官方微博: http://weibo.com/304456943

报名号 #1963 2008年 第一届"数学中国杯"

数学建模网络挑战赛

承 诺 书

我们仔细阅读了首届"数学中国杯"数学建模网络挑战赛的竞赛规则。

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。

我们允许数学中国网站(www.madio.net)公布论文,以供网友之间学习交流,数学中 国网站以非商业目的的论文交流不需要提前取得我们的同意。

我们的参赛报名号为:

参赛队员 (签名):

队员 1: 张晓祥

队员 2: 陈尧昕

队员3: 邱冰

参赛队教练员 (签名):

参赛队伍组别:大学组

报名号 #1963 2008年 第一届"数学中国杯"

数学建模网络挑战赛

编号专用页

赛队伍的参赛号码: (请各个参赛队提前填写好): 1963

竞赛统一编号(由竞赛组委会送至评委团前编号):

Ser Many Madrio

竞赛评阅编号(由竞赛评委团评阅前进行编号):

 \Box

豇

报名号 # 1963

2008年 第一届"数学中国杯" 数学建模网络挑战赛

不切对法游戏

		Ashavi manis	~	
关 键 词	分类讨论,	打中概率,	飞机函数坐标模型,	穷举均
分法, 打点	函数关系法			

摘 要:

本文就如何用最少的次数打中机头建立数学模型。

问题一是针对 7×7的表格,一架飞机,通过均分法和打点函数关系法对问题进行不同角度的分析和求解;

问题二是针对 9×9 的表格, 两架飞机, 在问题一的基础上建立模型, 但是因为格子数增多, 飞机数量加一, 故问题变得更复杂, 单纯的通过穷举法计算量过大, 不现实, 所以改进了打点函数关系法, 使其适应大面积的计算, 在策略上注重选取所打点的位置来减少分析的复杂度, 先单独分析一架飞机再组合起来, 完成对机头的寻找;

综合上述方法均分法有点类似于穷举法,当数据量加大是不易在进行计算,而打点函数关系法,因为在整个图表中建立了一定的函数关系,可以在一定程度上简化问题。

参赛队号 1963

数学中国YY网校频道:159214

所选题目 飞机对战游戏

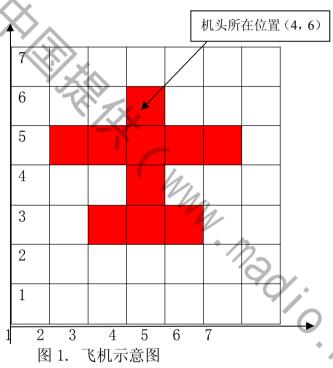
参赛密码

(由组委会填写)

报名号 # **1963** 1、问题重述

问题重述:有一种在学生中间比较流行的双方对战游戏。在游戏前双方各准备一张坐标纸,在上面分别制作7×7的方格,如图1所示。在自己的方格中画一架飞机,飞机呈"士"字形,其中上面的一长横占5个格子,下面的短横占3个格子,一竖占4个格子,最上面突出的一个格子代表机头。所画飞机的位置以及机头的指向由游戏者自己决定,游戏结束前双方不能互看对方的坐标纸。游戏时双方交替用"炮弹"打击对方,攻击的一方报告"炮弹"打击的位置,被攻击的一方报告是否命中飞机。例如:被攻击方的飞机画法如图1所示,攻击者报告"炮弹"的打击位置是(4,3),从图中可知,"炮弹"恰好落在飞机所在的红色格子上面,被攻击方报告飞机被击中,接下来刚才的被攻击方变成攻击方进行上面的攻击步骤,双方交替攻击对方,如果某一方被命中机头,游戏结束,被命中机头的一方失败。游戏双方都在通过打击后对方的反馈信息来猜测对方飞机的位置。

游戏比赛采用19局10胜制。



问题一:设计一个人机对战的"飞机对战"游戏。要求先由计算机进行攻击,以取胜为目标,给出进行游戏的策略。

问题二:考虑在 9×9 坐标纸上画两架飞机的游戏方式,两架飞机所占的格子不能重合,游戏方法同上。其中一架飞机被命中机头时要报告有一架飞机被击落。当某方的两架飞机都被击落时游戏结束,被击落方失败。分析这种游戏方式与只画一架飞机的游戏方式在策略上的不同点。

2、符号说明

- 2.1(x, y)——机头所在位置的坐标
- 2.2 i——每次所打击位置的横坐标
- 2.3 i——每次所打击位置的纵坐标
- 2.4 ss——每次打击后造成的结果, ss=0 表示未击中, ss=1 表示击中, ss=2 表示打中机头
- 2.5 fa, a 为标志量,通过他们为 0 或为 1 时来给矩阵中的每个格子赋值

3、模型建立

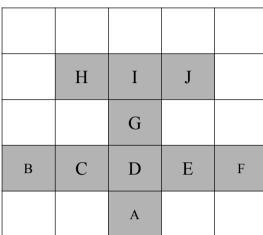
由于此题是在坐标中给出飞机的模型,故用十个坐标来表示飞机体的具体位置,但因为飞机的方向问题,所以建立四个方向的模型,用其机头的坐标(X,Y)为标准来表示飞机模型,即:

		A		
В	C	D	Е	F
		G		
4	Ü	I	J	
		42	S	

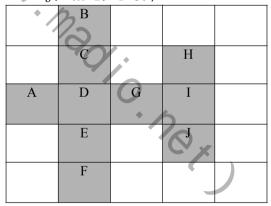
- A: (X, Y):
- B: (X-2, Y-1);
- C: (X-1, Y-1):
- D: (X, Y-1):
- E: (X+1, Y-1);
- F: (X+2, Y-1):
- G: (X, Y-2);
- H: (X-1, Y-3):
- I: (X, Y-3);
- J: (X+1, Y-3):

		В		
Н		С		
I	G	D	A	
J		Е		
		F		

- A: (X, Y):
- B: (X-1, Y+2);
- C: (X-1, Y+1);
- D: (X-1, Y);
- E: (X-1, Y-1):
- F: (X-1, Y-2);
- G: (X-2, Y);
- H: (X-3, Y+1):
- I: (X-3, Y);
- J: (X-3, Y-1);



- A: (X, Y);
- B: (X-2, Y+1);
- C: (X-1, Y+1):
- D: (X, Y+1):
- E: (X+1, Y+1);
- F: (X+2, Y+1);
- G: (X, Y+2);
- H: (X-1, Y+3):
- I: (X, Y+3);
- J: (X+1, Y+3);



- A: (X, Y):
- B: (X+1, Y+2);
- C: (X+1, Y+1);
- D: (X+1, Y):
- E: (X+1, Y-1);
- F: (X+1, Y-2);
- G: (X+2, Y);
- H: (X+3, Y+1);
- I: (X+3, Y):
- J: (X+3, Y-1);

官方微博:http://weibo.com/304456943

报名号 #1963

模型一的取值范围是 X(-[3,5], Y(-[4,7];

模型二的取值范围是 X(-[3,5], Y(-[1,4]:

模型三的取值范围是 X(-[1,4], Y(-[3,5];

模型四的取值范围是 X(-[4,7], Y(-[3,5];

可计算出一共存在的飞机排列方法有 4*4*(5-3+1) =48 种;

用 Matlab 对模型进行分析, 计算出每一点机身和机头经过的次数, 得出:

用 Matlab 对模型:	进行分	析,让	十算出	每一点	机身和	印机头	经过的
47		2	5	6	5	2	
747	2	6	12	14	12	6	2
X	5	12	22	26	22	12	5
	6	14	26	32	26	14	6
	5	12-	22	26	22	12	5
	2	6	12	14	12	6	2
		2	5	16/1	5	2	

每一点机身经过的次数

					A		
		1	1	1	9		
		1	1	1		0	
1	1	3	2	3	1	1	20
1	1	2	4	2	1	1	(C)
1	1	3	2	3	1	1	
		1	1	1			
		1	1	1			

每一点机头经过的次数

一、均分法

先通过对每种飞机的模型情况进行分析,排列出每个点上经过的次数,尽量取出经过该点的概率为 1/2 的点,如果没有找到在对其分裂成两份继续按相同的方法进行寻找,运算次数理论上能为 Log2[n],但计算量大,适应于小型数据量:

具体方法:

根据表格中的数据,可知第一次应打在坐标为(4,4)位置,通过参赛者的反馈可知道是否打中,则分为三种情况:

- 1) ss=2, 即打中机头, 游戏结束。
- 2) ss=1, 即打中飞机。
- 3) ss=0, 即没有打中飞机。

利用均分法,对飞机可能出现的模型分别进行分析,如先对 (d) 中的飞机模型进行具体的分析,此模型机头的横纵坐标范围是 $3 \le x \le 5$, $4 \le y \le 7$, $x \le y$ 的范围为每次发射提供了依据,为结果的判断提供循环次数。

构建一个 7×7 的矩阵,设其中的每个元素均为 0,通过排除法及标志量从所有可能的飞机模型中将不可能出现的飞机模型排除。

如果没有打中(ss=0),通过编程把发射的格子上可能存在飞机的模型排除掉,找出经过这个格子可能存在的所有的飞机,通过循环,进一步进行打击目标,再将不可能出现的飞机模型排除掉,如果飞机是(d)模型的,此时就可以找到参赛者的飞机模型,将飞机各部位所在的格子的值加 1(将值由 0 变为 1),那么7×7的矩阵中由 1 所组成的图形就是要找的飞机模型。如果飞机不是(d)模型的,那么向下进行,将机头 x 的范围变为 $1 \le x \le 4$, $3 \le y \le 5$,即转到(c)模型,与上面相同进行排除,赋值,寻找目标飞机所在的位置,如果没有找到,再通过改变 x、y 的范围,转到(a)模型,(b)模型,对以上进行循环,通过游戏者是否打中 ss 值得反馈对标志量进行赋值和修改,从而对 7×7 矩阵中的每个格子进行赋值,通过多次的排除,直到找到目标飞机的位置,由于我们已经设出机头所在位置的坐标为(x, y),游戏结束。

其后附有对主要代码进行解释的源代码段。

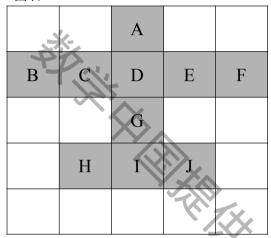
此法简单运用技巧性少,但对编程要求高。

打点函数关系法

本方法采用飞机上不同部位与四周的联系来推断第一次打击机体上的准确位置再通过 函数关系求取机头的坐标来打击目标。从上面的推断可得出在中央坐标(4,4)点机体所在 的概率最大,所以第一打击目标为中央点(4,4)。

此时可分为第一次打中或未打中两种情况。

一、第一次打中情况下,通过计算可得出打击点为 A、C、D、E、G、H、I、J点的概 图表 1



率都为 4/32, 而 B、C 为不可能的打击点, 所以 可以根据机身上每一点的坐标和机头的坐标来 建立函数关系,从而求出机头的坐标。

因为飞机的方向未定所以要建立飞机的四 个方向模型,来确定其相应机头的位置。此处采 用飞机上每一部位与其上、下、左、右的关系来 表示每一部位(左,上,右,下),用1、0来表 示在一方向上是否相连,即1表示相连,0表示 不相连。如左图 A (0, 0, 0, 1)。

建模:

设每一点的坐标为(X,Y);

A: $(0, 0, 0, 1) \longrightarrow (X, Y);$ $(0, 1, 0, 0) \longrightarrow (X, Y);$ B: $(0, 0, 1, 0) \longrightarrow (X+2, Y+1)$; $(0, 0, 1, 0) \longrightarrow (X+2, Y-1);$ C: $(1, 0, 1, 0) \longrightarrow (X+1, Y+1);$ $(1, 0, 1, 0) \longrightarrow (X+1, Y-1)$: D: $(1, 1, 1, 1) \longrightarrow (X, Y+1)$; $(1, 1, 1, 1) \longrightarrow (X, Y-1);$ E: $(1, 0, 1, 0) \longrightarrow (X-1, Y+1);$

 $(1, 0, 1, 0) \longrightarrow (X-1, Y-1);$

		A		
В	C	D	Е	F
		G		
	Н	I	J	
图表 2				

A:	4				
В:	. 1	Н	I	J	
C:			G		
D:	В	C	D	E	F
Е:			A	. /	

图表 3

F: $(1, 0, 0, 0) \longrightarrow (X-2, Y+1);$

F: $(1, 0, 0, 0) \longrightarrow (X-2, Y-1);$

G: $(0, 1, 0, 1) \longrightarrow (X, Y+2)$;

G: $(0, 1, 0, 1) \longrightarrow (X, Y-2);$

H: $(0, 0, 1, 0) \longrightarrow (X+1, Y+3);$

H: $(0, 0, 1, 0) \longrightarrow (X+1, Y-3)$;

I: $(1, 1, 1, 0) \longrightarrow (X, Y+3)$:

I: $(1, 0, 1, 1) \longrightarrow (X, Y-3)$:

J: $(1, 0, 0, 0) \longrightarrow (X-1, Y+3);$ J: $(1, 0, 0, 0) \longrightarrow (X-1, Y-3);$

例, B(0,0,1,0)——)(X+2,Y+1)表示B点的左、上、下不在飞机体上,右在飞机 上, \longrightarrow (X+2, Y+1) 表示从 B 点到机头 A 点所需的函数运算;

		В		
Н		С		
4	G	D	A	
J	7/7	E		
		F		

	В			
	С		Н	
A	D	G	I	
	Е		J	
	F			

图表 4

A: (1, 0, 0, 0)

 $-\rangle$ (X+1, Y-2): B: (0, 0, 0, 1)

C: $(0, 1, 0, 1) \longrightarrow (X+1, Y-1);$

D: $(1, 1, 1, 1) \longrightarrow (X+1, Y)$:

E: $(0, 1, 0, 1) \longrightarrow (X+1, Y+1)$;

F: $(0, 1, 0, 0) \longrightarrow (X+1, Y+2)$;

G: $(1, 0, 1, 0) \longrightarrow (X+2, Y);$

H: $(0, 0, 0, 1) \longrightarrow (X+3, Y-1)$:

I: $(0, 1, 1, 1) \longrightarrow (X+3, Y)$:

J: $(0, 1, 0, 0) \longrightarrow (X+3, Y+1);$

图表 5

A: $(0, 0, 1, 0) \longrightarrow (X, Y);$

B: $(0, 0, 0, 1) \longrightarrow (X-1, Y-2);$

C: $(0, 1, 0, 1) \longrightarrow (X-1, Y-1);$

D: $(1, 1, 1, 1) \longrightarrow (X-1, Y)$:

E: $(0, 1, 0, 1) \longrightarrow (X-1, Y+1);$

F: $(0, 1, 0, 1) \longrightarrow (X-1, Y+2);$

 $G: (1, 0, 1, 0) \longrightarrow (X-2, Y);$

H: $(0, 0, 0, 1) \longrightarrow (X-3, Y-1);$

I: $(1, 1, 0, 1) \longrightarrow (X-3, Y)$;

J: $(0, 1, 0, 0) \longrightarrow (X-3, Y+1)$;

此上为对飞机各个方向上的建模情况, 易知只要知道飞机上具体部位的坐标就可根据 一〉后的函数来求取机头 A 的坐标点;

此处采用对打击点上下左右与飞机体的联系来求取飞机的具体位置。即在打击到机体 一点是对其上下左右各个点进行打击来求出飞机的具体位置是 A、B、C、D、E、F、G、H、 I、J中的哪一个。

因为在击打到机头游戏结束,在 7*7 的方格中央点进行取点故无法取得 B、F 两格的数 据,当对上、下、左、右四格进行打击时机头在此四点的机体同时坠落,故在其下的讨论中 无需对 A, B, F, D 点进行讨论。

(1) 机头在 A 点打击次数和概率,此类点有 4 个:第一次打中即可

打击次数 N	1
P (A)	1/32

(2) 中央点(4,4)为D的打击次数和概率:共存在此类点M个,均匀分布在M点 的上、下、左、右,故:

打击次数 N	2	3	4	5
P (D)	1/4*1/32	1/4*1/32	1/4*1/32	1/4*1/32

(3)中央点(4,4)为 H, J的打击次数和概率:此两类点共有8个,在不同的两类 飞机中打击所得的数据相同,故需增一次打击,有:

打击次数 N	6	7
P (H, J)	1/2*1/32	1/2*1/32

(4) 对图中的 I 点,得出的数据可以唯一确定,存在此类点四个,故概率为:

打击次数 N	6
b (1)	1/32

(5) 在判断 C、E、G 三点时存在难度,因为存在六个相应的数据,故采用一定的技巧进行分解,对数据(0,1,0,1)即图一中的 G 点,图二中的 G 点,图四、图五中的 C、E 点,在竖直方向上分别加、减一次 2 即对(X, Y+2)、(X,Y-2),进行打击,记录下打中飞机的坐标(X, Y1),可辨别出 G 点;如果还是无法打击到机头,则打击点为 C 或 E 点,再打击点为(X+1,(Y1+4)/Z)和(X-1,(Y1+4)/Z)两点,每一点为机头的概率相同,存在此类点有 G 点有 A 个,C、E 点有 B 个故有:

打击次数 N	6	7	8	9
P (G)	1/2*1/32	1/2*1/32	0	0
P (C, E)	0	0	1/2*1/32	1/2*1/32

从而可求出在第一次打击中央打击中后打到机头的期望值为 E1(N)=4*P(A)*1+4*[2*P(D)+3*P(D)+4*P(D)+5*P(D)]+8*[6*P(H,J)+7*P(H,J)]+4*6*P(I)+4*[6*P(G)+7*P(G)]+8*[8*P(G)+9*P(G)]=47/8=5.875次;

注: 如果采用上、下、左、右、斜上左,斜下左,斜上右,斜下右,可一次确定飞机的准确部位,但会增加不必要的打击次数,又在中心点的上下左右四处机头的概率最大分布较均匀,且能较好地记录下打击的位置故选取上、下、左、右四点进行定点打击。用同样的方法对斜上左,斜下左,斜上右,斜下右四点进行定点打击最后的期望 E=15/2。

二、在第一次打未中的情况下,从 Matlab 的分析中可得出,此时飞机排列共有 16 种,

飞机机体各个部位所占的次数如下:

	1	3	2	3	1	
1		5		5		1
3	5	10	10	10	5	3
2		10		10		2
3	5	10	10	10	5	3
1		5		5		1
	1	3	2	3	1	

飞机机头所占的方格的次数如下

1/2/2/1/1 H H 3/3 1 H H 3/3/3/1 1										
				0/	6					
		1		1						
	1		2		1					
		2		2						
	1		2		1					
		1		1						
			L							

从以上的分布来看可见机头和机体都集中在中央编号(4,4)的周围,机头和机体在中央的上、下、左、右亦为最大概率点,故选取这四点为最佳打击点,再按照上面的打点法进行记录:

以下选取 A、B、C、D 四点为打击点,记录 16 种飞机在方格中排列点是否经过上四点,

		F		G		
	Е		В		Н	
4.		A		С		
	L		D		I	
		K		J		

1 代表经过,0 代表不经过,按 $A \times B \times C \times D$ 四点进行排列,如 F (1, 1, 0, 1),表示 F 为机头的飞机经过 $A \times B \times D$ 点,但不经过 C 点:

A	1	1	0	0
A	1	0	0	1
В	1	1	0	0
В	0	1	1	0
С	0	1	1	0
С	0	0	1	1
D	0	0	1	1
D	1	0	0	1
Е	1	1	1	0
F	1	1	0	1
G	0	1	1	1
Н	1	1	1	0
I	1	0	1	1
J	0	1	1	1
K	1	1	0	1
1/L	1	0	1	1

机头点 A B C D

从每次打击的结果数据记录可以确定机头的位置; 因为打击到机头,游戏结束,并且通过对各种数据的分析,可以减少一次对 A、B、C、D 四点的打击来确定机头的位置;

(1) 对 A、B、C、D 四点的分析,此四点中存在机头的次数为 8 个,故

打击次数 N	2	3	4	5
概率 P(A)	1/4*1/16	1/4*1/16	1/4*16	1/4*16

(2) 对 E, F, G, H, I, J, K, L 分析, 此 8 种排列方法相同, 需求的打击次数和概率相同, 先对 A、B、C 三点进行打击, 得出三组不同的数据, 每组数据包含两个点, 在对每个点分别进行打击, 每个概率都为 1/2, 故:

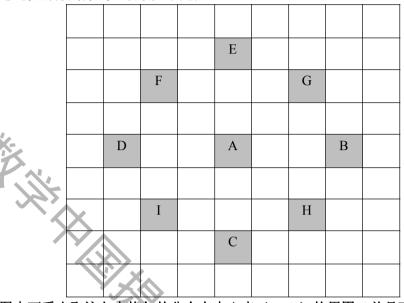
打击次数 N	5	6
概率 P(E)	1/2*1/16	1/2*16

此时的期望值 E2(N)=8*(2+3+4+5)P(A)+8*(5+6)*P(E)=9/2=4.5(次) 对一、二两种情况的共同期望值为: E(总数)=E1(N)*32/48+E2(N)*16/48=65/12=5.4167(次)

官方微博:http://weibo.com/304456943

报名号 #1963

三、当问题变为 9*9 的 81 个格子,排列两架飞机时,同样采样打点此法,但是因为此 时飞机的排列空间变大, 目多处一个飞机, 问题变得很复杂, 飞机排列的可能性也大大增强, 故在应用时要加大采样点的个数,取样点要尽可能大的缩小飞机的排列范围,用极可能少的 点来确定飞机的排列方式,故取如下九点:



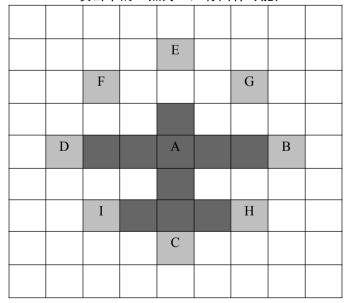
从图中可看出取这九点均匀的分布在中心点(5,5)的周围,并且可以很好地探测到 整体表格中飞机的分布情况,此方法首先考虑一个飞机在各点周围的分布情况,再将其组 合为两个飞机。又因为方格为正方形,故只考虑其中的一个方向,其余方向同理可得;

- 1、 当九点中没有一点被打中的情况时;
- 2、 当九点中只有一点打中, 分析其中的 A、B、G 三点, 其余点与这三点类似;
- 3、 当九点中有两点打中时, 分析 AB、AG、FG、GB, 这四种情况;
- 4、 当九点中有三点打中时,分析 ABG、GBH、AHG,这四种情况;
- 5、 当九点中有三点以上打中时, 其机头比在这九点中故不必将其进行讨论。

分别将各种类型称为 A、B、G、AB、AG、FG、GB、ABG、GBH、AHG 以及 0(无 10 PA 击中点),这11类情况;

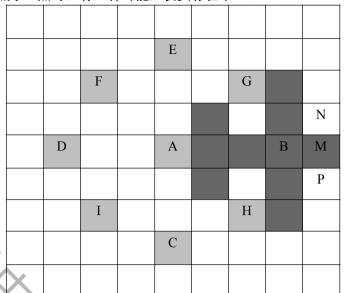
(1) 当只有一点被击中时,

1`设击中的一点为 A, 有四种可能:

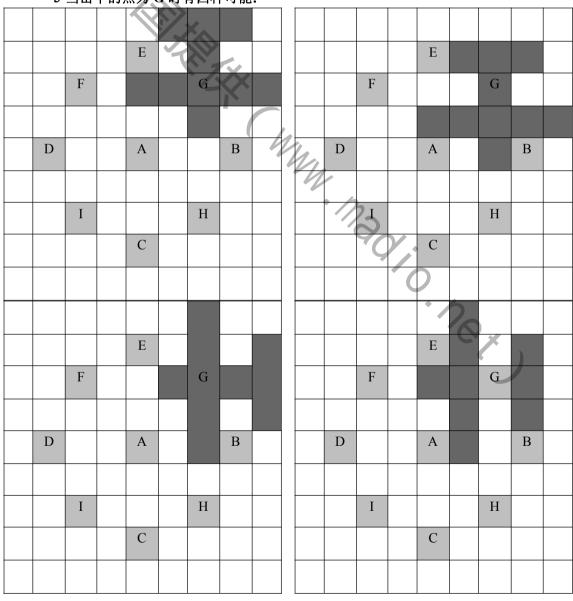


根据机头的不同方向存在四 种不同的排列方法:

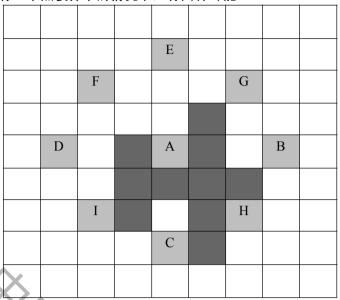
2`击中的点为 B 点时,有三种可能,机头存在于 M、N、P:



3`当击中的点为 G 时有四种可能:



(2) 当没有一个点被打中的情况下,有四种可能:



根据方向的不同还存在其余的三种情况;

(3)

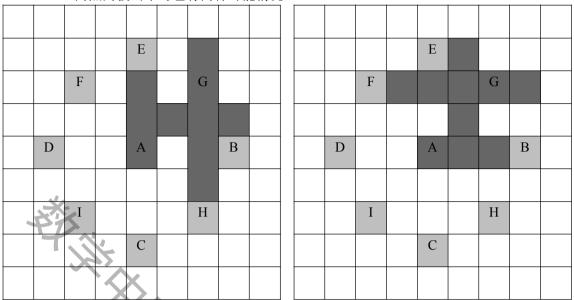
3) **当有两个点被击中时,** 1`仅 AB 之间不可能同时被击中,此时排列不出飞机模型;

2`仅 FG 之间也不可能同时被击中,此时也排列不出飞机模型;

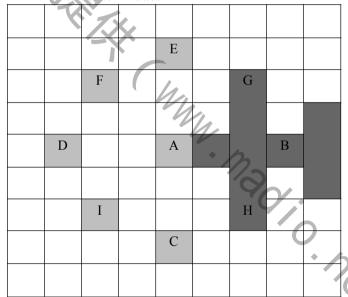
3'BG 两点之间被同时击中时,可分出两种情况,

						4	1	1/2								
			Е					1				Е				
		F		G						F				G		
									\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	7(/,•					
	D		A		В				D		(A			В	
												* /	20			
		I		Н						I			C	H		
			С									С				

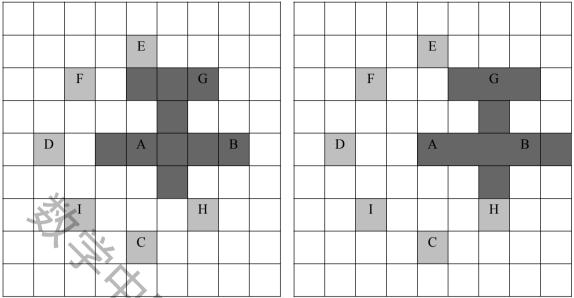
4'AG 两点间被击中时也有两种可能情况,



- (4) 当仅有三点被击中的情况下,
- 1`仅 ABG 三点不可能同时被击中,此时无法画出飞机模型;
- 2`仅为 GBH 三点击中时, 仅有一种情况:



3'ABG 三点被击中时,有两种不同情况:

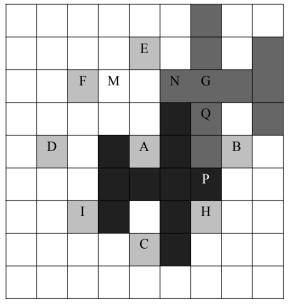


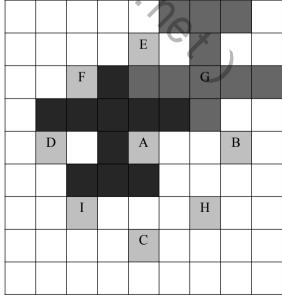
- (5) 同时存在四点被击中 AFGE 四点, 但机头为 A 点已被击落, 超过四点的可能性为 0; 接下来将一个飞机的可能方案组合成两种,因为一个飞机最多为三个被击中所以两架 飞机最多的可能为击中六个,从0开始进行组合,有:
 - a) 0: 0+0;
 - b) 1: 0+1;
 - c) 2: 0+2, 1+1;
 - d) 3: 0+3, 1+2;
 - e) 4: 1+3, 2+2;
 - f) 5: 2+3;
 - g) 6: 3+3°
 - i. 0: 0+0;

此种组合不可能存在, 故舍去;

1: 1+0: ii.

· Many · 进行到 无击中点和一个击中点进行组合,不能和 A、B 类进行组合,此时无法画出飞机 模型,选取 G类和 0类进行组合,有四种组合方式;





官方微博:http://weibo.com/304456943

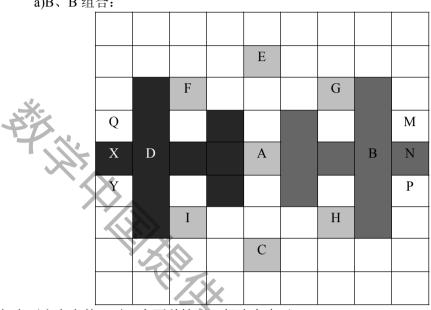
报名号 #1963

将每种飞机与其他一种进行组合,有四种排列方式;从而可以确定机头的位置, 只能在 M、N、P、Q 这四点中,要打下机头最多再加四次打击次数;

iii. 2: 0+2, 1+2;

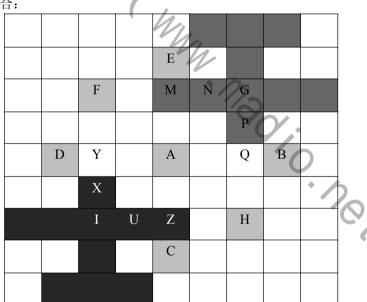
0与BG、AG类之间无法组合成飞机模型,选取B、B组合,G、G组合,B、G组 合, A、G 组合;

a)B、B组合:



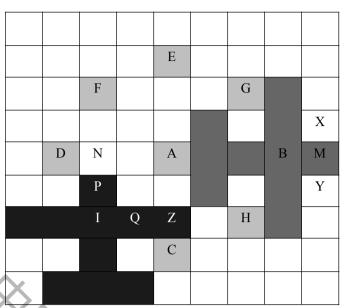
再加上下方向上的一对,有两种情况,机头存在于 M、N、P、Q、X、Y中;

b)G、G组合:



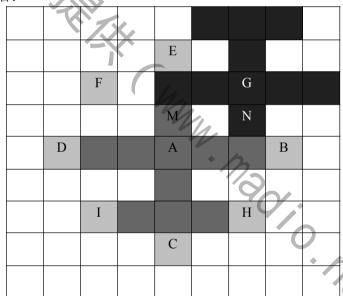
此种组合两两互不干涉,所以 I、G 点间不同的组合有 4*4=16 种,F、H 两点与 I、G 相同,有 16 种组合,机头排列在 M, N, P, Q, X, I, U, Z 这八点间;

c)B、G间的组合:



此种情况机头存在 M, X, Y和 P、N、Q、Z之间,同理 F和 B, E和 H, E和 I, D和 G, D和 H, C和 F, C和 G, 这七组和 G、I相同;

d) A、G组合:



仅存在一种情况,机头为 M 和 N,同理 F 和 A,I 和 A,H 和 A 的情况与 A 和 G 相同;

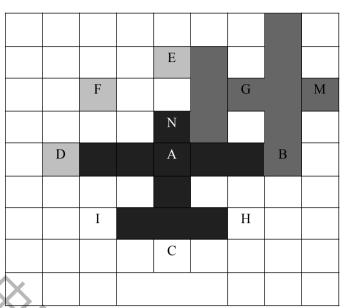
iv. 3: 0+3=1+2;

0 和三个点间无法有相应的模型组合;存在 A 和 BG 类,B 和 AG、BG 类,G 和 AG、BG 类的组合;

a) A和BG类组合:

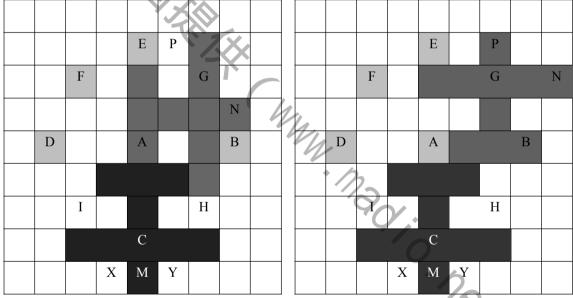
官方微博:http://weibo.com/304456943

报名号 #1963



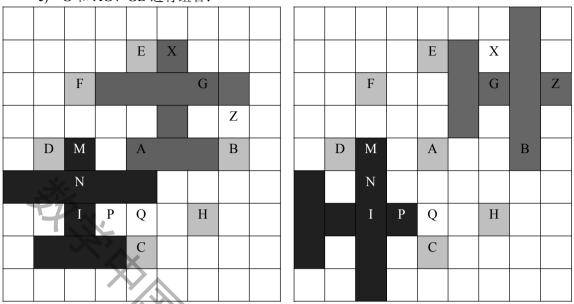
此类只有一种情况, 机头为 M 和 N, 同理当方向不同时, 得出的结果是相同的;

b) B和AG,GB进行组合:



此类 B 与 AG, GB 每类存在两种不同情况, 机头为 P, N 和 M、X、Y,

c) G和AG、GB进行组合:

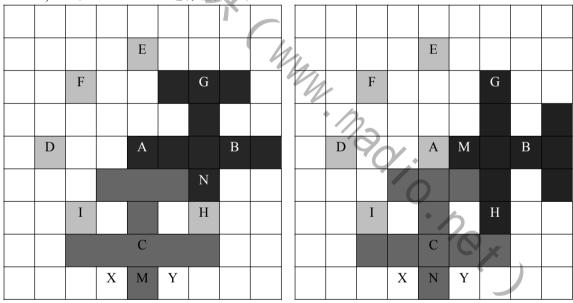


此类机头位于 M、N、P、Q 和 X、Z 之间的两点中;

4=1+3=2+2

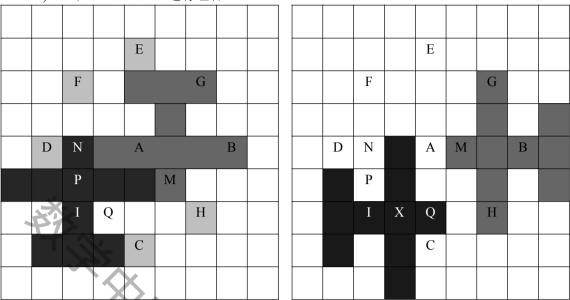
1+3 的组合存在于 B 和 ABG、GBH, G 和 ABG、GBH; 2+2 的组合存在 AG 和 BG

a) B和ABG、GBH进行组合



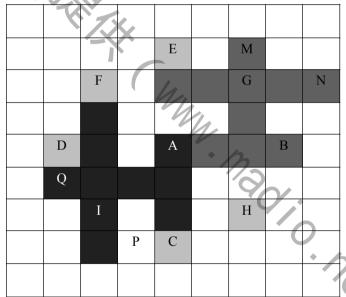
机头在 M、X、Y 和 N 两位置;

b) G和ABG、GBH进行组合



在 G 和 ABG 进行组合时,有一种不可排出机型故舍去,机头在 M 和 P、N、Q 中产生,在 G 和 GBH 进行组合时,机头在 M 和 N、P、X、Q 中产生;

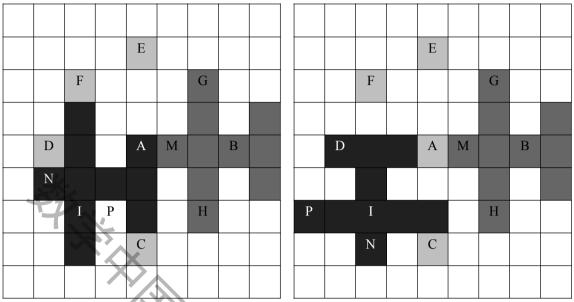
c) GB和AG进行组合



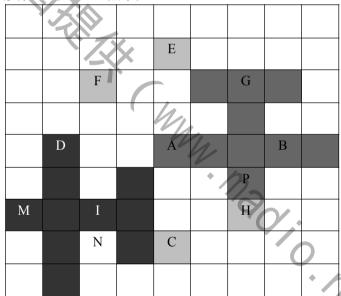
在此类组合中机头位于 Q、P和 M、N中;

vi. 5=2+3

2+3 存在于 BG 和 ABG, AG 和 GBH, BG 和 GBH 之间



此类情况下机头在 M 和 N、P 点中;



在此类情况下机头在 M、N 和 H 中;

vii. 6=3+3

3+3 存在于两个 GBH 中;

	2 2 11	hr 1 1.	, , 01	JII ' ';				
				Е				
		F				G		
	D		N	A	M		В	
	*							
7	P	1				Н		
		161	^	С				

此时机头在M和N中。

因为方格为 9*9 的正方形格, 故在不同方向上的打点结果都可以转换到上面所列的几种情况; 这样就完成了 9*9 方格中确定机头的打点函数法。

4、模型的比较和评价

第一类采用均分法,便于理解,理论上任何情况下的查找情况下寻找机头次数不会超过Log2[n],但过程繁杂计算量大,难于寻找到合适的二分点,但数据量增大时难干控制。

第二类采用了打点函数法,技巧性强,并非通过计算量来确定打击一定而是通过打点得到的数据进行分析,给每种情况建立合理的函数关系式,自然计算量相应的减少,但需要的技巧性强,难于把握全面,同时在取点过程中无法达到最优,只能根据数据分析和数据之间的关系来减少打击次数,在期望上要高于均分法,但适应于数据量大的事件,因为这种方法对于事件可能性概率的依赖小。

附录一、用 Matlab 软件给出均分法的算法过程的源代码:

i=[4 0 0 0 0 0 0]; //定义打击点的 x 坐标数组, 并赋值第一次打击 x=4 j=[4 0 0 0 0 0 0]; //定义打击点的 y 坐标数组, 并赋值第一次打击 y=4 ss=[0 0 0 0 0 0]; //定义用户输入的打击结果;

max=48; //初始飞机可能数

Chessmax=32: //过初始点(4, 4)的可能飞机数批:

for n=1:7

数学中国YY网校频道:159214

000000;00000]; //定义7*7 初始矩阵,记录每点可能飞机经过次数

Chessboard2=[0 0 0 0 0 0 0;0 0 0 0 0 0;0 0 0 0 0;0 0 0 0 0;0 0 0 0;0 0 0;0 0 0;0 0 0;0 0;0 0 0;0 0

```
000000;00000]; //定义7*7 初始矩阵,记录每点可能
  飞机头经过次数
fa=[0\ 0\ 0\ 0\ 0\ 0\ 0]:
                                                                                                                           //定义标志量
                      disp('attack ('): //向用户显示打击点坐标;
i (n)
j(n)
ss(n)=input('whether hit the plane(2 is hit the head of plane, 1 is hit, 0
is not hit)','s')-48
if ss(n)==2; //2 表示击中机头
                      disp('hit the head')
                                                                      //程序结束,跳出
                      break
else
for x=3:5
                                                                          //没击中机头时
                      for v=4:7
for cc=1:n
      if(ss(cc)==0) //没击中飞机
if((x=i(cc))\&(y==j(cc))) | ((x-2==i(cc))\&(y-1==j(cc))) | ((x-1==i(cc))\&(y===i(cc))) | ((x-1==i(cc))\&(y==i(cc))) | ((x-1==i(cc))\&(y==i(cc))) | ((x-1==i(cc))\&(y==i(cc))) | ((x-1==i(cc))\&(y==i(cc))) | ((x-1==i(cc))\&(y==i(cc))) | ((x-1==i(cc))\&(y==i(cc))) | ((x-1==i(cc))\&(y==i(cc)) | ((x-1=i(cc))\&(y==i(cc)) | ((x-1=
y-1==j(cc)) ((x==i(cc))&(y-1==j(cc))) | ((x+1==i(cc))&(y-1==j(cc))) | ((x+1==i(cc))&(y-1==j(cc)) | ((x+1==i(cc))&(y-1==j(cc))) | ((x+1==i(cc))&(y-1==j(cc)) | ((x+1==i(cc))&(y-1==i(cc)) | ((x+1==i(cc))&(y-1==i(cc)
x+2=i(cc)) & (y-1=i(cc))) \( (x=i(cc)) & (y-2=i(cc))) \| ((x-1=i(cc)) & (y-1=i(cc)) \| (x-1=i(cc)) & (y-1=i(cc)) & (y-1=i
3==j(cc)) | ((x==i(cc))\&(y-3==j(cc))) | ((x+1==i(cc))\&(y-3==j(cc))) //
  飞机经过该点时,使标志量为 1
                                                                                                                                                                                 hun nac
                            fa(cc)=1:
                            else
                                                              fa(cc)=0:
                             end
     end
                            if(ss(cc)==1)
if((x-2=i(cc))&(y-1==j(cc)))|((x-1==i(cc))&(y-1==j(cc)))|((x==i(cc)))|
(v-1=i(cc)) (x+1=i(cc)) (v-1=i(cc)) (v-1=i(cc))
((x=i(cc))\&(y-2=j(cc)))|((x-1=i(cc))\&(y-3=j(cc)))|((x=i(cc))\&(y-3=j(cc)))|
-3=i(cc)) ((x+1==i(cc))&(y-3==i(cc)))
                            fa(cc)=1:
else
                      fa(cc)=0:
end
end
fa(cc)=(fa(cc)&ss(cc))|(~fa(cc)&~ss(cc)); //击中时经过该点或者没击中
时不经过该点,则标志量为1
a=fa(cc)&a;//所有的打击点标识量都为 1,则 a=1;
end
if(a==1)
                                                                   Chessboard1(x, y) = Chessboard1(x, y)+1; //飞机经过该点 则
加1
```

```
Chessboard1 (x-2, y-1) = Chessboard1 (x-2, y-1)+1;
          Chessboard1 (x-1, y-1) = Chessboard1 (x-1, y-1)+1;
           Chessboard1 (x, y-1) = Chessboard1 (x, y-1)+1;
            Chessboard1 (x+1, y-1) = Chessboard1 (x+1, y-1)+1;
             Chessboard1 (x+2, y-1) = Chessboard1 (x+2, y-1)+1;
               Chessboard1 (x, y-2) = Chessboard1 (x, y-2)+1;
                Chessboard1 (x-1, y-3) = Chessboard1 (x-1, y-3)+1;
                 Chessboard1 (x, y-3) = Chessboard1 (x, y-3)+1;
                  Chessboard1 (x+1, v-3) = Chessboard1 (x+1, v-3)+1:
                  Chessboard2(x, y)= Chessboard2(x, y)+1;//飞机头经过该
点,则加1
end
    end
end
for x=3:5
    for v=1:4
          a=1:
for cc=1:n
    if(ss(cc)==0)
        if
(((x=i(cc))\&(y=i(cc))) | ((x+2=i(cc))\&(y+1=i(cc))) | ((x+1=i(cc))\&(y+1=i(cc))) |
+1=i(cc)) ((x=i(cc))&(y+1=i(cc))) ((x-1=i(cc))&(y+1=i(cc))) ((x-1=i(cc)))
-2=i(cc)) & (y+1=i(cc))) | ((x=i(cc)) & (y+2=i(cc))) | ((x+1=i(cc)) & (y+3)
==j(cc)) ((x==i(cc))&(y+3==j(cc))) (x-1==i(cc))&(y+3==j(cc))))
   fa(cc)=1:
        else
            fa(cc)=0;
        end
    end
     if(ss(cc)==1)
if(((x+2=i(cc))\&(y+1==j(cc)))|((x+1==i(cc))\&(y+1==j(cc)))|((x==i(cc))
(v+1=i(cc)) ((x-1=i(cc)) (v+1=i(cc)) (v+1=i(cc))
|((x=i(cc))\&(y+2=j(cc)))|((x+1=i(cc))\&(y+3=j(cc)))|((x=i(cc))\&(y+3=j(cc)))|
+3=i(cc)) | ((x-1=i(cc)) & (y+3=i(cc)))
   fa(cc)=1:
else
    fa(cc)=0;
end
fa(cc) = fa(cc) & ss(cc) | fa(cc) & ss(cc) :
a=fa(cc)&a;
end
if(a==1)
        Chessboard1 (x, y) = Chessboard1(x, y) + 1;
```

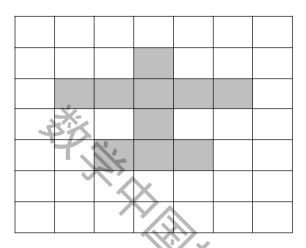
```
Chessboard1 (x+2, y+1) = Chessboard1 (x+2, y+1)+1;
                            Chessboard1 (x+1, y+1) = Chessboard1 (x+1, y+1)+1;
                               Chessboard1 (x, y+1) = Chessboard1 (x, y+1)+1;
                                 Chessboard1 (x-1, y+1) = Chessboard1 (x-1, y+1)+1:
                                    Chessboard1 (x-2, y+1) = Chessboard1 (x-2, y+1)+1;
                                       Chessboard1 (x, y+2) = Chessboard1 (x, y+2)+1;
                                          Chessboard1 (x+1, y+3) = Chessboard1 (x+1, y+3)+1;
                                             Chessboard1 (x, y+3) = Chessboard1 (x, y+3)+1;
                                                Chessboard1 (x-1, y+3) = Chessboard1 (x-1, y+3)+1:
                                                Chessboard2(x, y) = Chessboard2(x, y) +1;
end
end
end
  for x=1
           for
                       for cc=1:n
(((x=i(cc))\&(y=j(cc)))|((x+1=i(cc))\&(y-2=j(cc)))|((x+1=i(cc))\&(y-2=j(cc)))|
-1=i(cc)) ((x+1=i(cc))&(y==i(cc))) ((x+1==i(cc))&(y+1==i(cc))) ((x+1==i(cc)))
+1=i(cc) & (y+2=i(cc)) | ((x+2=i(cc)) & (y==i(cc)) | ((x+3=i(cc)) & (y-1)
==j(cc)) ((x+3==i(cc))&(y==j(cc))) ((x+3==i(cc))&(y+1==j(cc))))
                                                         fa(cc)=1:
                                                   else
                                                              fa(cc)=0;
                                                   end
                                     end
                                    if(ss(cc)==1)
                       if
 (((x+1==i(cc))\&(y-2==j(cc)))|((x+1==i(cc))\&(y-1==j(cc)))|((x+1==i(cc))\&(y-1==j(cc)))|
\&(y=j(cc))) | ((x+1=i(cc)) \& (y+1=j(cc))) | ((x+1=i(cc)) \& (y+2=j(cc))) |
((x+2=i(cc))&(y==j(cc))) | ((x+3==i(cc))&(y-1==j(cc))) | ((x+3==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1==i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(cc))&(y-1=i(
y==j(cc)) ((x+3==i(cc))&(y+1==j(cc))))
                      fa(cc)=1:
else
           fa(cc)=0;
                      end
                                     end
fa(cc) = fa(cc) \&ss(cc) | fa(cc) \& ss(cc) ;
a=fa(cc)\&a;
end
if(a==1)
           Chessboard1 (x, y) = Chessboard1(x, y) + 1;
```

```
Chessboard1 (x+1, y-2) = Chessboard1 (x+1, y-2)+1;
                           Chessboard1 (x+1, y-1) = Chessboard1 (x+1, y-1)+1;
                              Chessboard1 (x+1, y) = Chessboard1 (x+1, y)+1;
                                Chessboard1 (x+1, y+1) = Chessboard1 (x+1, y+1)+1:
                                   Chessboard1 (x+1, y+2) = Chessboard1 (x+1, y+2)+1;
                                      Chessboard1 (x+2, y) = Chessboard1 (x+2, y)+1;
                                         Chessboard1 (x+3, y-1) = Chessboard1 (x+3, y-1)+1;
                                            Chessboard1 (x+3, y) = Chessboard1 (x+3, y)+1;
                                               Chessboard1 (x+3, y+1) = Chessboard1 (x+3, y+1)+1:
                                               Chessboard2(x, y) = Chessboard2(x, y) +1;
end
     for x
           for v
                                    for cc=1:n
                                               if(ss(cc)==0)
if(((x=i(cc))\&(y=j(cc)))|((x-1=i(cc))\&(y+2=j(cc)))|((x-1=i(cc))\&(y+2=j(cc)))|
 (y+1==j(cc))) | ((x-1==i(cc)) & (y==j(cc))) | ((x-1==i(cc)) & (y-1==j(cc))) | (
 (x-1=i(cc))\&(y-2=i(cc))) ((x-2=i(cc))\&(y=i(cc))) ((x-3=i(cc))\&(y=i(cc)))
+1=i(cc)) ((x-3=i(cc))&(y=-j(cc))) ((x-3=i(cc))&(y-1=-j(cc))))
                                                          fa(cc)=1:
                                                    else
                                                               fa(cc)=0:
                                                    end
                                               end
                              if(ss(cc)==1)
if(((x-1==i(cc))\&(y+2==j(cc))) | ((x-1==i(cc))\&(y+1==j(cc))) | ((x-1==i(cc))\&(y+1==i(cc))) | ((x-1=i(cc))\&(y+1==i(cc))) | ((x-1=i(cc))\&(y+1==i(cc)) | ((x-1=i(cc))\&(y+1==i(cc)) | ((x-1=i(cc))\&(y+1=i(cc)) | ((x-1=i(cc))\&(y+1=i(cc)) | ((x-1=i(cc))\&(y+1=i(cc)) | ((x-1=i(cc))\&(y+1=i(cc)) | ((x-1=i(cc))@(y+1=i(cc)) | ((x-1=i(cc))@(y+1=i(cc)) | ((x-1=i(cc))@(y+1=i(cc)) | ((x-1=i(cc))@(y+1=i(cc)) | ((x-1=
c)) & (v=i(cc))) | ((x-1=i(cc)) & (v-1=i(cc))) | ((x-1=i(cc)) & (v-2=i(cc))
) |((x-2=i(cc))\&(y==j(cc)))|((x-3==i(cc))\&(y+1==j(cc)))|((x-3==i(cc)))|
\&(y==j(cc))) | ((x-3==i(cc)) \& (y-1==j(cc)))
   fa(cc)=1;
else
           fa(cc)=0:
                      end
                              end
fa(cc) = fa(cc) \&ss(cc) | fa(cc) \& ss(cc) ;
a=fa(cc)&a;
end
if(a==1)
                     Chessboard1 (x, y) = Chessboard1(x, y) + 1;
                        Chessboard1 (x-1, y+2) = Chessboard1 (x-1, y+2)+1;
                           Chessboard1 (x-1, y+1) = Chessboard1 (x-1, y+1)+1;
```

```
Chessboard1 (x-1, y) = Chessboard1 (x-1, y)+1;
            Chessboard1 (x-1, y-1) = Chessboard1 (x-1, y-1)+1;
              Chessboard1 (x-1, y-2) = Chessboard1 (x-1, y-2)+1;
               Chessboard1 (x-2, y) = Chessboard1 (x-2, y)+1:
                Chessboard1 (x-3, y+1) = Chessboard1 (x-3, y+1)+1;
                 Chessboard1 (x-3, y) = Chessboard1 (x-3, y)+1;
                  Chessboard1 (x-3, y-1) = Chessboard1 (x-3, y-1)+1;
                  Chessboard2(x, y) = Chessboard2(x, y) +1;
end
    end
  end
 if ss(n)=0
     max=max-Chessmax
 else
     \max = Chessboardl(i(n), j(n))
 end
 mid=Chessboard1(1,1
           x, y)~=0&abs(2*Chessbe,的打击点
mid=Chessboard1(x, y);
u=x;
~v;
 for x=1:7
     for y=1:7
         if
Chessboard2(x, y)^{\sim}=0&abs(2*Chessboard1(x, y)-max) <abs(2*mid-max); // 找
出最接近中点的打击点
         end
     end
 end
 i(n+1)=u
 j(n+1)=w
 Chessmax=Chessboard1(i(n+1), j(n+1))
   Chessboard1
    Chessboard2
end
end
```

附录二、我们这里也给出了另一种设计的方案,

问题二:考虑在 9×9 坐标纸上画两架飞机的游戏方式,两架飞机所占的格子不能重合,游戏方法同上。其中一架飞机被命中机头时要报告有一架飞机被击落。当某方的两架飞机都被击落时游戏结束,被击落方失败。分析这种游戏方式与只画一架飞机的游戏方式在策略上的不同点。



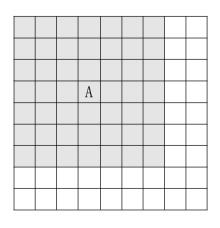
С	b	
a	d	

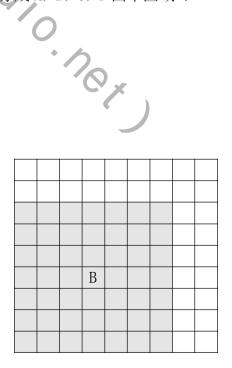
如图, 先从 7*7 的格子分析, 可以证明对于 7*7 的方格中, 中间 3*3 的格子中总是包含如上图的三个相连的图形 (方向不定,证明略), 也就是说如果推出图中存在如图的格子 a,

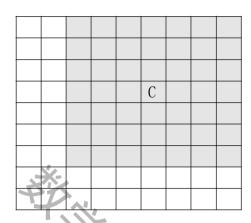
b,则 c 或者 d 又且只有一个存在。由此分析 7*7 的格子时可以对下图的点按照 概率大小进行打击。

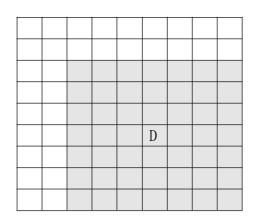
	2		4	
		1		
	5		3	

下面进行 9*9 的格子的两个飞机的 打击推测: 首先可以将 9*9 的格子进行 分区: 如图分成 A, B, C, D 四个区域:









可以证明,所有两个飞机的组合都可以分别划分在两个区域内,然后按照刚 开始建立的 7*7 的飞机模型,对 A,B,C,D 四个区域里每一个最中间的 3*3 格

					XA		
	6		10)	8	1	
		2		4		X	
	13		1		11	X	•
		5		3			
	7		12		9		•

数学中国YY网校频道:159214

子按照初始概率进行打击,即如图,如果其中不存在机头,则进行下面论证:如图观察,2,3,4,5地位等价;

6, 7, 8, 9地位等价;

10, 11, 12, 13 地位等价;

如果 2 点或者 6 点被打中,则证明一定存在一个飞机可以被划分到 A 区域(注意不是一定就属于 A 区域);

同理 5,7 对应 B 区域;

4, 9 对应 C 区域:

3,8 对应D区域。

而且 6, 7, 8, 9一定是只属于对应 A, B, C, D 区域的飞机的一部分。

10,11,12,13 中被击中,有两种可能一是证明这两家飞架飞机中有一个处于中间地带(可以属于两个区域);此时点(2,3,4,5)和(6,7,8,9)应成对称出现;二是该点只属于一个区域的飞机;

再通过相互间的组合关系来计算出每个飞机属于四个 7*7 区域中的哪一个, 然后就可以利用 7*7 格子的方法来进行运算。