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：

1198

参赛队员 (签名) ：

队员 1：肖智勇

队员 2：魏浩

队员 3：黄继达

参赛队教练员 (签名)：

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）：

1198

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　基于决策分析的飞机对战游戏的策略研究

关 键 词　对策论 随机模型 混合策略 Nash 均衡 动态规划

摘　　要：

本文在原有动态规划模型的基础上，引入矩阵进行分析，并将进攻阶段具体分为试探性进攻阶段与决定性进攻阶段。为了提高模型在实战中的应变能力，我们引入了关于试探阶段的混合策略，并在此基础上建立随机概率模型：

$$x = \begin{bmatrix} x_{17} & x_{27} & \cdots & x_{77} \\ x_{16} & x_{26} & \cdots & x_{76} \\ \vdots & \vdots & \ddots & \vdots \\ x_{11} & x_{21} & \cdots & x_{71} \end{bmatrix}$$

通过对该随机概率模型的实现，我们用 VB 编写出了一个小型游戏程序来进行实践模拟，结果发现该模型具有较高的可行性。

同时我们将问题二进行了矩阵转换，使得利用问题一的模型就可以来解决问题二，通过比较得了它们之间策略上的不同点。

我们用 N 人非合作对策的相关理论建立了问题三的模型：

$$\Gamma = (N, \{S_i\}, \{P_i\}), i = 1,2,3$$

并分别从防守和进攻两个方面为玩家提供了两种策略。

防守策略集：$SD_i = \left\{\text{相互距离小}, \text{相互距离大}\right\}, i = 1,2,3$

进攻策略集：$SA_i = \{\text{风险型策略}, \text{稳妥型策略}\}, i = 1,2,3$

同时对该模型我们也引入混合策略相关理论，得到混合策略模型：

$$X_i = \{xd_1^{(i)} \times xa_1^{(i)}, xd_1^{(i)} \times xa_2^{(i)}, xd_2^{(i)} \times xa_1^{(i)}, xd_2^{(i)} \times xa_2^{(i)}\} \qquad i = 1,2,3$$

其中有：

$$xd_1^{(i)} \times xa_1^{(i)} + xd_1^{(i)} \times xa_2^{(i)} + xd_2^{(i)} \times xa_1^{(i)} + xd_2^{(i)} \times xa_2^{(i)} = 1 \qquad i = 1,2,3$$

并通过数学规划对其进行求解，最终就可以为游戏者设计出一个最优游戏策略。

参赛队号　1198

所选题目　D 题

参赛密码 ＿＿＿＿＿＿

（由组委会填写）

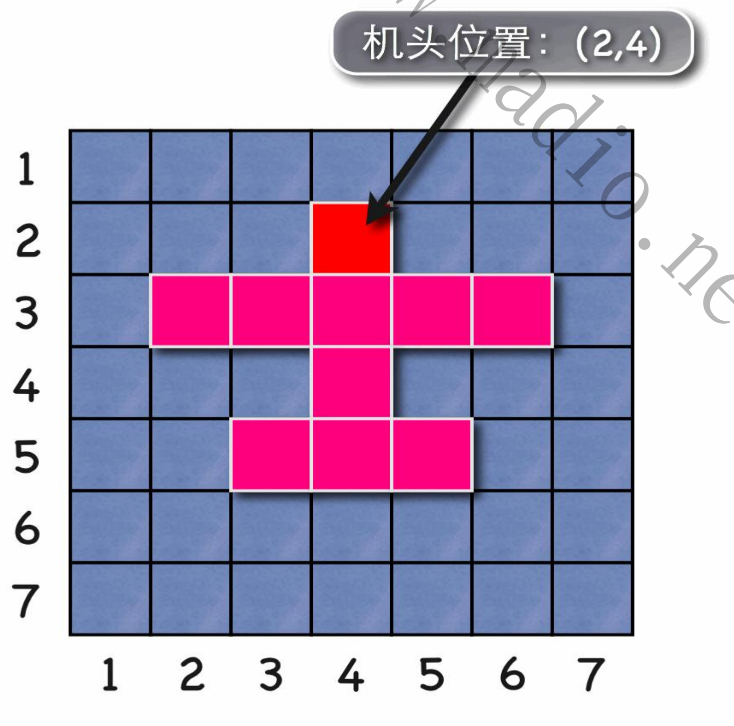
一、 问题的提出

1. 问题的背景

人工智能游戏充满了趣味性，而游戏人工智能领域目前正处于一个进化心理学、主流人工智能、游戏设计与实现多个领域的交点，并且游戏最终是一种商业，如何设计一个“飞机对战”游戏，使其符合一定的“作战规则”，使玩家能从游戏中得到乐趣，并在此基础上设计开发出一个三人乃至多个玩家的网络游戏，使之在一定的游戏规则和游戏方法下有较强的耐玩性，并具有一定的实用价值和存在潜在的商业价值，是一个值得研究的问题。

2. 问题的重述

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。



图表 1

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，

以计算机取胜为目标，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。
问题三：如果将问题二中的游戏方式设计为一个网络游戏，由三个真人对战，三人轮流作为攻击方，攻击方可以选择另外两方之一作为攻击对象，每次只能发射一发"炮弹"。如果某一方的两架飞机均被击中，他将退出这一局游戏，另外两方仍将继续游戏直到二者决出胜负。打中机头可以得 1 分，打中机头并把被攻击方踢出局可以得 3 分，打中飞机其它部位或者未击中不得分，比赛采用 18 局，最后总得分（累加每局得到的分数）最高的一方获胜，如果出现平分，加赛一局决定胜负。考虑每局都首先由你作为攻击方，设计一套游戏策略，使你能在比赛中取胜。（假定游戏中，不存在任意两方联合的可能）

二、 基本假设

1. 在相互攻击时，双方或三方之间没有任何倾向性的暗示或提示。
2. 在相互攻击的过程中，双方或三方均以取胜为目标，采取最优策略进行攻击，即在某局失败之前，均不放弃分析对方的可能位置与自己的攻击策略。
3. 考虑到以计算机取胜为目标，则问题一和问题二仅给出计算机的攻击策略，不考虑计算机被攻击的情况。
4. 在对抗的过程中，游戏玩家不通过一定方式了解计算机其他游戏玩家的所有分析情况与分析的攻击策略。
5. 假设在三人对战中，每个人都采取最优策略进攻。

三、 名词约定及符号的说明

1. 名词约定

AI: 就是人工智能(artificial intelligence)的缩写,编程技术为处理和显示符号资料而非数字资料,解释"怎样揭示或推导一条特殊信息的技能"，处理不完全或不确定数据的技能，处理新情况的技能。[1]

2. 符号说明

i：飞机所在位置中的某格的横坐标。

j：飞机所在位置中的某格的纵坐标。

m：表示机头的位置，即$10i+10j$，取值为$11\sim 77$。

n：表示机头的方向，即每一架飞机的方向右机头的位置所确定，取值为 1，2，3，4；其中 1 表示机头朝上，2 表示向右，3 表示向下，4 表示向右。

$CV(i,j)$（Cube Value）：飞机的机头或某一格机身在第i行，第j列的可能性。

$w(i,j,m,n)$：飞机上的某一格位置的权重值，取值为 0，1，9，其中 0 表示该位置没有飞机，1 表示某格有一格机身，9 为机头。

$head(i,j)$：表示飞机头部的一个数组，取值为-1，0 当 $head(i,j)=-1$ 时，规定 (i,j) 位置正是实际机头的位置。而当 $head(i,j)=0$ 时，规定 (i,j) 位置不是实际机头的位置

$direct$：取值为 1，2，3，4。当 $direct=1$ 时，规定方向向上；当 $direct=2$ 规定方向向右；当 $direct=3$ 规定方向下；当 $direct=4$ 规定方向向左。

Ef 值：在 9×9 的方格里的两架飞机之间的相互影响度。

四、 问题的分析与模型的建立

对问题一的分析

前一篇论文中一的模型：

关于 1 架飞机模型，我们可以将该模型分为 3 个阶段：

(一)　求出最佳攻击位置

(二)　进攻并回馈信息

(三)　分析信息

第一阶段：最佳攻击位置的求解

$$\begin{cases} D_1=\max\{CV(i,j)\} \quad \text{其中 } isFact(i,j)=-1 \\ w(i,j,m,n)=\begin{cases} 9 & \text{规定由}(m,n)\text{所确定的飞机的机头就在}(i,j)\text{位置，} \\ 1 & \text{规定由}(m,n)\text{所确定的飞机的机身在}(i,j)\text{位置。} \\ 0 & \text{规定由}(m,n)\text{所确定的飞机在}(i,j)\text{位置没有权值。} \end{cases} \\ CV(i,j)=\sum_{m=11}^{77}\sum_{n=1}^{4}w(i,j,m,n) \\ ph=9 \\ pb=1 \\ none=0 \\ isFact(i,j)=\begin{cases} ph & \text{说明}(i,j)\text{位置已经被攻击过，而且攻到的是机头。} \\ pb & \text{说明}(i,j)\text{位置已经被攻击过，而且攻到的是机身。} \\ none & \text{说明}(i,j)\text{位置已经被攻击过，但是没有攻到敌方。} \\ -1 & \text{说明}(i,j)\text{位置还没有被攻击过。} \end{cases} \end{cases}$$

第二阶段：攻击并回馈信息

$$\begin{cases} head(i,j)=\begin{cases} -1 & \text{规定}(i,j)\text{位置正是实际机头的位置。} \\ 0 & \text{规定}(i,j)\text{位置不是实际机头的位置。} \end{cases} \\ \sum_{i=1}^{XX}\sum_{j=1}^{YY}head(i,j)=-1 \\ XX=YY=7 \\ isFact(i,j)=\begin{cases} ph & \text{当 } head(i,j)=-1 \text{ 时} \\ pb & \text{当}(i,j)\text{位置为机身时} \\ none & \text{当}(i,j)\text{位置没有飞机时} \end{cases} \end{cases}$$

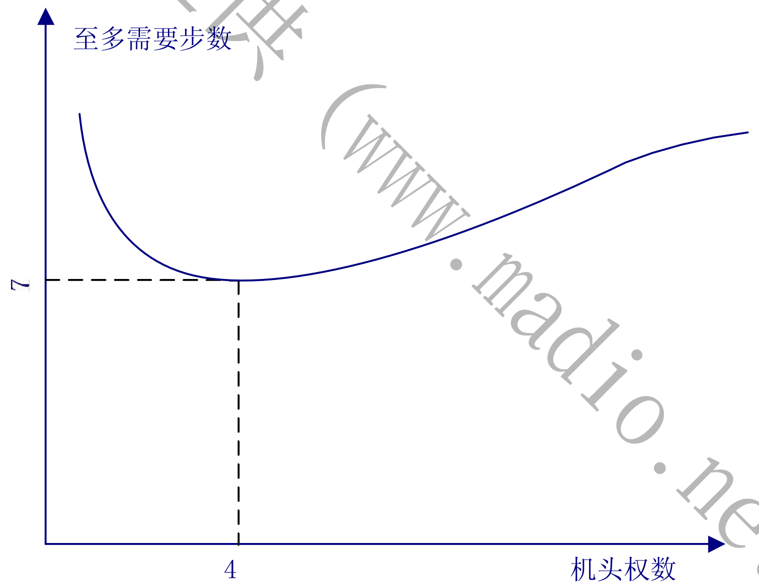
第三阶段：信息的分析

$$\begin{cases} \text{游戏结束} & isFact(i,j)=ph\text{时，} \\ w(i,j,m,n)=0\ \begin{cases}\text{飞机}(m,n)\text{不包括}(i,j)\text{位置} \\ \text{飞机}(m,n)\text{包括}(i,j)\text{位置，}\end{cases} & \begin{matrix} isFact(i,j)=pb\text{时，} \\ isFact(i,j)=none\text{时，}\end{matrix} \end{cases}$$

机头权数分析：

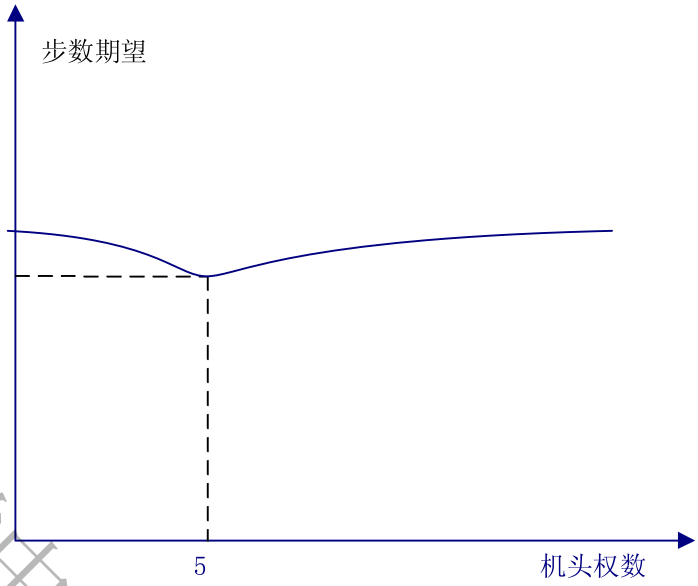
由于我们进攻的目标是对方的机头，机头被击中就意味着游戏结束，所以在设置权数时我们将机头的权数设的要比其他位置大一些，首先，我们将机身的权数就 1，起初，我们将其设为 9，因为击中机身除了机头以外的其他 9 个位置，飞机的形状和位置也就确定了，就相当于知道了机头的位置，可是，我们猜想，在实践中， 9 可能不是一个最理想的权数，也许机头的权数有某一个临界值，当权数超过该临界值后，如果再将其增大，对于下一步的进攻策略的并不会带来太大影响，而如果小于这个临界值，会增加进攻的模糊度，也就是说，降低了击中机头飞可能性。

通过编程测试，我们得到一些结果：如，我们将机头的机权数从 2 开始一直到 18 进行了多次测试和数据的分析，得到当机头权数设为 4 时至多只需要 7 步计算机就可以击中对方的机头，而当机头权数小于这或大于这个权数 4 时，我们发现就可能需要 8 次或更多的次数。这个结果如图表 2 所示：



图表 2

但是，我们却发现机头权数为 4 时，即使至多只需 7 次就可以将对方击败，可那只是最坏的打算，而如果我们以期望去计算能攻到敌方的步数时就会发现其实 5 才最好的选择，我们以期望值来重画图表 2 就会发现，当以期望值为纵坐标时，5 才是临界点，如图 3 所示：



图表 3

因此，根据以上分析，我们选取机头权数为 5。

在实践中，也正是如此，当机头权数 5 时，计算机获胜的概率得到了大大的提高。

策略分析：

问题一是要我们给出一个策略来使计算机战胜人（玩家），所以以下我们用对策论的知识，并结合了动态规划和随机模型的一些理论来分析该问题。

我们将计算机和玩家分别看成两个局中人 player（1）和 player（2），下面我们就计算机以怎样的策略才能在十九局十胜制的游戏中取胜做出具体的分析。

首先，我们可以将计算机可行的进攻策略归纳到一个策略集 S（Strategy）中，在采用其中的某一个策略时表格中每一点的权重值 $CV(i, j)$ 称为该策略的状态支付函数 Pay。

对于计算机的进攻我们可以有三种策略集的划分：

① 计算机的每一步进攻都可以看作是一个策略，对于在 7×7 的表格中计算机就有 49 种策略，这 49 种策略就可以构成一个策略集，而当表格中有一些位置进行了进攻以后，策略集中的元素就减少了。如当只有 35 个位置没有被进攻过时，这时的策略集就只有 35 种策略了。这种思路中的策略集是进行着变动的，是一种动态策略集的思路。

② 整体进攻策略，即在每一回合中从游戏开始到该回合结束时，计算机进攻的每一步组成一个整体的策略，这一个整体的策略就构成了计算机的一个进攻方案，计算机可以通过运算，得到许多的进攻方案。这样这些进攻方案就可以组成一个整体的进攻策略集了，一旦开始进攻，计算机就可以从这些方案中取出一套方案来进行进攻。

③ 混合进攻策略。我们将进攻分为两个阶段。第一个阶段：试探性的进攻或者是模糊进攻，即在表格中由计算机任意去进攻可能有机头的位置，该阶段一方面是试探对方飞机的大体方位，为下一阶段的决定性进攻做铺垫，另一方面是为增加计算机进攻策略的随机性，减小玩家了解计算机进攻思路的可能性。

第二阶段：决定性的进攻，即按照我们事先为其设计的思路去进攻。基于第一阶段的试探，在该阶段计算机进攻的成功率会得到大大的提高。

于是这个策略集就可以写成：

$$S=\{\text{试探性进攻},\text{决定性进攻}\}$$

对前一篇论文所建的模型的总结：

通常我们应该去进攻是机头可能性最大的位置，上一篇论文中的模型就是基于此思想建立的，可是此模型有一定的局限性，随机性太小，经过若干次的对战后玩家很容易摸清计算机的思路和弱点，这样在十九局十胜制的游戏中，大大降低了计算机获胜的概率，因此，基于该模型，我们加入各种随机因素，来增加计算机策略的不可预测性，并建立改进模型。

通过对上一篇论文中的模型的实践我们发现在刚开始的几次进攻中，计算机总是去进攻靠近表格中心的位置，玩家在经过几次对战后，又很容易发现计算机的该弱点，这样玩家在摆放飞机的时候就会有意识地去回避把机头放在中心区域，从而为自己赢得一定的先机。因此，我们考虑去建立混合策略模型。在刚开始的几次进攻中，让计算机去试探性的进攻，而不是按照该模型“死板”的策略去进攻。所谓试探性的进攻，就是我们事先为计算机选定一些对方可能摆放飞机头的位置，让计算机随机的去进攻，这样不仅增加了计算机进攻策略的不可预测性，而且，为下一阶段的进攻提供的更多的反馈信息。

改进模型的具体分析：

改进模型一：

利用上一篇的论文，我们将上面策略集中的第一种思路与第二种思路加入该模型中，使得该模型能得到进一步的完善。

首先对混合进攻策略进行分析，在游戏开始时是利用试探性进攻的策略来进行进攻，设试探性进攻的次数设为 $probe$。

那么我们究竟该进行几次试探性的进攻呢？

显然，如果试探的次数 $probe$ 过大，计算机的风险也就增大了，毕竟，我们进行试探一方面是增加计算机策略的随机性，另一方面就是获取反馈信息，而这种随机性的进攻就像玩家刚开始进攻一样，是根据自己的感觉去选择进攻位置，并没有太多理论上的依据，所以，这种试探性的进攻不能太多。但是，如果试探的次数少了，计算机策略的随机性又会降低，而且得到的反馈信息也太少，并不利于第二阶段的决定性进攻。我们通过 VB 编程初步实现了人机对战，并利用该程序调试出试探进攻的次数为 2 次比较合适。也就是说，在计算机进行第二阶段决定性进攻的第一次进攻已经有了试探进攻是的两个反馈信息，很明显，在这种情况下，计算机进攻的成功率有了很大的提高！

因为该试探性进攻具有随机性，故我们引入混合策略的相关知识来建立一个随机策略阶段模型。

设混合策略模型：

$$\mathrm{x}=\begin{bmatrix} x_{17} & x_{27} & \cdots & x_{77} \\ x_{16} & x_{26} & \cdots & x_{76} \\ \vdots & \vdots & \ddots & \vdots \\ x_{11} & x_{21} & \cdots & x_{71} \end{bmatrix}$$

其中，x_{ij} 表示在该混合策略中，向 (i,j) 位置进行进攻的概率。

这就是我们利用混合策略知识建立的一个随机概率模型。而对于概率矩阵

x 的求法，我们利用初始状态的数据来求出它。

对初始状态，求出初始：

$$CV(i,j)=\sum_{m=1}^{77}\sum_{n=1}^{4} w(i,j,m,n)\text{（参考我们的前一篇论文）} \quad (1)$$

利用公式（1），就可以求出如下的初始决策矩阵：

$$CV(i,j)=\begin{bmatrix} 0 & 2 & 9 & 10 & 9 & 2 & 0 \\ 2 & 6 & 16 & 18 & 16 & 6 & 2 \\ 9 & 16 & 30 & 38 & 30 & 16 & 9 \\ 10 & 18 & 38 & 48 & 38 & 18 & 10 \\ 9 & 16 & 30 & 38 & 30 & 16 & 9 \\ 2 & 6 & 16 & 18 & 16 & 6 & 2 \\ 0 & 2 & 9 & 10 & 9 & 2 & 0 \end{bmatrix}$$

并且我们利用该矩阵，先求出矩阵的各元素的和 M，然后用上面矩阵中的各元素除以 M 就可以得到概率矩阵 x：

$$x=\begin{bmatrix} 0 & 0.002 & 0.013 & 0.014 & 0.013 & 0.002 & 0 \\ 0.002 & 0.008 & 0.023 & 0.026 & 0.023 & 0.008 & 0.002 \\ 0.013 & 0.023 & 0.044 & 0.056 & 0.044 & 0.023 & 0.013 \\ 0.014 & 0.026 & 0.056 & 0.071 & 0.056 & 0.026 & 0.014 \\ 0.013 & 0.023 & 0.044 & 0.056 & 0.044 & 0.023 & 0.013 \\ 0.002 & 0.008 & 0.023 & 0.026 & 0.023 & 0.008 & 0.002 \\ 0 & 0.002 & 0.013 & 0.014 & 0.013 & 0.002 & 0 \end{bmatrix}$$

利用该概率矩阵就可以对敌方进行随机试探了。

然后在决定性进攻阶段，我们还是以前面的思路来进行进攻。即取中的最大值所代表的位置去进行进攻。

在这一阶段的实践中，我们发现了一些问题，那就是我们的计算机进行扫描的是同一个方向，故如果被玩家发现这一点，那么，他就可以抓住计算机的这个弱点，将方向调为与计算机进行扫描方向的相反方向。

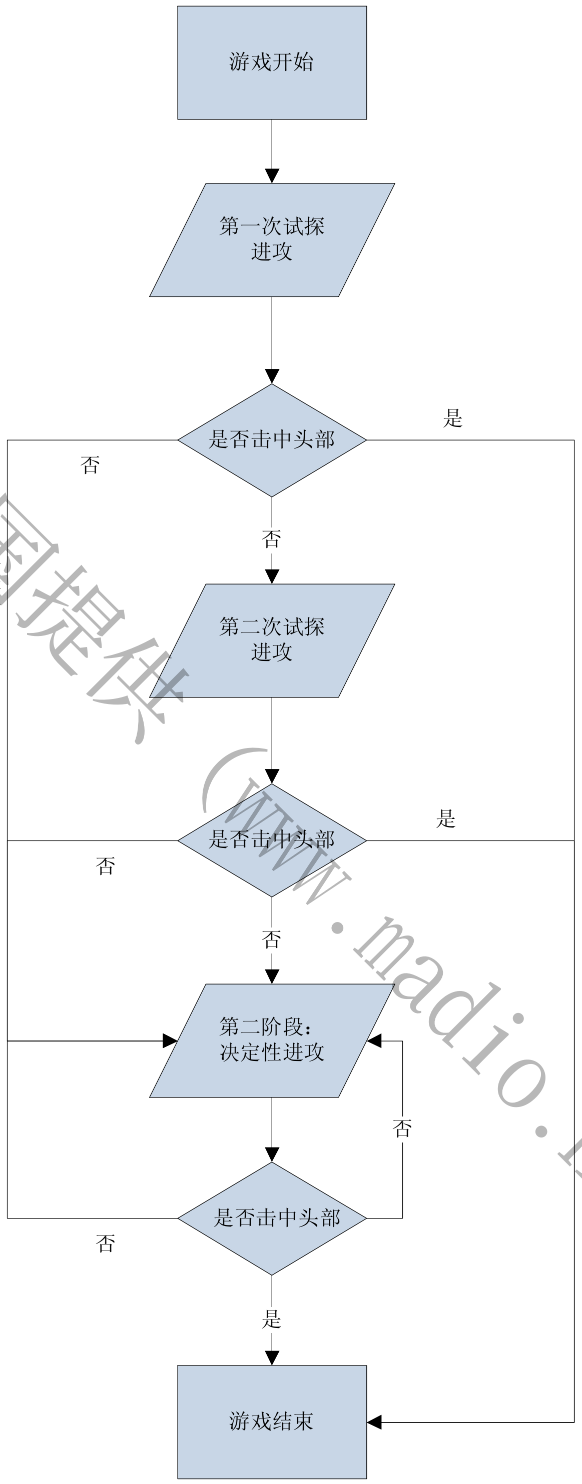
为了解决这个问题，我们再次引入随机模型，我们在扫描求最大值的过程中，取随机数，使得能在取最大值中（假如最大值有多个的话，而这种情况是经常会发生的）计算具有一种随机性。如下面的决策矩阵中：

$$CV(i,j)=\begin{bmatrix} 0 & 0 & 1 & 3 & 6 & 1 & 0 \\ 0 & 1 & 1 & 11 & 2 & 4 & 1 \\ 0 & 3 & 8 & 8 & 11 & 7 & 6 \\ 0 & 1 & 0 & 0 & 5 & 9 & 6 \\ 0 & 0 & 6 & 7 & 5 & 5 & 6 \\ 0 & 0 & 1 & 4 & 2 & 4 & 1 \\ 0 & 0 & 0 & 1 & 5 & 1 & 0 \end{bmatrix}$$

可以看出，在上述的决策矩阵中，有两个最大值点（4，6）、（5，5），那么，我们的程序就从这两个最大值点中随机取一个进行进攻。

综上所述，我们可以得该模型的流程图如下：

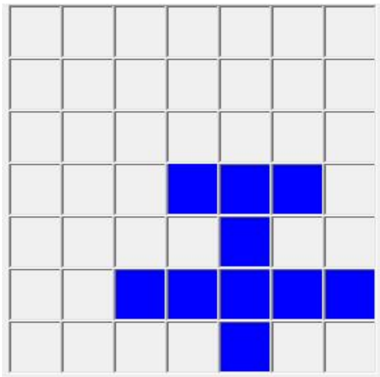
计算机进攻流程图



图表 4

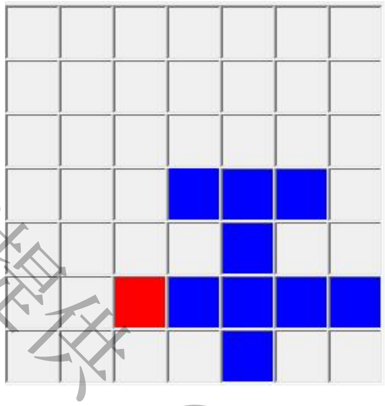
下面利用上述经过我们完善的模型来进行一次模拟进攻：

首先，为了体现计算机的能力，我们将飞机隐藏在一个角落，如图表 5 所示：



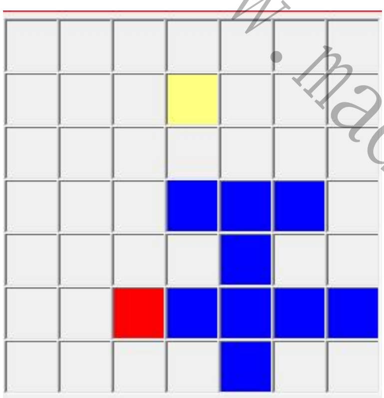
图表 5

然后，我们使计算机对玩家进行了第一次进攻，结果如图表 6 所示：



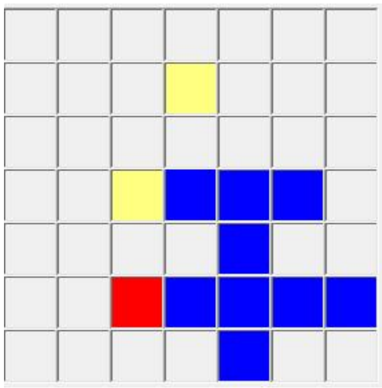
图表 6

从图表 6 可以看出，计算机在进行试探性进攻中攻到了敌方的机身。
图表 7 是第 2 次试探性进攻：



图表 7

下面是第二阶段决定性进攻：



图表 8

图表 8 表示第 3 次进攻，同时也是第 1 次决定性进攻。

在进行完这次进攻以后，就可以发现，决策矩阵变为：

$$CV(i,j)=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 5 & 0 & 0 \end{bmatrix}$$

从上面的可以看出，此时，机头的位置已经唯一确定了，即（5，1）。接下来，计算机就会对（5，1）点进攻，得到的结果如图表 9：

图表 9

通过上面的模拟进攻，我们发现 4 次就可以将对方击败，如果用以前的模型在这里要用到 8 次。可知，被完善后的模型，具有更好的稳定性，以及更强的攻击性。利用这个模型可以大大地减少进攻的步数，以达到更高的获胜率。

改进模型二：

该模型我们使用第二种策略方法，即为计算机设计出许多整体的策略，并且我们给出这些策略的混合策略模型，使计算机在实战中用我们的混合策略来进行随机地选择进攻策略。

关于各种整体策略的取法，我们可以将 $\frac{ph}{pb}$ 取不同的值，而得到不同的进攻

策略。如首先我们令 $\frac{ph}{pb}=4$，得到一个整体策略。然后，我们又可以令 $\frac{ph}{pb}=5$ 得到另一个整体策略。这样下去，我们就可以得一些可行的整体策略，将这些策略合在一起就可以形成一个计算机的整体策略集。

我们令 S_n 为我们所取得的所有整体策略集的个数。那么可得到整体策略的策略集为：

$$S_w=\left\{S_1,S_2,...,S_{S_n}\right\} \tag{2}$$

对上面的策略集，我们取混合策略：

$$X_w=\left\{x_1^w,x_2^w,...,x_{S_n}^w\right\} \tag{3}$$

其中，x_i^w 表示策略集中第 i 个策略被采取的概率。

那么现在我们就需要来计算出 X_w，首先，我们可以对 S_n 种策略，每一策略求出它击败敌机所需要的步数的期望。设第 i 步所得到的期望为 E_i，则可得到期望向量为：

$$E=(E_1,E_2,...,E_{S_n}) \tag{4}$$

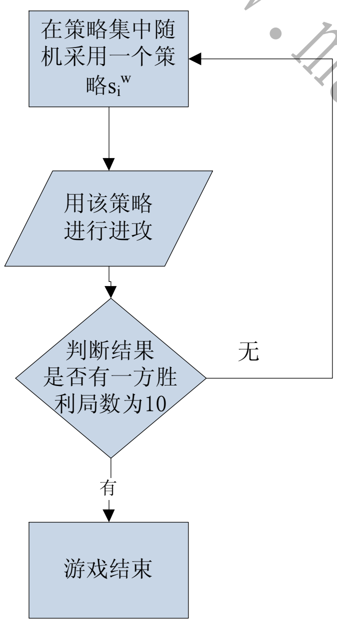
那么，令 E_w 为 E 的和：

$$E_w=\sum_{i=1}^{S_n}E_i \tag{5}$$

则根据上面的式子就可以求出 X_w 了：

$$X_w=\frac{E}{E_w} \tag{6}$$

模型二的进攻策略流程图如图表 10 所示：



图表 10

对问题二的分析

前一篇论文中问题二的模型：

第一阶段：

$$
\begin{cases}
p=1 \\
\downarrow \\
\text{启用模型一，求解攻击} \\
\downarrow \\
w(i,j,m,n,p)=0 \quad \text{此时 } p=1 \\
isFact(i,j,p)=\begin{cases} ph & \text{如果}(i,j)\text{位置是机头} \\ pb & \text{如果}(i,j)\text{位置是机身} \\ none & \text{如果}(i,j)\text{位置没有飞机} \end{cases} \\
w(i,j,m,n,p)=0 \quad \text{其中 } p=2 \text{ 且 } (m,n) \text{ 包含 } (i,j)
\end{cases}
$$

第二阶段：

$$
\begin{cases}
p=2 \\
\downarrow \\
\text{启用模型一，求解攻击} \\
\downarrow \\
p=2\text{ 时，攻击成功}
\end{cases}
$$

问题二改进后的模型：

对于问题二，我们在上次的研究成果上增加矩阵思想，使得本次的思想更易于实现，从而更容易将其转化成程序。

首先我们通过计算求出两架飞机可能的摆放位置的数量。我们的计算思路是这样的：

先得到一架飞机在9×9的格了上的所有可能性，可以得出，每个方向都有 30 种可能性。加起来就有 120 种可能性了。我们将我们的计算分为 120 种可能，每种可能求出另一架飞机的可能位置。我们用一种排除法，就是在另一架飞机的 120 种可能性中，进行排除，如果该飞机与另一架飞机有重合的部分就将它排除。设第i种第一架飞机的情况中与第一架飞机有重合部分的种数记为C_i，则总共有的可能性就为：

$$Z=\sum_{i=1}^{120}(120-C_i) \tag{7}$$

即在9×9的格子里有Z种可能的两个飞机的摆放位置。

我们也可用 1151 队的程序来计算Z，得到：

$$Z=4704$$

对于每一个摆放位置，我们都定义一个矩阵来表示我这个位置的飞机可能性。这样，我们就需要定义Z种个矩阵，在 VB 中，我们用二维数组来表示矩阵，而这里有Z个矩阵，那么，我们就可以用一个三维数组$MB(i,j,k)$来表示这Z个矩阵，其中，i,j表示二维矩阵，而k用来表示矩阵的个数，易知，i,j为 1 到 9 的数字，而k为 1 到Z的数字。在这里我们用矩阵的形式来表示关于i,j的数组，这样在计算上就可以将其认为是一个以k为维数的一维矩阵组，如其中

一种情况可用如下表示：

$$ME(i,j,1)=\begin{bmatrix}0&0&5&0&0&0&0&0&0\\1&1&1&1&1&0&5&0&0\\0&0&1&0&1&1&1&1&1\\0&1&1&1&0&0&1&0&0\\0&0&0&0&0&1&1&1&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\end{bmatrix}$$

在这个矩阵中的i,j只是表示一个矩阵，它在我们的 VB 计算中是有用的，而在此是它仅仅是表示说明这是一个矩阵，其中的 1 表示的是这个矩阵的种数，在此说明，这是第 1 种两架飞机的摆放情况。而在矩阵中的 5 表示的是机头，在矩阵中的 1 表示的是机身。

这样通过如上的设置就可以在初始状态下把所有可能的情况表示出来。然后，跟据我们上一篇论文的研究成果就可以对这Z个矩阵进行求和，就可以得到一个和矩阵，这个矩阵就是，我们在第一篇论文在提到的可能分布，即一个概率分布。跟据这个可能分布矩阵就可以求出可能性最大的点，以下是求和矩阵的公式：

$$CV(i,j)=\sum_{k=1}^{Z}MB(i,j,k) \tag{8}$$

其中，$CV(i,j)$在这里表示的是一个矩阵，而在实际的 VB 程序中，它表示的是一个数组，该数组等价于我们现在所讨论的矩阵。这样，这个问题就更加简化了。因为把这个和矩阵（即决策矩阵）求出来以后，就只需要求出，该和矩阵中的最大值就可以得到这我们需要进攻的点(i,j)了，于是，在此的目标函数为：

$$D_2=\max\{CV(i,j)\} \tag{9}$$

在此式中i,j都分别为 1 到 9 的数字。

通过对上述的可能性矩阵进行求和，然后对和矩阵进行求最大值，我们就可以得到我们要进行进攻的点(i,j)了。得到这点以后，跟据我们上一篇论文进行的分析，就可以进行我们动态规划的第二步了。

第二步：对目标位置(i,j)进行进攻并取得反馈信息。

对于(i,j)位置进行进攻，将得到三种结果，针对这三种结果，我们将$isFact(i,j)$进行设置。具体实现如下

$$isFact(i,j)=\begin{cases}ph & \text{当}(i,j)\text{位置为机头时}\\ pb & \text{当}(i,j)\text{位置为机身时}\\ none & \text{当}(i,j)\text{位置为没有机头也没有机身时}\end{cases}$$

这样第二步就可以这样完成了，接下来就可以进行我们的第三步：

第三步：对反馈信息的处理

得到了反馈信息以后就可以对其进行处理了，而反馈信息有三种，那么就可以对三种反馈信息进行分别处理。

第一种情况是没有击中，那么说明，在(i,j)位置的数值不为 0（是该位置有飞机的可能性矩阵）的矩阵，已经没有了意义，那么就可以将这些矩阵都设为 0。

第二种情况击中机身，那么说明，在在(i,j)位置的数值为 0（是该位置没有飞机的可能性矩阵）的矩阵，已经没有了意义，那么就可以将这些矩阵都设为 0。

第三种情况是击中的是机头。那么，我们就可以对AD（already die）进行自动加 1，然后，我们就可以进行判断。如果，此时$AD=2$说明，计算机已经获胜了。而如果此时$AD=1$，说明计算机此时已经进攻完毕一架飞机，而被攻到的飞机是会被公开的，那么就相当于进行了模拟的 10 次进攻（一架飞机占有 10 个格子）。如此，就可以对此飞机进行 10 次信息的自动反馈，可以从该处自动转到第二步进行第二步的信息反馈。

这样，我们的第二个关于9×9的飞机模型就建立了。它是完全建立在第一个模型关于7×7的格子的飞机模型的基础上的。而且，这次，我们使用的矩阵方法，在 VB 上利用数组进行实现，更加方便，更具有可操作性。

对于，随机模型，我们可以利用，此次关于第一个模型的改进而使用的方法，应用于这个9×9的模型中，因为，通过这次对9×9模型的改善，它已经与第一个模型有了很强的相似性了。

对问题三的分析

对于问题三，很明显，这是一个多人非合作的对策问题。关于对策问题，我们考虑与对策有关的因素：

局中人（Player）：就是参与对策的人或组织所组成的一人集合，用 N 表示：

$$N=\{1,2,3\}$$

在此，因为局中人只有 3 个，那我们就只考虑局中人总数为 3 的对策。

策略集（Strategy）：局中人所使用的策略的集合，用 S 表示：

$$S_i=\left\{S_1^i,S_2^i,...,S_{m_i}^i\right\}$$

局势（Situation）：当每一个局中人都从自己的策略集中选了一种策略以后就形成了一个局势，用 s 表示

$$s=(s_1,s_2,s_3)$$

对于每一个局势 s，局中人 i 得到的支付为：

$$\mathrm{P}_i(s)=P_i(s_1,s_2,s_3)\qquad i=1,2,3$$

对整个这样一个对策，我们用如下符合表示：

$$\Gamma=(N,\{S_i\},\{P_i\})$$

对于这样一个三人的非合作对策，我们用混合策略（即重复策略）来解。取混合策略：

$$X_i=(x_1^{(i)},x_2^{(i)},...,x_{m_i}^{(i)}),i=1,2,3$$

其中，$x_k^{(k)}\ge 0\quad (k=1,2,...,m_i),\ \sum_{k=1}^{m_i}x_k^{(i)}=1$。

接下来，我们对该问题进行具体的分析：

一般情况下，玩家在玩游戏是会出于两个方面的考虑来决定采用何种游戏策略，即进攻和防守，同样在本问题中我们也基于这种思想来设计我们的策略。

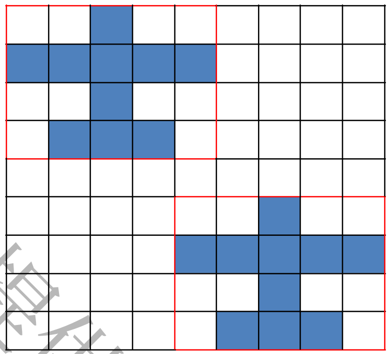
防守：对于防守我们主要来考虑如何来摆放两架飞机，两架飞机和一架飞

机最主要的区别就是这两架飞机之间会产生一定的影响，我们的目的就是如何来摆放飞机使这种影响达到最大。

为了分析这种影响，首先我们来定义两个飞机之间的影响度 Ef（effect）：用一个 4×5 或者 5×4 的矩形去覆盖表格中相应的区域，在覆盖时有可能会覆盖到两个飞机的机身中的某些点，我们将覆盖到的点的个数的最大值定义为两个飞机之间的影响度，记为 Ef（effect），并用它来衡量摆放飞机的可行性，Ef 值越大，说明此时的摆放策略可行性比较大，相反，如果 Ef 比较小，虽然此时的摆放策略是可行的，但是，被对手击中机头的可能性大大增加了。

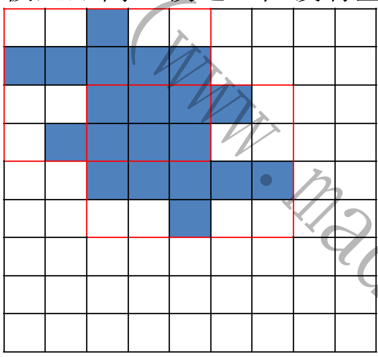
我们先来讨论这两架飞机的摆放方式。从空间的角度来看其应该有两种关系：

1、 两个飞机在表格中彼此距离比较远。（如图 11 所示）



图表 11

2、 两架飞机在表格中彼此距离比较近，但没有重叠部分。（如图 12 所示）



图表 12

下面我们先来分析一下一架飞机，对于一架飞机，无论怎样摆放，它都处于一个 4×5 或者 5×4 的矩形里，如果利用它们的第一种关系，那么就是要使它们所处的矩形不相交，如果要是利用它们的第二种关系，就是要使它们所处的矩形相交。

对于第一种情况；显然，在这种情况下飞机之间的影响度 Ef 会相对的小一些，所以我们不提倡采用此种摆放策略。

对于第二种情况:就是让两架飞机所处的矩形相交，在这种情况下，它们之间的 Ef 值相应的就会大一些，也就是说，采用第二种情况增加给其他选手下一步进攻的模糊度，从而降低其击中我们机头的可能性，所以，在实战中建议采用第二种情况去摆放飞机！

但是，在实战中如果总是采用第二种情况的策略，其他对手就会发现我们摆放飞机的特点，从而会针对我们摆放飞机的特点来设计出相应的策略，因而使得我们的危险系数增大。基于此种考虑，我们在实战中要“动静结合”，即在经过几个回合后，随机地尝试着去采用第一种情况的摆放策略。

综上所述，防守的策略集为：

$$SD_i = \left\{\text{相互距离小}, \text{相互距离大}\right\}, i = 1,2,3 \tag{10}$$

对上述策略集，取混合策略：

$$XD_i = \{xd_1^{(i)}, xd_2^{(i)}\}, i = 1,2,3 \tag{11}$$

根据上面的分析可知，在混合策略中：

$$xd_1^{(i)} \gg xd_2^{(i)} \qquad i = 1,2,3 \tag{12}$$

并且有：

$$xd_1^{(i)} + xd_2^{(i)} = 1 \qquad i = 1,2,3 \tag{13}$$

即在实战中，采用第一种策略的概率要比采用第二种策略的概率要大得多，也就是说，在多回合的比赛中，采用第二种策略只是偶尔的。

进攻:我们进攻的目的就是为了得到高分，如果基于这种高分的目的去考虑的话，我们应当去进攻那个机头位置相对明确的飞机。而在实战中，如果我们以全局的角度去考虑的话，在某一些回合中，我们不一定要去进攻那些相对明确的位置，而是去进攻在其它两个对手中总分最高的玩家的飞机，这样，如果目前我们的总分比其它两个对手都高的话，我们可以最大程度地拉大与第二名之间的差距，从而增加我们获胜的可能性，另一方面，如果我们总分不是最高的，可以最大程度地减小与第一名之间的差距，从而加快追赶第一名的步伐。

在一定程度上，这种进攻策略会有一定的风险，所以我们可以把这种进攻策略与另一种更加稳妥的进攻策略结合使用，这种稳妥的方式就是始终去进攻最有可能得分的位置。

综上所述，可知，进攻策略为：

$$SA_i = \{\text{风险型策略}, \text{稳妥型策略}\}, i = 1,2,3 \tag{14}$$

进攻的混合策略为：

$$XA_i = \{xa_1^{(i)}, xa_2^{(i)}\} \qquad i = 1,2,3 \tag{15}$$

其中：

$$xa_1^{(i)} + xa_2^{(i)} = 1 \qquad i = 1,2,3 \tag{16}$$

综合防守与进攻的策略，可得到每个人的策略集为：

$$\begin{aligned} S_i = \{&(\text{相对距离小}, \text{风险型进攻}), (\text{相对距离小}, \text{稳妥型进攻}), \\ &(\text{相对距离大}, \text{风险型进攻}), (\text{相对距离大}, \text{稳妥型进攻})\}, i = 1,2,3 \end{aligned} \tag{17}$$

通过上面的混合策略可求得整体的混策略为：

$$X_i = \{xd_1^{(i)} \times xa_1^{(i)}, xd_1^{(i)} \times xa_2^{(i)}, xd_2^{(i)} \times xa_1^{(i)}, xd_2^{(i)} \times xa_2^{(i)}\} \qquad i = 1,2,3 \tag{18}$$

其中有：

$$xd_1^{(i)} \times xa_1^{(i)} + xd_1^{(i)} \times xa_2^{(i)} + xd_2^{(i)} \times xa_1^{(i)} + xd_2^{(i)} \times xa_2^{(i)} = 1 \qquad i = 1,2,3 \tag{19}$$

关于上述混合策略的求解问题，首先我们知道，它是一定有解的。因为根据 Nash 定理：

任何 n 人对策 $\Gamma = (N, \{S_i\}, \{P_i\})$ 都有混合策略的平衡点（称为 Nash 平衡点）。

故由上述定理可得，上面的混合策略是有解的。

至于求解的方法，我们可以参考谢政的对策论，其中说，非合作对策与数学规划是用关系的，也就是说，非合作对策可以用数学规划来解，如对上面的混合策略可以用如下的数学规划来解：

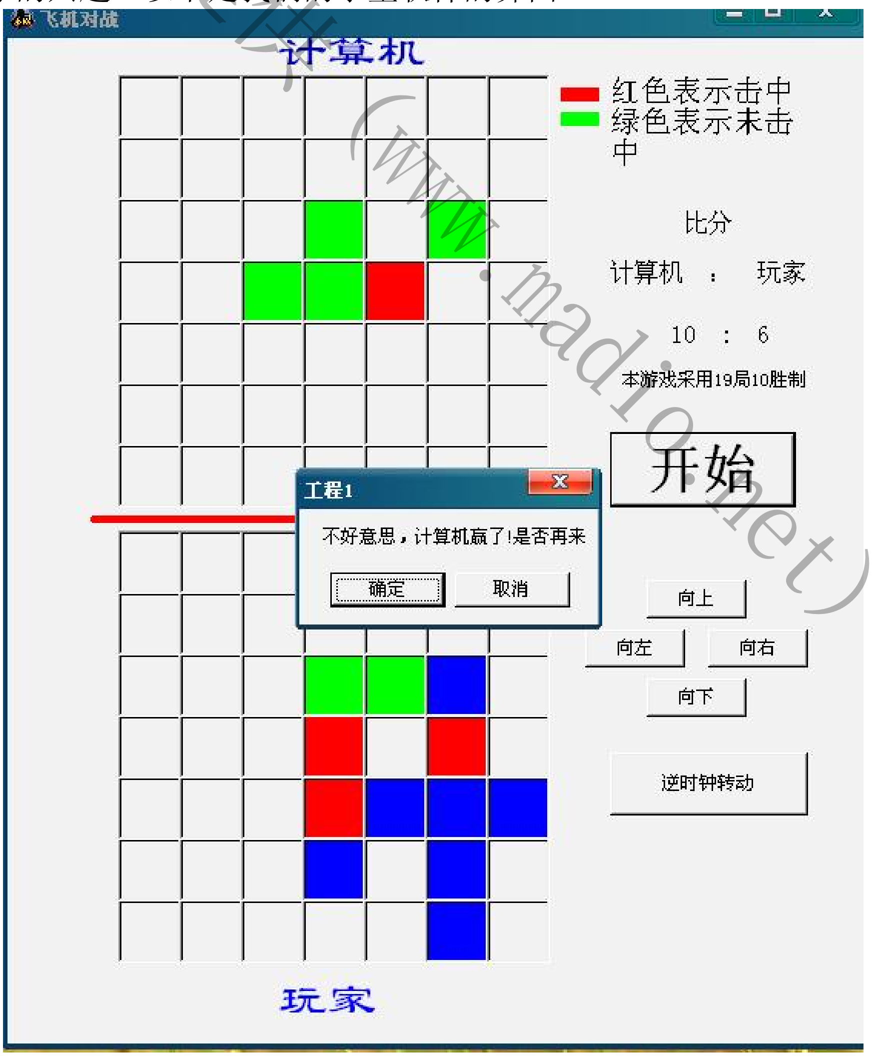
$$
\begin{cases}
\max f(X_1,X_2,X_3,v_1,v_2,v_3)=\sum\limits_{i=1}^{3}\left(\sum\limits_{k_1=1}^{4}\sum\limits_{k_1=2}^{4}\sum\limits_{k_1=3}^{4}a_{k_1k_2k_3}^{(i)}x_{k_1}^{(1)}x_{k_2}^{(2)}x_{k_3}^{(3)}-v_i\right)\\
s.t.\sum\limits_{k_1=1}^{4}\sum\limits_{k_2=1}^{4}\sum\limits_{k_3=1}^{4}a_{k_1k_2k_3}^{(i)}x_{k_1}^{(1)}x_{k_2}^{(2)}x_{k_3}^{(3)}\le v_i,\ \ i=1,2,3\\
\sum\limits_{k=1}^{4}x_k^{(i)}=1,\ \ i=1,2,3\\
x_k^{(i)}\ge 0,\ \ i=1,2,3;k=1,2,3,4
\end{cases}
$$

当给出每个局势的支付后就可以求出混合策略的解了。

五、　对模型的求解

我们取问题一的原模型的改进模型一，建立具有随机性以及进攻阶段性的混合策略模型，并在 VB 开发环境中进行具体实现。

通过多次调试程序，改正错误，我们得到了一个小型的游戏程序，并且进行了多数人的实际测试。我们发现在测试中，它具有很好的进攻性，并且很容易激发玩家的兴趣。以下是我们的小型软件的界面：



以上是我们的软件推出的第二个版本，第一个版本在实际测试中出现了诸多错误，于是我们对其进行了多次调试消除错误，并对其进行了更进一步的完善，从而推出了这第二个版本。

六、 结果分析

对于上面的软件，我们在测试过程中得到如下结果：

计算机在实践过程，获胜的概率非常大，在我们进行的测试中，尽管玩家用尽各种理性思维，但基本上还是计算机获胜。而只有少数情况下计算机险败于玩家，如以比分 9：10 险败给玩家。

从上述结果来看，该模型具有非常好的可行性。

七、 模型的进一步讨论

关于改进模型一的实现软件的讨论：

虽然，从整体看，我们的软件，已经没有错误了。但是，从玩家的角度看，它还存在着一些不方便的地方。

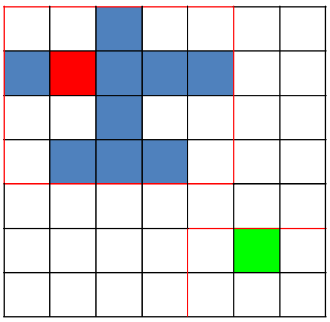
在实际测试中，从玩家的反馈信息来看，它在实践方面还存在一些不方便的因素。因此，我们打算对它进行进一步的完善，从而能推出第三个版本。我们打算尽量从玩家的角度使它变得更加方便，并且，另一方面，我们还要对算法进行进一步分析改善，使计算机获胜的可能性在原有基础上再次增加。

如，我们的初步改进方案为：

- ✧ 使玩家击中计算机的机头时，可以用红色显示该机头位置，这样，更能鼓励玩家进行下一局的进攻。
- ✧ 使计算机在击中玩家机头，本局将要结束时，把计算机的飞机用红色显示。这样，能给玩家一定的提示，以提高玩家的水平。
- ✧ 同时，我们还要对计算机的算法进行进一步改进，使计算机在随机进攻阶段更具有智能性。

改进后的模型一的讨论：

在试探进攻阶段，我们进行了两次试探，而我们所建立的模型在这两次试探时完全是随机的，并没有去仔细地考虑这两次试探进攻之间的联系。但在实战中，这两次试探进攻之间会有一定的联系，比如第一次试探性进攻进攻到（2，6）位置，如图 13 中红色位置所示：



图表 13

而第二次试探进攻进攻到（6，2）位置，如图 13 中绿色位置所示。根据我们前文的分析，飞机只能处在一个 4×5 或者 5×4 的矩形中，而在图表 13 中这两个点是不可能处于同一个 4×5 或者 5×4 的矩形中，也就是说，这两个点是不可能同时在一架飞机上的，所以，我们的第二次试探进攻在该回合中反馈的信息对我们在第二阶段决定性进攻时是没有太大帮助的！因此，我们可以进一步去优化试探进攻的策略，在第二次试探时，进行区域性的试探或者是不完全的随机试探，也就是说排除那些不可能在一架飞机上的位置，而去进攻其他位置。

关于开发 9×9 两架飞机游戏的讨论：

在上面的 7×7 一架飞机模型的基础上，我们将利用它的实现过程来进行 9×9 两架飞机软件的开发。并且我们将对其中的各种参数进行进一步的设定，如随机性阶段或是机头权数等。我们会再次利用混合策略进行随机性设定，有望在第三阶段开发出 9×9 两架飞机的软件。

八、 模型的优缺点分析与拓展方向

1. 优缺点分析

优点：模型在运用动态规划思想的基础上，针对各个问题考虑了各种随机因素对攻击策略的影响，如我们将机头与机身的权值比问题 $\frac{ph}{pb}$ 设定为 1~9 的变量，综合考虑各种因素对 $\frac{ph}{pb}$ 进行取值。并据此建立了相关的矩阵和概率模型，使得计算机在与玩家的游戏过程中，计算机的攻击策略集变得更广一些，且计算机在进攻过程中，首先有一个试探阶段，在几个不同的区域中验证是否存在飞机，这使得后面的进攻更具有很强的目的行，即在后面的决定性进攻阶段取胜的可能性非常大，文中就攻击系统研究的关键是目标选择与战术决策问题研究，并给出了基于一般性原则以及随机因素的数学模型和算法较好地满足和支持了攻击策略的给出与选择。

缺点：对于飞机在试探地过程中，试探的步数是一个难以确定的因素，玩家的进攻状态，即是冒险型还是稳重型是一个不确定的因素，因为对于每个人的进攻、防守情况是以每个人自己的主观意识决定的，因此很难根据这种人的主观因素设定算法，因此模型在考虑攻击策略试探阶段的步数时，只经过有限玩家群体试验来统计得出的拟合结果为 2 步，另外，在考虑机头预机身的权值问题时，虽然我们进行大量试验，最终确定为 $\frac{ph}{pb}=5$ 时，对于计算机的攻击策略稳定性较好，这仍是经过有限玩家群体的试验结果，并不一定适合每一个人。

在建模的过程中我们可能忽略了部分次要因素，如没考虑到局于局之间的影响、前一次的结果与后面的联系。在模型的分析、准备和模型的建立过程中，对一些数据不明确的前提下我们加入了一些主观因素，这对模型有一定的影响。另外，在计算机的运行所涉及的策略是按一系列算法设计的，没有主观能动性。

2. 模型的拓展方向

纵向拓展：我们可以在对战中，记录作战的过程，对这些数据进行一定的拟合，从而确定在进攻过程中，所采取的策略，这样能更大地提高成功率。增

加其它策略模型。根据模型编写不同的函数程序，然后用网格法求出综合评价值最高的策略模型。

同样我们可以结合考虑攻防两个方面来进行模型评价，讨论进攻和防守策略，对于 19 局 10 胜的赛制和两架飞机的情况进行处理，还可以运用几何概率统计原理和判断线性相关等运算方法，讨论模型的稳定性和智能性。

横向拓展：在该模型的基础上如果深入研究，对模型作完善的处理，考虑一系列心理等因素，则此模型可以运用到黑白棋、五子棋、象棋甚至是围棋的计算机设计和一系列的益智玩具的开发上。

九、 模型的评价与推广

模型较好的运用对策论和动态规划的思想，使模型在推广和应用方面具有较强的稳定性和通用性。在建立模型中我们尽可能的不拘泥于原有的数据和原有的条件，从而使模型具有较好的可改变性和可移植性，在改变一定的环境条件下，此模型仍具有较强的适用性（如五子棋游戏的设计）。由于所给问题是一个实际问题，因此模型的建立是在一定的实际应用背景下完成的，而且许多游戏的最终目的是商业，因此所建的模型具有一定的商业价值，为各种人工智能小游戏的开发提供了理论依据。结果有着较好的稳定性。我们对已知条件进行了进一步的分析，结果发现结论允许测量值的变化范围是很大的，即使我们在测量时有一定的误差，最优解也不会发生变化，从而说明了结果具有较好的稳定性。

本模型所用的思想为博弈论或“游戏理论”（Game Theory）,即参与竞争的一些独立决策的个人、组织（团、队、国家等）面对一定的环境条件，在一定的竞争规则的约束下，从各自允许的策略集中选出有效的策略，达到以谋略取胜的思维与较量的行为过程[8]，在系统的分析和辩证的思维看待竞争问题后采用策略理论和规划的方法建立合理的模型，具有较强的推广性和应用价值。

十、 参考文献

[1] 张云青译，张国维校.什么是人工智能(AI). 国外先进技术 .第8卷第4期 1993年8月
[2]谢政，对策论.长沙：国防科技大学出版社，2004.3
[3]刘德铭 黄振高，对策论及其应用.长沙：国防科技大学出版社，1994.12
[4]1151队，人机对战游戏的策略分析.数学中国杯网络挑战赛第一阶段，2008.4
[5]张艳，新编 Visual Basic 面向对象程序设计.徐州：中国矿业大学出版社，2007.6
[6]王才宏 杨世荣 董 茜，目标选择决策的组合熵权系数方法研究.弹箭与制导学报,第 26 卷第 4 期.www.paper.edu.com
[7]谷源盛.运筹学.重庆：重庆大学出版社，2001.8

十一、　附录

```
'////////////////////////计算机飞机的随机设置程序///////////////////////
Dim suiji As Integer
Dim x0 As Integer, y0 As Integer
Randomize
suiji = Int(Rnd * 4) + 1
direct1 = suiji
Select Case direct1
    Case 1
        x0 = Int(Rnd * 3) + 3
        y0 = Int(Rnd * 4) + 4
        'head1(x0, y0) = -1
    Case 2
        x0 = Int(Rnd * 4) + 4
        y0 = Int(Rnd * 3) + 3
    Case 3
        x0 = Int(Rnd * 3) + 3
        y0 = Int(Rnd * 4) + 1
    Case 4
        x0 = Int(Rnd * 4) + 1
        y0 = Int(Rnd * 3) + 3
End Select
head1(x0, y0) = True
'////////////求概率矩阵 X 的程序///////////////////////
For i = 1 To 7
    For j = 1 To 7
        bb = cubeValue(i, j) + bb
    Next j
Next
For i = 1 To 7
    For j = 1 To 7
        Form2.Text1.Text = Form2.Text1.Text & _
        (Int((cubeValue(i, j) / bb) * 1000)) / 1000 & "      "
    Next
    Form2.Text1.Text = Form2.Text1.Text & vbCrLf
Next
'////////////重新开始一局的过程///////////////////////
Private Sub Restart()
    isTurn = 1
    way = 0

    For i = 1 To 7
```

```
        For j = 1 To 7
            head(i, j) = 0
            head1(i, j) = 0
        Next j
    Next i

    score.Caption = jc & "    :    " & wc

    isStart = False
    head(4, 5) = -1
    Direct = 1

    For i = 1 To 49
            lblCube(i).BackColor = &H8000000F
            lblCube1(i).BackColor = &H8000000F
    Next i

    lblCube(32).BackColor = &HFF0000
    For i = 0 To 4
        lblCube(23 + i).BackColor = lblCube(32).BackColor
    Next i
    lblCube(18).BackColor = lblCube(32).BackColor
    lblCube(10).BackColor = lblCube(32).BackColor
    lblCube(11).BackColor = lblCube(32).BackColor
    lblCube(12).BackColor = lblCube(32).BackColor

End Sub
'////////////////////////探测程序//////////////////////////////////////////
Private Sub tanCe()
    Randomize
    If Rnd * 100 >= 90 Then
        Call tanCe1
        Exit Sub
    End If
    maxX = Int(Rnd * 7) + 1
    If maxX >= 3 And maxX <= 5 Then
        maxY = Int(Rnd * 7) + 1
    Else
        maxY = Int(Rnd * 3) + 3
    End If
    If maxX = 7 Then maxX = 6
    If maxY = 7 Then maxY = 6
    If maxX = 1 Then maxX = 2
    If maxY = 1 Then maxY = 2
```

```
        If (Not (lblCube((maxY - 1) * 7 + maxX).BackColor = &H8000000F Or _
        lblCube((maxY - 1) * 7 + maxX).BackColor = &HFF0000)) _
        Or cubeValue(maxX, maxY) = 0 Then
            Call tanCe
        End If
End Sub
Private Sub tanCe1()
        Randomize
        maxX = Rnd * 2 + 1
        If maxX = 2 Then
            maxX = 7
        End If
        maxY = Rnd * 2 + 1
        If maxY = 2 Then
            maxY = 7
        End If
        If (Not (lblCube((maxY - 1) * 7 + maxX).BackColor = &H8000000F Or _
        lblCube((maxY - 1) * 7 + maxX).BackColor = &HFF0000)) _
        Or cubeValue(maxX, maxY) = 0 Then
            Call tanCe1
        End If
End Sub
'///////////////////////////////计算决策矩阵程序///////////////////////
For j = 7 To 1 Step -1
        For i = 1 To 7
                cubeValue(i, j) = 0
                If (lblCube((j - 1) * 7 + i).BackColor = &H8000000F Or lblCube((j _
 – 1 ) * 7 + i).BackColor = &HFF0000) Then
                    For m = 11 To 77
                        For n = 1 To 4
                            cubeValue(i, j) = cubeValue(i, j) + w(i, j, m, n)
                        Next n
                    Next m
                End If
                Form2.Show
                Form2.Text1.Text = Form2.Text1.Text & cubeValue(i, j) & Space(3)
                If cubeValue(i, j) > max Then
                    max = cubeValue(i, j)
                    maxX = i
                    maxY = j
                End If
        Next
        Form2.Text1.Text = Form2.Text1.Text & vbCrLf
Next
```