y(4)=200+5050;
    end
    if b>156 && e~=0;
        y(4)=0+5050;
    end
    for n=4:4:8;
        x(n)=x(4)+3800*(n-4);
        y(n)=y(4);
    end
    for n=11:4:15;
        x(n)=x(4)+3800*(n-13);
        y(n)=y(4)-5050;
    end
    for n=16:18;
        x(n)=0;
        y(n)=0;
    end
end

[x2, y2]=four2(t);
[x3, y3]=four3(t);
[x4, y4]=four4(t);
[x5, y5]=three5(t);
[x6, y6]=three6(t);
[x7, y7]=three7(t);
[x8, y8]=three8(t);
x=x1+x2+x3+x4+x5+x6+x7+x8;
y=y1+y2+y3+y4+y5+y6+y7+y8;
end
if t<12
    for n=1:14;
        x(n)=2300*(n-1);
        y(n)=5050;
    end
    for n=15:28;
        x(n)=2300*(n-10);
        y(n)=0;
    end
end
hold on
for i=1:12
    plot(x(i), y(i), 'r-.*');
end
grid on
end
end

```

### 三. 28 台探测车:

由位置求解当前时刻的主程序:

```

clc;
clear all;
t=0;
x(1)=0;
y(1)=0;
while
~(x(1)==2200&&y(1)==5250)==1
    t=t+6;
    if t>=42;
        [x1, y1]=four1(t);

```

任意时刻探测车的位置:

```

clc;
clear all;
t=60;
if t>=42;
    [x1, y1]=four1(t);
    [x2, y2]=four2(t);
    [x3, y3]=four3(t);

```

```

[x4, y4]=four4(t);
[x5, y5]=three5(t);
[x6, y6]=three6(t);
[x7, y7]=three7(t);
[x8, y8]=three8(t);
x=x1+x2+x3+x4+x5+x6+x7+x8;
y=y1+y2+y3+y4+y5+y6+y7+y8;
end
if t<12
    for n=1:14;
        x(n)=2300*(n-1);
        y(n)=5050;
    end
    for n=15:28;
        x(n)=2300*(n-10);
        y(n)=0;
    end
end
hold on
for i=1:12
    plot(x(i), y(i), 'r-.*');

text(x(i), y(i), ['(', num2str(x(i)
), ', ', num2str(y(i)), ')']);
end
grid on

function [x y]=four1(t)
a=floor(t/240);
b=rem(t, 240);
if b<=204;
    x(1)=a*100;
end
    if b>204 && b<=228;
        x(1)=a*100+100*(b-204)/24;
    end
    if b>228;
        x(1)=a*100+100;
    end
c=floor(b/48);
d=rem(b, 48);
e=rem(a, 2);
if b<=192 && e==0 && d<=12;
    y(1)=c*50+5050;
end
if b<=192 && e==0 && d>12 &&
d<=24;
    y(1)=c*50+(d-12)*50/12+5050;
end
if b<=192 && e==0 && d>24;
    y(1)=c*50+5100;
end
    if b<=192 && e~=0 && d<=12;
        y(1)=200-c*50+5050;
    end
    if b<=192 && e~=0 && d>12 &&
d<=24;
        y(1)=200-c*50-(d-
12)*50/12+5050;
    end
    if b<=192 && e~=0 && d>24;
        y(1)=150-c*50+5050;
    end
end
    if b>192 && e==0;
        y(1)=200+5050;
    end
end
    if b>192 && e~=0;
        y(1)=0+5050;
    end
end

```

```

end
for n=1:4:13;
    x(n)=x(1)+2300*(n-1);
    y(n)=y(1);
end
for n=14:28;
    x(n)=0;
    y(n)=0;
end

function [x,y]=four2(t)
t2=t-6;
a=floor(t2/240);
b=rem(t2,240);
if b<=204;
    x(15)=a*100;
end
if b>204 && b<=228;
    x(15)=a*100+100*(b-204)/24;
end
if b>228;
    x(15)=a*100+100;
end
c=floor(b/48);
d=rem(b,48);
e=rem(a,2);
if b<=192 && e==0 && d<=12;
    y(15)=c*50;
end
if b<=192 && e==0 && d>12 &&
d<=24;
    y(15)=c*50+(d-12)*50/12;
end

if b<=192 && e==0 && d>24;
    y(15)=c*50+50;
end
if b<=192 && e~=0 && d<=12;
    y(15)=200-c*50;
end
if b<=192 && e~=0 && d>12 &&
d<=24;
    y(15)=200-c*50-(d-
12)*50/12;
end
if b<=192 && e~=0 && d>24;
    y(15)=150-c*50;
end
if b>192 && e==0;
    y(15)=200;
end
if b>192 && e~=0;
    y(15)=0;
end
for n=15:4:27;
    x(n)=x(15)+2300*(n-15);
    y(n)=y(15);
end
for n=1:14;
    x(n)=0 ;
    y(n)=0;
end
for n=28;
    x(n)=0 ;
    y(n)=0;
end
end

```

```

function [x, y]=four3(t)
t2=t-12;
a=floor(t2/240);
b=rem(t2, 240);
if b<=204;
    x(2)=a*100+2300;
end
    if b>204 && b<=228;
        x(2)=a*100+100*(b-
204)/24+2300;
    end
    if b>228;
        x(2)=a*100+100+2300;
    end
c=floor(b/48);
d=rem(b, 48);
e=rem(a, 2);
if b<=192 && e==0 && d<=12;
    y(2)=c*50+5050;
end
if b<=192 && e==0 && d>12 &&
d<=24;
    y(2)=c*50+(d-12)*50/12+5050;
end
if b<=192 && e==0 && d>24;
    y(2)=c*50+5100;
end
    if b<=192 && e~0 && d<=12;
        y(2)=200-c*50+5050;
    end
    if b<=192 && e~0 && d>12 &&
d<=24;
        y(2)=200-c*50-(d-
12)*50/12+5050;
    end
    if b<=192 && e~0 && d>24;
        y(2)=150-c*50+5050;
    end
end
if b>192 && e==0;
    y(2)=200+5050;
end
if b>192 && e~0;
    y(2)=0+5050;
end
for n=2:4:14;
    x(n)=x(2)+2300*(n-2);
    y(n)=y(2);
end
for n=1;
    x(n)=0;
    y(n)=0;
end
for n=15:28;
    x(n)=0;
    y(n)=0;
end
end

function [x, y]=four4(t)
t2=t-18;
a=floor(t2/240);
b=rem(t2, 240);
if b<=204;
    x(16)=a*100+2300;
end
    if b>204 && b<=228;
        x(16)=a*100+100*(b-
204)/24+2300;
    end
end

```

```

    if b>228;
        x(16)=a*100+100+2300;
    end
c=floor(b/48);
d=rem(b, 48);
e=rem(a, 2);
if b<=192 && e==0 && d<=12;
    y(16)=c*50;
end
if b<=192 && e==0 && d>12 &&
d<=24;
    y(16)=c*50+(d-12)*50/12;
end
if b<=192 && e==0 && d>24;
    y(16)=c*50+50;
end
if b<=192 && e~=0 && d<=12;
    y(16)=200-c*50;
end
if b<=192 && e~=0 && d>12 &&
d<=24;
    y(16)=200-c*50-(d-
12)*50/12;
end
if b<=192 && e~=0 && d>24;
    y(16)=150-c*50;
end
if b>192 && e==0;
    y(16)=200;
end
if b>192 && e~=0;
    y(16)=0;
end
for n=16:4:28;
    x(n)=x(16)+2300*(n-16);
    y(n)=y(16);
end
for n=1:15;
    x(n)=0;
    y(n)=0;
end
function [x y]=three5(t)
t2=t-24;
a=floor(t2/240);
b=rem(t2, 240);
if b<=204;
    x(3)=a*100+4600;
end
if b>204 && b<=228;
    x(3)=a*100+100*(b-
204)/24+4600;
end
if b>228;
    x(3)=a*100+100+4600;
end
c=floor(b/48);
d=rem(b, 48);
e=rem(a, 2);
if b<=192 && e==0 && d<=12;
    y(3)=c*50+5050;
end
if b<=192 && e==0 && d>12 &&
d<=24;
    y(3)=c*50+(d-12)*50/12+5050;
end
if b<=192 && e==0 && d>24;
    y(3)=c*50+5100;
end

```

```

    if b<=192 && e~=0 && d<=12;
        y(3)=200-c*50+5050;
    end
    if b<=192 && e~=0 && d>12 &&
d<=24;
        y(3)=200-c*50-(d-
12)*50/12+5050;
    end
    if b<=192 && e~=0 && d>24;
        y(3)=150-c*50+5050;
    end
    if b>192 && e==0;
        y(3)=200+5050;
    end
    if b>192 && e~=0;
        y(3)=0+5050;
    end
    for n=3:4:11;
        x(n)=x(3)+2300*(n-3);
        y(n)=y(3);
    end
    for n=1:2;
        x(n)=0;
        y(n)=0;
    end
    for n=12:28;
        x(n)=0;
        y(n)=0;
    end
end

function [x,y]=three6(t)
t2=t-30;
a=floor(t2/240);
b=rem(t2,240);
if b<=204;
    x(17)=a*100+4600;
end
    if b>204 && b<=228;
        x(17)=a*100+100*(b-
204)/24+4600;
    end
    if b>228;
        x(17)=a*100+100+4600;
    end
c=floor(b/48);
d=rem(b,48);
e=rem(a,2);
if b<=192 && e==0 && d<=12;
    y(17)=c*50;
end
if b<=192 && e==0 && d>12 &&
d<=24;
    y(17)=c*50+(d-12)*50/12;
end
if b<=192 && e==0 && d>24;
    y(17)=c*50+50;
end
    if b<=192 && e~=0 && d<=12;
        y(17)=200-c*50;
    end
    if b<=192 && e~=0 && d>12 &&
d<=24;
        y(17)=200-c*50-(d-
12)*50/12;
    end
    if b<=192 && e~=0 && d>24;
        y(17)=150-c*50;
    end
end

```

```

if b>192 && e==0;
    y(17)=200;
end
if b>192 && e~ =0;
    y(17)=0;
end
for n=17:4:25;
    x(n)=x(17)+2300*(n-17);
    y(n)=y(17);
end
for n=1:16;
    x(n)=0 ;
    y(n)=0;
end
for n=26:28;
    x(n)=0 ;
    y(n)=0;
end

function [x,y]=three7(t)
t2=t-36;
a=floor(t2/240);
b=rem(t2, 240);
if b<=204;
    x(4)=a*100+6900;
end
if b>204 && b<=228;
    x(4)=a*100+100*(b-
204)/24+6900;
end
if b>228;
    x(4)=a*100+100+6900;
end
c=floor(b/48);
d=rem(b, 48);
e=rem(a, 2);
if b<=192 && e==0 && d<=12;
    y(4)=c*50+5050;
end
if b<=192 && e==0 && d>12 &&
d<=24;
    y(4)=c*50+(d-12)*50/12+5050;
end
if b<=192 && e==0 && d>24;
    y(4)=c*50+5100;
end
if b<=192 && e~ =0 && d<=12;
    y(4)=200-c*50+5050;
end
if b<=192 && e~ =0 && d>12 &&
d<=24;
    y(4)=200-c*50-(d-
12)*50/12+5050;
end
if b<=192 && e~ =0 && d>24;
    y(4)=150-c*50+5050;
end
if b>192 && e==0;
    y(4)=200+5050;
end
if b>192 && e~ =0;
    y(4)=0+5050;
end
for n=4:4:12;
    x(n)=x(4)+2300*(n-4);
    y(n)=y(4);
end
for n=1:3;
    x(n)=0;

```

```

        y(n)=0;
    end
    for n=13:28;
        x(n)=0;
        y(n)=0;
    end

function [x,y]=three8(t)
t2=t-42;
a=floor(t2/240);
b=rem(t2,240);
if b<=204;
    x(18)=a*100+6900;
end
if b>204 && b<=228;
    x(18)=a*100+100*(b-
204)/24+6900;
end
if b>228;
    x(18)=a*100+100+6900;
end
c=floor(b/48);
d=rem(b,48);
e=rem(a,2);
if b<=192 && e==0 && d<=12;
    y(18)=c*50;
end
if b<=192 && e==0 && d>12 &&
d<=24;
    y(18)=c*50+(d-12)*50/12;
end
if b<=192 && e==0 && d>24;
    y(18)=c*50+50;
end

        if b<=192 && e~0 && d<=12;
            y(18)=200-c*50;
        end
        if b<=192 && e~0 && d>12 &&
d<=24;
            y(18)=200-c*50-(d-
12)*50/12;
        end
        if b<=192 && e~0 && d>24;
            y(18)=150-c*50;
        end
        if b>192 && e==0;
            y(18)=200;
        end
        if b>192 && e~0;
            y(18)=0;
        end
        for n=18:4:26;
            x(n)=x(18)+2300*(n-18);
            y(n)=y(18);
        end
        for n=1:17;
            x(n)=0;
            y(n)=0;
        end
        for n=27:28;
            x(n)=0;
            y(n)=0;
        end
end
end

```