

# 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](http://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：1170

参赛队员（签名）：

队员 1：魏翔

队员 2：刘飞

队员 3：王能波

参赛队教练员（签名）： 唐强

参赛队伍组别：大学组

# 2008 年 第一届“数学中国杯”

## 数学建模网络挑战赛

### 编 号 专 用 页

参赛队伍的参赛号码：（请各个参赛队提前填写好）：

1170

竞赛统一编号（由竞赛组委会送至评委团前编号）：

---

竞赛评阅编号（由竞赛评委团评阅前进行编号）：

# 2008 年 第一届“数学中国杯” 数学建模网络挑战赛

题 目                      飞机对战游戏策略设计

关 键 词                      二分法、摆放方式矩阵、覆盖矩阵

## 摘 要：

本题提出的是一个人机对战的游戏问题，要求给出计算机攻击的策略。

本文首先从飞机位置和方向的可能情形入手，采用 0—1 矩阵表示可能的飞机摆放方式，即飞机所占的格子对应的元素为 1，其余格子对应的元素为 0。然后通过 Matlab 编程生成了所有的飞机摆放方式矩阵，问题 1 有 48 个，问题 2 有 2352 个。同时选取特定的多种摆放方式，将它们对应的矩阵相加得到覆盖矩阵，得到各种摆放方式在某一格子上的覆盖次数。接着，本文创造性的采用了“二分法”的思想，提出了逐步判别、筛选满足条件的飞机摆放方式，最终锁定一架飞机的攻击策略。这一策略对于一架和两架飞机的情形都有很好的效果，是一个快速、有效的动态搜索方案。对于问题 1，我们给出了计算机程序判别路径的流程图及较详细的程序设计方案，程序员可以根据此方案编写出可行的人机对战游戏软件。对于问题 2，计算机的攻击策略的基本设计方案也是建立在双机摆放方式矩阵和“二分法”动态搜索的基础上的，文中给出了一条判别路径，证明了我们的方案是完全可行的。

根据“二分法”的原理，我们的策略对于问题 1 可以在至多攻击 7 次获胜，对于问题 2 至多攻击 14 次能够获胜，另外还设计了一些加速获胜的技巧渗入方案之中。

参赛队号    1170

所选题目    D 题

参赛密码                      \_\_\_\_\_  
(由组委会填写)

数学中国提供 (www.madio.net)

## 一、问题重述

在游戏前双方各准备一张坐标纸，在上面分别制作  $7 \times 7$  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。

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  $9 \times 9$  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。

## 二、模型假设

- 1) 飞机的各种摆放方式是等可能的；
- 2) 游戏一旦开始，飞机的位置是固定不动的；
- 3) 被攻击的一方没有虚报击中情况；
- 4) 不考虑对方对计算机的攻击的具体情况，这一方面计算机仅是生成飞机位置并当对方攻击时报告是否被击中，而本文主要考虑的是计算机如何快速有效地攻击对方。

## 三、符号说明

$$1) \quad A_k = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} \end{pmatrix} \quad \text{表示某架飞机的摆放方式的对应矩阵，其中}$$

$$a_{ij} = \begin{cases} 1, \text{该种摆放方式的飞机占据点 } (i, j), \text{ 如} \\ 0, \text{其它} \end{cases} \quad A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{表示机头坐标为}(2,4)\text{的方}$$

向向上的飞机的摆放方式。

$$2) \quad B_k = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} & b_{17} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} & b_{27} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} & b_{36} & b_{37} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} & b_{46} & b_{47} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} & b_{56} & b_{57} \\ b_{61} & b_{62} & b_{63} & b_{64} & b_{65} & b_{66} & b_{67} \\ b_{71} & b_{72} & b_{73} & b_{74} & b_{75} & b_{76} & b_{77} \end{pmatrix} \quad \text{表示多种可能的飞机摆放方式的覆盖矩阵，有}$$

$$\text{特定的多个 } A_k \text{ 相加而得 } b_{ij} = \begin{cases} 1, \text{多种摆放方式的覆盖点 } (i, j) \\ 0, \text{其它} \end{cases}$$

## 四、模型建立与计算机策略设计

### （一）问题 1 的建模与计算机策略设计

#### 1、问题 1 模型的建立和算法的设计

（1）为了解题的方便，将原试题中坐标轴顺时针旋转  $90^\circ$ ，如下图所示

|                |   |   |   |   |   |   |                   |
|----------------|---|---|---|---|---|---|-------------------|
| 1              | 2 | 3 | 4 | 5 | 6 | 7 | $\xrightarrow{Y}$ |
| 2              |   |   |   |   |   |   |                   |
| 3              |   |   |   |   |   |   |                   |
| 4              |   |   |   |   |   |   |                   |
| 5              |   |   |   |   |   |   |                   |
| 6              |   |   |   |   |   |   |                   |
| 7              |   |   |   |   |   |   |                   |
| $\downarrow X$ |   |   |   |   |   |   |                   |

（2）确定飞机摆放可能的方式

飞机摆放的方式取决于机头的位置和飞机的方向，如下图所示，粗实线框内的 12 个格子可以作为向上方向的机头，又如，粗虚线框内的 12 个格子可以作为向左方向的机头。另外，向右方向和向下方向的飞机数量分别为 12 架，因此，共有  $12 \times 4 = 48$  架飞机，可能成为机头的格子有  $7 \times 7 - 4 \times 4 = 33$  个。

|  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

（3）建立飞机摆放矩阵

|    |   |   |   |   |   |   |
|----|---|---|---|---|---|---|
|    | 0 | 0 | 0 | 0 | 0 | 0 |
|    | 0 | 0 | 0 | 1 | 0 | 0 |
|    | 0 | 1 | 1 | 1 | 1 | 0 |
| A= | 0 | 0 | 0 | 1 | 0 | 0 |
|    | 0 | 0 | 1 | 1 | 1 | 0 |
|    | 0 | 0 | 0 | 0 | 0 | 0 |
|    | 0 | 0 | 0 | 0 | 0 | 0 |

上图表示的是题目所画出的一架飞机。我们的表示方法就是在飞机所占的格子对应的元素为 1，其余格子对应的元素为 0。

为此，我们利用 Matlab 设计了一个小程序，生成了所有 48 架飞机所对应的摆放矩阵。

(4) 采用“二分法”逐步判定飞机位置和方向

1° 将 48 个飞机摆放矩阵相加，得到飞机覆盖矩阵

$$B_0 = \begin{bmatrix} 0 & 2 & 5 & 6 & 5 & 2 & 0 \\ 2 & 6 & 12 & 14 & 12 & 6 & 2 \\ 5 & 12 & 22 & 26 & 22 & 12 & 5 \\ 6 & 14 & 26 & 32 & 26 & 14 & 6 \\ 5 & 12 & 22 & 26 & 22 & 12 & 5 \\ 2 & 6 & 12 & 14 & 12 & 6 & 2 \\ 0 & 2 & 5 & 6 & 5 & 2 & 0 \end{bmatrix}$$

2° 按照二分法原理，从  $B_0$  中选择最接近  $48/2=24$  的元素，并确定其坐标  $(a_1, b_1) = (5, 3)$ （如有多个相同值，则从中任意选择一个），然后从 48 架飞机中挑选出覆盖该位置的飞机及其对应摆放矩阵，并把它们相加，这就得到了占据坐标  $(5, 3)$  的 22 架飞机的飞机覆盖矩阵（如下  $B_1$ ）；同时可以挑选出不覆盖该位置的飞机 26 架，也可以得到它们的覆盖矩阵（如下  $B_2$ ）。

|         |   |    |    |    |    |    |   |
|---------|---|----|----|----|----|----|---|
| $B_0 =$ | 0 | 2  | 5  | 6  | 5  | 2  | 0 |
|         | 2 | 6  | 12 | 14 | 12 | 6  | 2 |
|         | 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 |
|         | 0 | 2  | 5  | 6  | 5  | 2  | 0 |

覆盖

|         |   |   |    |    |    |   |   |
|---------|---|---|----|----|----|---|---|
| $B_1 =$ | 0 | 0 | 2  | 0  | 0  | 0 | 0 |
|         | 1 | 1 | 6  | 2  | 2  | 0 | 0 |
|         | 3 | 6 | 13 | 9  | 8  | 2 | 0 |
|         | 5 | 7 | 14 | 12 | 9  | 2 | 0 |
|         | 5 | 9 | 22 | 14 | 13 | 6 | 2 |
|         | 2 | 4 | 9  | 7  | 6  | 1 | 0 |
|         | 0 | 2 | 5  | 5  | 3  | 1 | 0 |

不覆盖

|         |   |   |    |    |    |    |   |
|---------|---|---|----|----|----|----|---|
| $B_2 =$ | 0 | 2 | 3  | 6  | 5  | 2  | 0 |
|         | 1 | 5 | 6  | 12 | 10 | 6  | 2 |
|         | 2 | 6 | 9  | 17 | 14 | 10 | 5 |
|         | 1 | 7 | 12 | 20 | 17 | 12 | 6 |
|         | 0 | 3 | 0  | 12 | 9  | 6  | 3 |
|         | 0 | 2 | 3  | 7  | 6  | 5  | 2 |
|         | 0 | 0 | 0  | 1  | 2  | 1  | 0 |

3° 炮击网格(5,3)，假设对方的回答是没有击中（如击中，算法类似），即对方的飞机没有覆盖(5,3)这个格子，则我们继续从  $B_2$  中选择覆盖值最接近  $26/2=13$  的元素，如  $B_2(4,3)=12$ 。

4° 炮击网格(4,3)，假设对方的回答是击中，即对方的飞机覆盖(4,3)这个格子，则类似于 3° 的做法，在没有覆盖网格(5,3)的基础上挑选出覆盖网格(4,3)的 12 架飞机，其覆盖矩阵为  $B_3$ 。

|        |   |   |    |    |    |    |   |
|--------|---|---|----|----|----|----|---|
| $B_2=$ | 0 | 2 | 3  | 6  | 5  | 2  | 0 |
|        | 1 | 5 | 6  | 12 | 10 | 6  | 2 |
|        | 2 | 6 | 9  | 17 | 14 | 10 | 5 |
|        | 1 | 7 | 12 | 20 | 17 | 12 | 6 |
|        | 0 | 3 | 0  | 12 | 9  | 6  | 3 |
|        | 0 | 2 | 3  | 7  | 6  | 5  | 2 |
|        | 0 | 0 | 0  | 1  | 2  | 1  | 0 |

|        |   |   |    |    |   |   |   |
|--------|---|---|----|----|---|---|---|
| $B_3=$ | 0 | 1 | 2  | 2  | 1 | 0 | 0 |
|        | 1 | 3 | 5  | 6  | 5 | 2 | 0 |
|        | 1 | 3 | 3  | 8  | 4 | 2 | 0 |
|        | 1 | 5 | 12 | 10 | 9 | 5 | 2 |
|        | 0 | 2 | 0  | 6  | 3 | 1 | 0 |
|        | 0 | 2 | 2  | 5  | 3 | 2 | 0 |
|        | 0 | 0 | 0  | 1  | 0 | 0 | 0 |

5° 继续从  $B_3$  中选择最接近  $12/2=6$  的元素，如  $B_3(2,4)=6$ ，炮击该网格，如果该回答被击中，则在原来的基础上挑选出覆盖网格(2,4)的 6 架飞机，其覆盖矩阵为  $B_4$ 。

|        |   |   |    |    |   |   |   |
|--------|---|---|----|----|---|---|---|
| $B_3=$ | 0 | 1 | 2  | 2  | 1 | 0 | 0 |
|        | 1 | 3 | 5  | 6  | 5 | 2 | 0 |
|        | 1 | 3 | 3  | 8  | 4 | 2 | 0 |
|        | 1 | 5 | 12 | 10 | 9 | 5 | 2 |
|        | 0 | 2 | 0  | 6  | 3 | 1 | 0 |
|        | 0 | 2 | 2  | 5  | 3 | 2 | 0 |
|        | 0 | 0 | 0  | 1  | 0 | 0 | 0 |

|        |   |   |   |   |   |   |   |
|--------|---|---|---|---|---|---|---|
| $B_4=$ | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
|        | 1 | 2 | 3 | 6 | 4 | 2 | 0 |
|        | 0 | 1 | 1 | 4 | 1 | 1 | 0 |
|        | 0 | 3 | 6 | 6 | 5 | 3 | 1 |
|        | 0 | 1 | 0 | 3 | 1 | 1 | 0 |
|        | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
|        | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

6° 继续从  $B_4$  中选择最接近  $6/2=3$  的元素，如  $B_4(4,2)=3$ ，炮击该网格，如果该网格回答被击中，则在前面的基础上挑选出又覆盖(4,2)格子的 3 架飞机，其覆盖矩阵为  $B_5$ 。

|        |   |   |   |   |   |   |   |
|--------|---|---|---|---|---|---|---|
| $B_4=$ | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
|        | 1 | 2 | 3 | 6 | 4 | 2 | 0 |
|        | 0 | 1 | 1 | 4 | 1 | 1 | 0 |
|        | 0 | 3 | 6 | 6 | 5 | 3 | 1 |
|        | 0 | 1 | 0 | 3 | 1 | 1 | 0 |
|        | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
|        | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|        |   |   |   |   |   |   |   |
|--------|---|---|---|---|---|---|---|
| $B_5=$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|        | 1 | 1 | 2 | 3 | 2 | 0 | 0 |
|        | 0 | 1 | 1 | 2 | 0 | 0 | 0 |
|        | 0 | 3 | 3 | 3 | 2 | 1 | 0 |
|        | 0 | 1 | 0 | 2 | 0 | 0 | 0 |
|        | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|        | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



7°继续从  $B_5$  中选择最接近  $3/2=1$  的元素,如  $B_5(2,1)=1$ ,炮击该网格,若该网格(2,1)回答被击中,则又在前面的基础上挑选出覆盖(2,1)的 1 架飞机,即覆盖矩阵为  $B_6$ 。

|        |   |   |   |   |   |   |   |
|--------|---|---|---|---|---|---|---|
| $B_5=$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|        | 1 | 1 | 2 | 3 | 2 | 0 | 0 |
|        | 0 | 1 | 1 | 2 | 0 | 0 | 0 |
|        | 0 | 3 | 3 | 3 | 2 | 1 | 0 |
|        | 0 | 1 | 0 | 2 | 0 | 0 | 0 |
|        | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|        | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|        |   |   |   |   |   |   |   |
|--------|---|---|---|---|---|---|---|
| $B_6=$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|        | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
|        | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|        | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
|        | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|        | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|        | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

这时,只剩下一架飞机,其机头坐标为(1,3),可以看出对方所画的飞机如下图:

|  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

以上只是假设计算机依次攻击  $(5,3) \rightarrow (4,3) \rightarrow (2,4) \rightarrow (4,2) \rightarrow (2,1) \rightarrow (1,3)$ ,而对方反馈依次为 击中  $\rightarrow$  未击中  $\rightarrow$  击中  $\rightarrow$  击中  $\rightarrow$  击中  $\rightarrow$  击中时的逐步判别最终锁定一架飞机的过程。共攻击次数 6 次。

(5) 采用“二分法”程序的效果分析

由于总飞机架次为  $48 < 2^6$ ,故在 6 次以内必定能够确定飞机的位置 and 方向,从而给出机头坐标,最后再增加一次攻击机头,因此在 7 次以内必定能够获胜,上述的那种情形恰为攻击 6 次获胜的例子。

在以下对于计算机如何给出全面的攻击策略的阐述中,我们将具体给出问题 1 的所有判别路径,即坐标选择的过程。

## 2、计算机进行游戏的策略

### (1) 建立数据库

给该计算机程序建立一个存储有 48 种飞机摆放方式的数据库,数据库中包括三部分:飞机序列号、飞机摆放矩阵、对应机头坐标

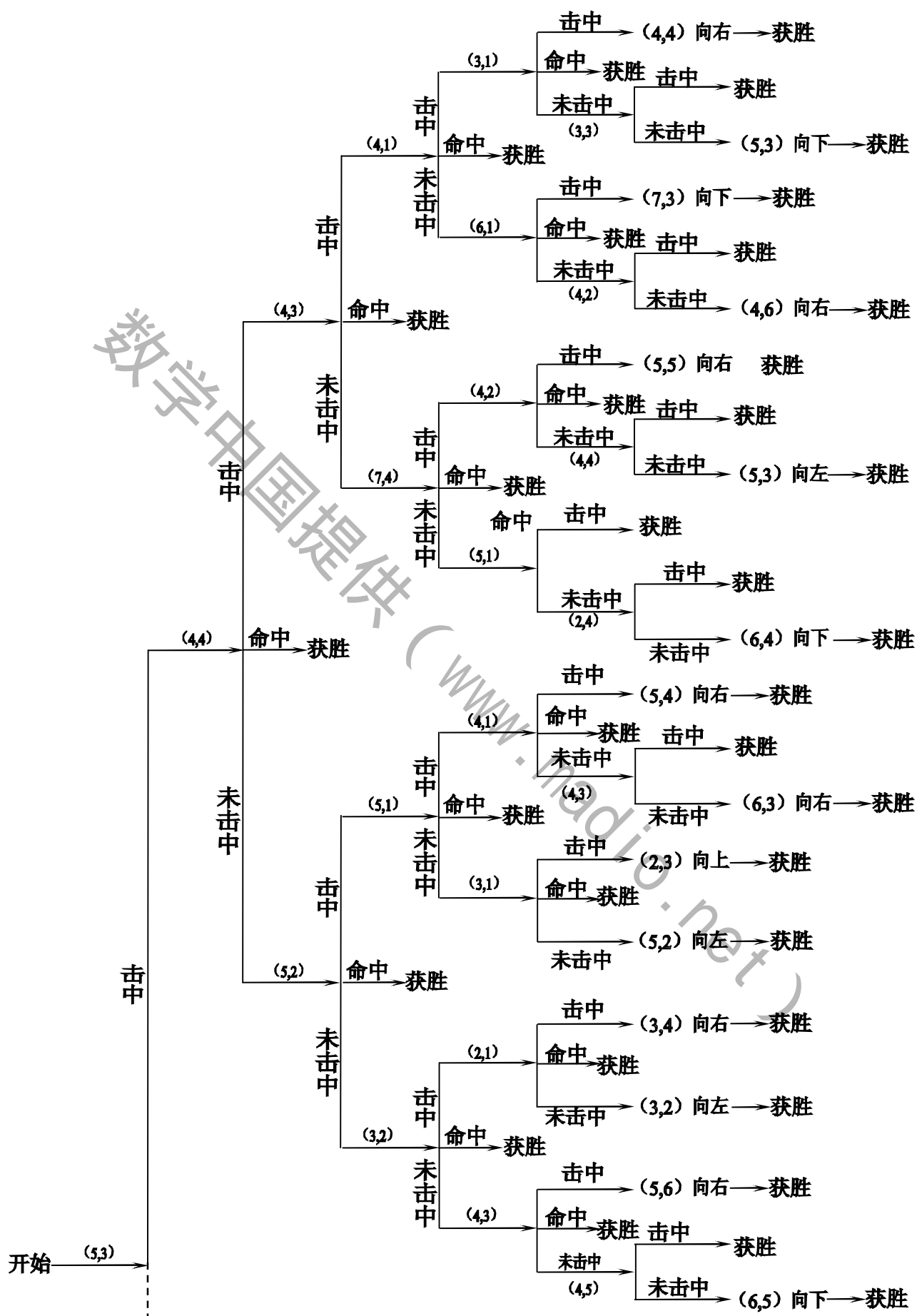
飞机序列号: 1~48

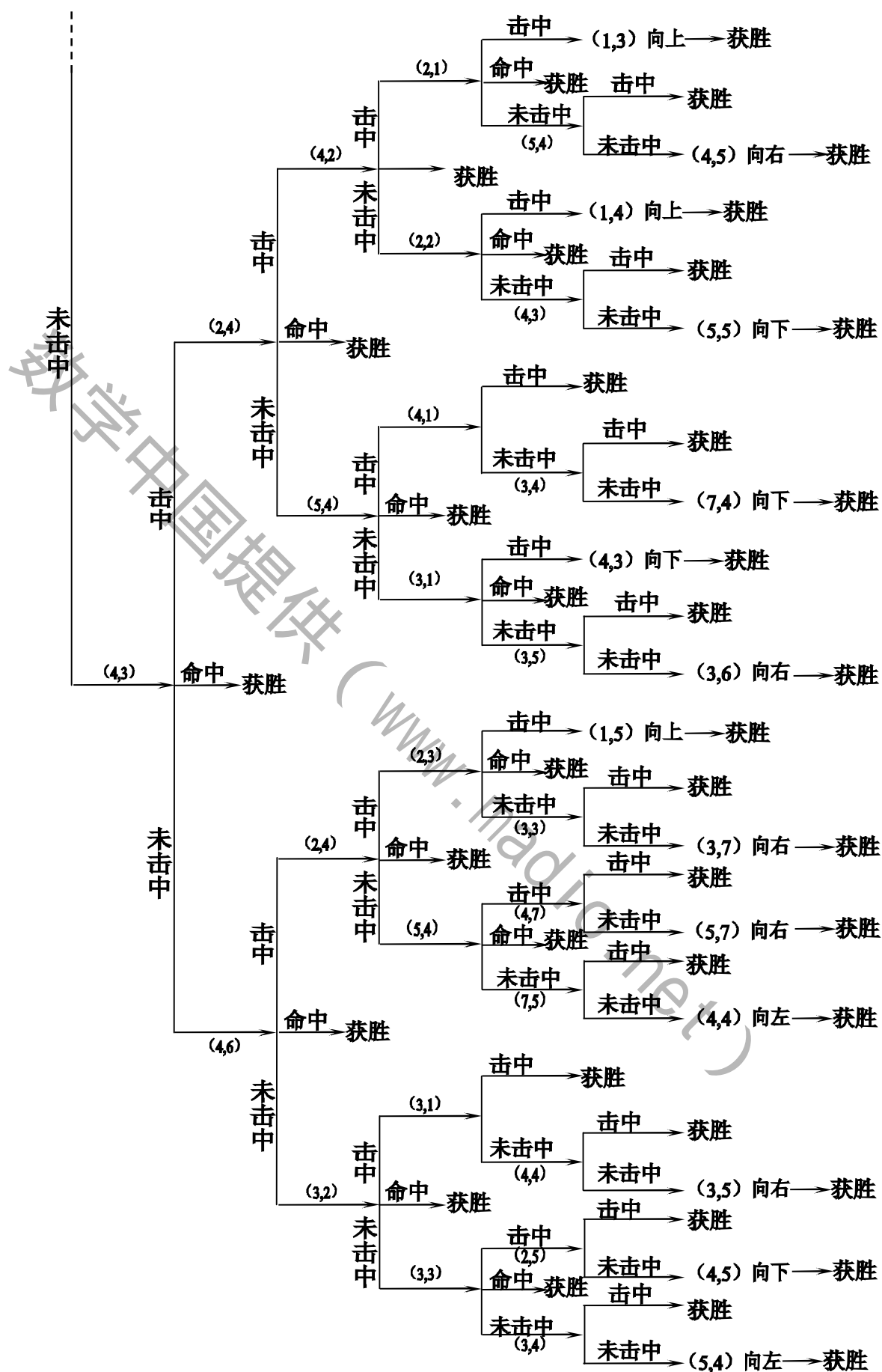
飞机摆放矩阵: 共 48 个,每个矩阵中飞机占据的格子对应数值为 1,其余对应为 0

对应机头坐标: 第  $k$  种飞机摆放方式的机头坐标  $(a_k, b_k)$ ,  $k = 1, 2, \dots, 48$

### (2) 计算机根据打击情况查找机头

设计程序中,计算机将按照既定坐标路径判别飞机摆放方式,从而确定机头坐标位置,其流程图见下页。





根据上述模型中的结论，计算机能够在 7 次之内命中机头，如流程图所示，第一次报告“炮弹”的打击位置是 (5,3)，若对方报告飞机被击中，则第二次打击位置是 (4,4)；若对方报告飞机未被击中，则第二次打击位置是 (4,3)。假设第一次对方报告飞机被击中，计算机将打击位置是 (4,4)，这时，若对方报告飞机被击中，则第三次打击位置是 (4,3)；若对方报告飞机未被击中，则第三次打击位置是 (5,2)。以下依次类推，把该流程图所示的各条路径设计为计算机程序，这样，“人机对战”中计算机将至多打击 7 次就可以命中机头。

### (3) 关于该程序方法的几点说明

1° 上面给出的判别路径不仅仅只有一种方案，其实可以组合产生多种方案，如在选择第一个坐标时就可以选择 (3,3) 或 (3,5) 或 (5,5)，它们的值都是 22，也可以选择另外四个值为 24 的格子，以它们开头都可以创建出多个方案。

2° 在攻击所选定的格子时，完全有可能直接命中上面提到的 33 个可能的机头位置，这样就可能提前获胜。

3° 为了尽快获胜，当接近判别路径的末端，仅有 2~3 架可能的飞机位置时，可以调出存储的机头坐标的数据，直接攻击其中某架飞机的机头位置。

## (二) 问题 2 的建模与计算机策略设计

### 1、问题 2 模型的建立

(1) 确定单架飞机可能的摆放方式，并建立摆放矩阵

同样利用前面设计的 Matlab 程序，可以获得  $9 \times 9$  格子中画一架飞机的所有可能情形，共有  $5 \times 6 \times 4 = 120$  种，可能作为机头的格子有  $9 \times 9 - 4 \times 4 = 65$ 。

(2) 确定两架飞机可能的摆放方式，并建立摆放矩阵

将生成的 120 种飞机位置进行两两比较，找出所有两两不重合的情形，将它们的摆放矩阵相加，得到双机的摆放矩阵，该矩阵同样是由 0 和 1 组成的，值为 1 的格子表示被两架飞机之一所覆盖，其余部分为 0。由程序经过  $C_{120}^2 = 7140$  次比对，发现有 2352 种可能的不重合的双机摆放方式。

(3) 依然采用“二分法”逐步缩小可能的飞机摆放方式数量，直至获胜。由于  $2352 < 2^{12}$ ，故知多攻击 12 次就可以确定两架飞机的摆放方式，最后攻击两个机头，至多  $12+2=14$  次就可获胜。

### 2、计算机进行游戏的策略

(1) 建立数据库

给该计算机程序建立一个存储有 2352 种双机摆放方式的数据库，数据库中包括三部分：飞机序列号、飞机摆放矩阵、对应机头坐标

飞机序列号：1~2352

飞机摆放矩阵：每个矩阵中飞机占据的格子对应数值为 1，其余对应为 0

对应机头坐标：每一种飞机摆放方式的对应的两个机头坐标

(2) 计算机策略举例

由于问题 2 考虑的是  $9 \times 9$  的双机矩阵，攻击次数可能达到 14 至多，因而无法像问题 1 那样给出完整的判别路径，由于篇幅有限，这里只就一条特殊的判别路径进行求解。

类似问题 1 中判别飞机摆放方式的过程，我们选定的这条判别路径是（共 11 个格子）：(4,4) → (3,3) → (4,3) → (4,1) → (6,6) → (8,6) → (6,7) → (6,4) → (6,2) → (8,5) → (8,4)，并且我们始终假定，这 11 次攻击后对方的报告都是被击中，以下是初始 2352 架飞机的覆盖矩阵  $B_0$  以及进行第  $k$  次（攻击选定的坐标并报告被击中）挑选后得到的覆盖矩阵

$B_k$ , 如下:

$B_0=$

|     |     |      |      |      |      |      |     |     |
|-----|-----|------|------|------|------|------|-----|-----|
| 0   | 131 | 299  | 308  | 280  | 308  | 299  | 131 | 0   |
| 131 | 332 | 594  | 647  | 654  | 647  | 594  | 332 | 131 |
| 299 | 594 | 956  | 1052 | 1054 | 1052 | 956  | 594 | 299 |
| 308 | 647 | 1052 | 1126 | 1056 | 1126 | 1052 | 647 | 308 |
| 280 | 654 | 1054 | 1056 | 960  | 1056 | 1054 | 654 | 280 |
| 308 | 647 | 1052 | 1126 | 1056 | 1126 | 1052 | 647 | 308 |
| 299 | 594 | 956  | 1052 | 1054 | 1052 | 956  | 594 | 299 |
| 131 | 332 | 594  | 647  | 654  | 647  | 594  | 332 | 131 |
| 0   | 131 | 299  | 308  | 280  | 308  | 299  | 131 | 0   |

$B_1=$

|     |     |     |      |     |     |     |     |     |
|-----|-----|-----|------|-----|-----|-----|-----|-----|
| 0   | 72  | 137 | 204  | 122 | 91  | 80  | 36  | 0   |
| 72  | 332 | 353 | 605  | 350 | 343 | 221 | 85  | 41  |
| 137 | 353 | 548 | 601  | 494 | 444 | 307 | 171 | 95  |
| 204 | 605 | 601 | 1126 | 593 | 580 | 430 | 244 | 107 |
| 122 | 350 | 494 | 593  | 444 | 513 | 434 | 253 | 117 |
| 91  | 343 | 444 | 580  | 513 | 522 | 474 | 273 | 140 |
| 80  | 221 | 307 | 430  | 434 | 474 | 438 | 273 | 139 |
| 36  | 85  | 171 | 44   | 253 | 273 | 273 | 148 | 59  |
| 0   | 41  | 95  | 107  | 117 | 140 | 139 | 59  | 0   |

$B_2=$

|     |     |     |     |     |     |     |     |    |
|-----|-----|-----|-----|-----|-----|-----|-----|----|
| 0   | 72  | 137 | 153 | 70  | 23  | 29  | 13  | 0  |
| 72  | 230 | 225 | 357 | 119 | 88  | 60  | 38  | 18 |
| 137 | 225 | 548 | 299 | 306 | 219 | 117 | 68  | 37 |
| 153 | 357 | 299 | 548 | 222 | 204 | 153 | 106 | 51 |
| 70  | 119 | 306 | 222 | 230 | 219 | 205 | 124 | 56 |
| 23  | 88  | 219 | 204 | 219 | 248 | 233 | 140 | 67 |
| 29  | 60  | 117 | 153 | 205 | 233 | 214 | 126 | 59 |
| 13  | 38  | 68  | 106 | 124 | 140 | 126 | 68  | 24 |
| 0   | 18  | 37  | 51  | 56  | 67  | 59  | 24  | 0  |

$B_3=$

|     |     |     |     |     |     |     |    |    |
|-----|-----|-----|-----|-----|-----|-----|----|----|
| 0   | 10  | 85  | 15  | 14  | 16  | 19  | 9  | 0  |
| 72  | 120 | 225 | 130 | 108 | 37  | 37  | 21 | 10 |
| 75  | 32  | 299 | 50  | 112 | 79  | 70  | 48 | 24 |
| 153 | 245 | 299 | 299 | 184 | 116 | 86  | 61 | 26 |
| 69  | 29  | 234 | 67  | 129 | 124 | 111 | 72 | 30 |
| 14  | 68  | 173 | 116 | 120 | 131 | 118 | 71 | 34 |
| 14  | 29  | 61  | 77  | 103 | 122 | 107 | 65 | 30 |
| 4   | 17  | 28  | 53  | 52  | 69  | 61  | 34 | 12 |
| 0   | 8   | 13  | 25  | 26  | 33  | 29  | 12 | 0  |

$B_4=$

|     |     |     |     |     |    |    |    |    |
|-----|-----|-----|-----|-----|----|----|----|----|
| 0   | 0   | 7   | 0   | 5   | 11 | 13 | 6  | 0  |
| 7   | 50  | 113 | 53  | 11  | 22 | 23 | 14 | 6  |
| 62  | 7   | 153 | 15  | 26  | 43 | 45 | 32 | 15 |
| 153 | 141 | 153 | 153 | 105 | 57 | 52 | 35 | 14 |
| 60  | 8   | 153 | 25  | 41  | 68 | 67 | 44 | 18 |
| 5   | 43  | 105 | 62  | 48  | 69 | 65 | 40 | 18 |
| 3   | 8   | 22  | 33  | 49  | 65 | 61 | 36 | 15 |
| 1   | 6   | 7   | 21  | 28  | 36 | 34 | 18 | 6  |
| 0   | 3   | 4   | 11  | 14  | 17 | 15 | 6  | 0  |

$B_5=$

|    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|
| 0  | 0  | 4  | 0  | 0  | 0  | 0  | 0  | 0  |
| 4  | 23 | 52 | 23 | 0  | 1  | 0  | 0  | 0  |
| 29 | 4  | 69 | 4  | 3  | 10 | 4  | 4  | 0  |
| 69 | 64 | 69 | 69 | 46 | 27 | 14 | 16 | 1  |
| 26 | 1  | 69 | 14 | 19 | 36 | 30 | 22 | 6  |
| 1  | 17 | 46 | 33 | 29 | 69 | 35 | 37 | 12 |
| 0  | 0  | 3  | 14 | 24 | 38 | 33 | 21 | 6  |
| 0  | 0  | 3  | 12 | 13 | 35 | 19 | 18 | 3  |
| 0  | 0  | 0  | 2  | 5  | 11 | 6  | 3  | 0  |

$B_6=$ 

|    |    |    |    |    |    |    |    |   |
|----|----|----|----|----|----|----|----|---|
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 |
| 0  | 14 | 27 | 14 | 0  | 0  | 0  | 0  | 0 |
| 13 | 0  | 35 | 0  | 0  | 0  | 0  | 0  | 0 |
| 35 | 35 | 35 | 35 | 22 | 5  | 0  | 0  | 0 |
| 13 | 0  | 35 | 4  | 4  | 15 | 6  | 6  | 0 |
| 0  | 8  | 22 | 19 | 16 | 35 | 17 | 17 | 3 |
| 0  | 0  | 2  | 12 | 16 | 26 | 20 | 15 | 3 |
| 0  | 0  | 2  | 10 | 11 | 35 | 13 | 15 | 3 |
| 0  | 0  | 0  | 2  | 5  | 11 | 6  | 3  | 0 |

 $\Rightarrow$  $B_7=$ 

|    |    |    |    |    |    |    |    |   |
|----|----|----|----|----|----|----|----|---|
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 |
| 0  | 7  | 13 | 7  | 0  | 0  | 0  | 0  | 0 |
| 6  | 0  | 17 | 0  | 0  | 0  | 0  | 0  | 0 |
| 17 | 17 | 17 | 17 | 11 | 5  | 0  | 0  | 0 |
| 6  | 0  | 17 | 2  | 1  | 7  | 3  | 3  | 0 |
| 0  | 4  | 11 | 9  | 14 | 17 | 17 | 11 | 3 |
| 0  | 0  | 0  | 2  | 1  | 10 | 6  | 3  | 0 |
| 0  | 0  | 0  | 4  | 9  | 17 | 11 | 9  | 3 |
| 0  | 0  | 0  | 0  | 0  | 3  | 3  | 0  | 0 |

 $B_8=$ 

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 3 | 5 | 3 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 9 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 9 | 9 | 9 | 7 | 3 | 0 | 0 | 0 |
| 2 | 0 | 9 | 2 | 1 | 5 | 1 | 1 | 0 |
| 0 | 4 | 7 | 9 | 8 | 9 | 9 | 5 | 1 |
| 0 | 0 | 0 | 2 | 1 | 6 | 2 | 1 | 0 |
| 0 | 0 | 0 | 2 | 5 | 9 | 5 | 3 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

 $\Rightarrow$  $B_9=$ 

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 4 | 4 | 4 | 4 | 1 | 0 | 0 | 0 |
| 0 | 0 | 4 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 4 | 4 | 4 | 3 | 4 | 4 | 3 | 1 |
| 0 | 0 | 0 | 0 | 0 | 2 | 2 | 1 | 0 |
| 0 | 0 | 0 | 1 | 2 | 4 | 3 | 3 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

 $B_{10}=$ 

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 |
| 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

 $\Rightarrow$  $B_{11}=$ 

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

可以看出  $B_{11}$  中两架飞机的机头坐标为(3,3)和(9,6)，这时还需攻击这两个格子以获胜。

### (3) 与问题 1 策略的不同之处

若命中其中的一架飞机，则立即从余下待判别的双机摆放方式中排除不包含该架飞机的摆放方式（设计程序搜索对应机头坐标的数据即可），从而可以更快获胜。

## 五、附录

程序过多，仅附部分，全部程序见电子档。

```
function [i,j]=f_local(A);
    f_num=sum(sum(A))/10;
    T=f_num/2-A;
    Q=abs(T);
    [x,y]=find(Q==min(min(Q)));
    i=x(1);
    j=y(1);
```

```
function T=fly11;
T=zeros(1,7);
for i=1:4
    for j=3:5
        A=zeros(7);
        A(i,j)=A(i,j)+1;
        A(i+1,j-2)=A(i+1,j-2)+1;
        A(i+1,j-1)=A(i+1,j-1)+1;
        A(i+1,j)=A(i+1,j)+1;
        A(i+1,j+1)=A(i+1,j+1)+1;
        A(i+1,j+2)=A(i+1,j+2)+1;
        A(i+2,j)=A(i+2,j)+1;
        A(i+3,j-1)=A(i+3,j-1)+1;
        A(i+3,j)=A(i+3,j)+1;
        A(i+3,j+1)=A(i+3,j+1)+1;
```

```
B=zeros(7);
k=8-j;
l=i;
B(k,l)=B(k,l)+1;
B(k,l+1)=B(k,l+1)+1;
B(k-1,l+1)=B(k-1,l+1)+1;
B(k-2,l+1)=B(k-2,l+1)+1;
B(k+1,l+1)=B(k+1,l+1)+1;
B(k+2,l+1)=B(k+2,l+1)+1;
B(k,l+2)=B(k,l+2)+1;
B(k,l+3)=B(k,l+3)+1;
B(k-1,l+3)=B(k-1,l+3)+1;
B(k+1,l+3)=B(k+1,l+3)+1;
```

```
C=zeros(7);
m=j;
n=8-i;
C(m,n)=C(m,n)+1;
C(m,n-1)=C(m,n-1)+1;
C(m+1,n-1)=C(m+1,n-1)+1;
C(m+2,n-1)=C(m+2,n-1)+1;
C(m-1,n-1)=C(m-1,n-1)+1;
C(m-2,n-1)=C(m-2,n-1)+1;
C(m,n-2)=C(m,n-2)+1;
C(m,n-3)=C(m,n-3)+1;
C(m-1,n-3)=C(m-1,n-3)+1;
C(m+1,n-3)=C(m+1,n-3)+1;
```

```

        D=zeros(7);
        p=8-i;
        q=8-j;
        D(p-3,q-1)=D(p-3,q-1)+1;
        D(p-3,q)=D(p-3,q)+1;
        D(p-3,q+1)=D(p-3,q+1)+1;
        D(p-2,q)=D(p-2,q)+1;
        D(p-1,q-2)=D(p-1,q-2)+1;
        D(p-1,q-1)=D(p-1,q-1)+1;
        D(p-1,q)=D(p-1,q)+1;
        D(p-1,q+1)=D(p-1,q+1)+1;
        D(p-1,q+2)=D(p-1,q+2)+1;
        D(p,q)=D(p,q)+1;

        T=[T;A;B;C;D];
    end
end
T(1,:)=[];

function HEAD=head(a)
k=0;
for j=1:7
    temp=find(a(:,j)==1);
    t(j)=length(temp);
    if length(temp)==4
        b=temp;
        c=j;
    end
end
end

if length(find(t~=4))==7
    k=1;
    a=a';
    for j=1:7
        temp=find(a(:,j)==1);
        t(j)=length(temp);
        if length(temp)==4
            b=temp;
            c=j;
        end
    end
end

if k==0
    if a(b(1,1),c+1)~=0
        HEAD=[b(4,1) c];
    else
        HEAD=[b(1,1) c];
    end
end

if k==1
    if a(b(1,1),c+1)~=0
        HEAD=[c b(4,1)];
    else
        HEAD=[c b(1,1)];
    end
end
end

```



```
function [TP,TN]=total1(x1,y1);
T=fly11;
TP=zeros(7);
TN=zeros(7);
for m=1:48
    if T(7*(m-1)+x1,y1)==1
        TP=TP+T(7*(m-1)+1:7*m,:);
    else
        TN=TN+T(7*(m-1)+1:7*m,:);
    end
end
Q=TP+TN;
```

数学中国提供 (www.madio.net)