人猫鸡米渡河问题的数学模型

摘要：人带着猫、鸡、米过河，从左岸到右岸，船除了需要人划之外（船除了要载人外），只能载猫、鸡、米三者之一，而当人不在场时猫要吃鸡、鸡要吃米。本文将设计一个安全过河方案，使渡河次数尽量地少。模仿“商人过河”的模型设计出新的数学模型。

关键字：穷举法，**Matlab** 运算求解。

一、问题的提出

课本 **P19.T5:** 模仿“商人过河”模型，做下面游戏：人带着猫、鸡、米过河，船除需要人划之外，至多能载猫、鸡、米三者之一，而当人不在场时猫要吃鸡、鸡要吃米。设计一个过河方案，建立数学模型，并使渡河次数尽量地少。

二、问题的分析

因为这是个简单问题，研究对象少，所以可以用穷举法，简单运算即可解题。

此问题是从状态向量 A（1,1,1,1）经过奇数次运算向量 B 变为状态向量 A（0,0,0,0）的状态。转移过程为什么是奇数次？我们注意到过河有两种，奇数次的为从左岸到右岸，而偶数的为右岸回到左岸，因此得到下述转移过程，所以最后应该是过河完成时状态转移数为奇数次。

三、问题的假设

1.1：假设船除了载人之外，至多只能载猫、鸡、米三者之一。

1.2：当人不在场时，猫一定会吃鸡、鸡一定会吃米。

四、定义符号说明：

我们将人，猫，鸡，米依次用四维向量中的分量表示，当一物在左岸时，相应的分量

记为 1，在右岸时记为 0.如向量（1,0,1,0）表示人和鸡在左岸，猫和米在右岸，并将这些向量称为状态向量。例如（1,1,1,1）表示它们都在左岸，（0,1,1,0）表示猫，鸡在左岸，人，米在右岸；由于问题中的限制条件，有些状态是允许的，有些状态是不允许的。凡问题可以允许存在的状态称为可取状态。A 向量定义为状态变量。比如 $A_1(1,0,1,0)$ 是一个可取状态向量，但 $A_2(0,0,1,1)$ 是一个不可取状态向量。此外，B 向量定义为运载变量。把每运载一次也用一个四维向量来表示。如 $B_1(1,1,0,0)$ 表示人和猫在船上，而鸡和米不在船上，这自然是可取的运载，因为船可载两物，而 $B_2(1,0,1,1)$ 则是不可取运载，依此规律类推。

五、模型的建立

对于这个问题我们用穷举的方法来解决，首先将此问题化为状态转移问题来解决。对本问题来说：

5.1、 可取状态向量 A 共有 10 个，可以用穷举法列出来：

$(1,1,1,1)$ $(0,0,0,0)$

$(1,1,1,0)$ $(0,0,0,1)$

$(1,1,0,1)$ $(0,0,1,0)$

$(1,0,1,1)$ $(0,1,0,0)$

$(1,0,1,0)$ $(0,1,0,1)$

右边 5 个正好是左边 5 个的相反状态。

5.2、 可取运载 B 共有 4 个：

$(1,1,0,0)$ $(1,0,1,0)$

$(1,0,0,1)$ $(1,0,0,0)$

5.3、可取运算：规定 A 与 B 相加时对每一分量按二进制法则（异或运算）进行 $(0+0=0,1+0=0+1=1,1+1=0)$。这样，一次渡河就是一个可取状态向量与一个可取运载向量相加，可取状态经过加法运算仍是一个可取状态，这种运算称为可取运算。

在上述规定下，问题转化为：从初始状态 $(1,1,1,1)$ 至少经过多少次（奇数次）可取运算才能转化为状态 $(0,0,0,0)$。

六、模型的求解

如果一个状态是可取的就打∨，否则就打×，虽然可取但已重复就打∧，于是问题可用穷举法解答如下：

1）$(1,1,1,1)+\begin{cases}(1,0,1,0)\\(1,1,0,0)\\(1,0,0,1)\\(1,0,0,0)\end{cases}\rightarrow\begin{cases}(0,1,0,1)\vee\\(0,0,1,1)\times\\(0,1,1,0)\times\\(0,1,1,1)\times\end{cases}$

（2）$(0,1,0,1)+\begin{cases}(1,0,1,0)\\(1,1,0,0)\\(1,0,0,1)\\(1,0,0,0)\end{cases}\rightarrow\begin{cases}(1,1,1,1)\wedge\\(1,0,1,1)\times\\(1,1,0,0)\times\\(1,1,0,1)\vee\end{cases}$

（3）$(1,1,0,1)+\begin{cases}(1,0,1,0)\\(1,1,0,0)\\(1,0,0,1)\\(1,0,0,0)\end{cases}\rightarrow\begin{cases}(0,1,1,1)\times\\(0,0,0,1)\vee\\(0,1,0,0)\vee\\(0,1,0,1)\wedge\end{cases}$

（4）$_1(0,0,0,1)+\begin{cases}(1,0,1,0)\\(1,1,0,0)\\(1,0,0,1)\\(1,0,0,0)\end{cases}\rightarrow\begin{cases}(1,0,1,1)\vee\\(1,1,0,1)\wedge\\(1,0,0,0)\times\\(1,0,0,1)\times\end{cases}$

（4）$_2(0,1,0,0)+\begin{cases}(1,0,1,0)\\(1,1,0,0)\\(1,0,0,1)\\(1,0,0,0)\end{cases}\rightarrow\begin{cases}(1,1,1,0)\vee\\(1,0,0,0)\times\\(1,1,0,1)\wedge\\(1,1,0,0)\times\end{cases}$

（5）$_1(1,0,1,1)+\begin{cases}(1,0,1,0)\\(1,1,0,0)\\(1,0,0,1)\\(1,0,0,0)\end{cases}\rightarrow\begin{cases}(0,0,0,1)\wedge\\(0,1,1,1)\times\\(0,0,1,0)\vee\\(0,0,1,1)\times\end{cases}$

（5）$_2(1,1,1,0)+\begin{matrix}(1,0,1,0)\\(1,1,0,0)\\(1,0,0,1)\\(1,0,0,0)\end{matrix}\rightarrow\begin{cases}(0,1,0,0)\wedge\\(0,0,1,0)\vee\\(0,1,1,1)\times\\(0,1,1,0)\times\end{cases}$

（6）$(0,0,1,0)+\begin{cases}(1,0,1,0)\\(1,1,0,0)\\(1,0,0,1)\\(1,0,0,0)\end{cases}\rightarrow\begin{cases}(1,0,0,0)\times\\(1,1,1,0)\wedge\\(1,0,1,1)\wedge\\(1,0,1,0)\vee\end{cases}$

（7）$(1,0,1,0)+\begin{cases}(1,0,1,0)\\(1,1,0,0)\\(1,0,0,1)\\(1,0,0,0)\end{cases}\rightarrow\begin{cases}(0,0,0,0)\vee\\(0,1,1,0)\times\\(0,0,1,1)\times\\(0,0,1,0)\wedge\end{cases}$

第 7 步已经出现了$(0,0,0,0)$状态，说明经 7 次运载即可，其过程为：

$\xrightarrow{去}(人,鸡)\xrightarrow{回}(人)\xrightarrow{去}(人,猫（或米）)\xrightarrow{回}(人，鸡)\xrightarrow{去}(人,米（或猫）)\xrightarrow{回}(人)\xrightarrow{去}(人,鸡)$

因此，该问题的最优方案有 2 种：

1、人先带鸡过河，然后人再回来，把米带过河，然后把鸡运回河岸，人再把猫带过河，最后人回来把鸡带过去。

2、人先带鸡过河，然后人再回来，把猫带过河，然后把鸡运回河岸，人再把米带过河，最后人回来把鸡带过去。

七、模型的评价

7.1、优点：

本算法将研究对象用四维向量中的分量用 0,1 表示，运用穷举法找出所有可取状态向量再用一些基础可取运算方法将结果列出来再以图形表示出来。模型简单，切合实际，易于理解，整个过程易懂合理。

7.2、缺点：

由于问题的求解没有使用 LINGO，LINDO 或 MATLAB 软件，当状态和决策过多时，采用上述方法求解显得繁琐，容易出错，所以下面给出此问题的 matlab 求解过程。

7．3、推广：

正如课本上的商人们安全过河问题，当商人和随从人数增加或小船的容量加大时，靠逻辑思考就有些困难了，而适当地设置状态和决策，确定状态转移率，建立多步决策模型，仍可方便有效地求解此类型问题。

八、mathlab 求解过程

8.1、模型假设与建立：

8.1.1、由上可知，可取状态向量 A 共有 10 个，即：

$(1,1,1,1)$ $(0,0,0,0)$

$(1,1,1,0)$ $(0,0,0,1)$

$(1,1,0,1)$ $(0,0,1,0)$

$(1,0,1,1)$ $(0,1,0,0)$

$(1,0,1,0)$ $(0,1,0,1)$

8.1.2、 可取运载 B 有 4 个 ：

（1,1,0,0）、 （1,0,1,0）、 （1,0,0,1）、 （1,0,0,0）。

8.2、算法设计：

8.2.1、规定 A 和 B 的每一分量相加时按二进制法则进行，这样一次渡河就是一个可取状态和一个可取运载相加，在判断和向量是否属于可取状态即可。

8.2.2、可以将可取状态及可取运载分别编成矩阵。共分为五个 m 文件，一个主文件 xduhe.m 数，四个子文件分别为：

8.2.2.1、duhe（L,B,M,s）函数：

用来实现渡河总思路。思路为：将起始矩阵 A 分别与可取运载相加（使用二进制法则），判断相加后的矩阵 C 是否是（0，0，0，0），如果是，则渡河成功。否则，用 fuhe(C,M) 函数判断 C 是否是可取状态，如果是，则打印并将 C 与初始矩阵合并成新矩阵，继续调用 duhe.m 函数。

8.2.2.2、fuhe(C,M)函数：

判断和矩阵 C 是否属于矩阵 M，如果是，则返回 1，否则返回 0.

8.2.2.3、Panduan（S）函数：

判断 S 矩阵中是否有两个相同的状态，即行向量。如果有，则返回 0，否则返回 1.

8.2.2.4、print(K,C,s)函数：

打印相应的状态。

8.3、程序代码：

8.3.1、xduhe.m 文件：

```
clear;clc;
A=[1,1,1,1];
B=[1,0,1,0;1,1,0,0;1,0,0,1;1,0,0,0];
M=[1,1,1,0;0,0,0,1;1,1,0,1;0,0,1,0;1,0,1,1;0,1,0,0;1,0,1,0;0,1,0,1];
duhe(A,B,M,1);
```

8.3.2、duhe.m 文件：

```
function duhe(L,B,M,s);
[h,l]=size(L);
for k=s:h
    for i=1:4
        C=mod(L(k,:)+B(i,:),2);
         if C==[0,0,0,0]
              print(B(i,:),C,s);
                  fprintf('渡河成功\n\n');
                  break;
          else if fuhe(C,M)==1
                  print(B(i,:),C,s);
                  S=[L;C];
                   if Panduan(S)==1
                   duhe(S,B,M,s+1);
                 else
                   fprintf('此渡河方案不可行\n\n');
                 end
             end
        end
    end
end
```

8.3.3、fuhe.m 文件：

```
function y=fuhe(C,M)
y=0;
for i=1:8
    if(C==M(i,:))
        y=1;
        break;
    end
end
```

8.3.4、Panduan.m 文件：

```
function z=Panduan(S)
z=1;
[m,n]=size(S);
for p=1:m
    for q=(p+1):m
        if S(p,:)-S(q,:)==[0,0,0,0]
            z=0;
            break;
        end
    end
end
```

8.3.5、print.m 文件：

```
function print(K,C,s)
fprintf('第%d 次渡河:',s);
if K(1)==1
    fprintf('人, ');
end
if K(2)==1
    fprintf('猫, ');
end
if K(3)==1
    fprintf('鸡, ');
end
if K(4)==1
    fprintf('米, ');
end
if C(1)==0
    fprintf('从左岸到达右岸\n');
else
    fprintf('从右岸回到左岸\n');
end
```

8.4、模型结论

在 matlab 中运行，结果如下：

```
第1次渡河:人，鸡，从左岸到达右岸
第2次渡河:人，从右岸回到左岸
第3次渡河:人，猫，从左岸到达右岸
第4次渡河:人，鸡，从右岸回到左岸
第5次渡河:人，鸡，从左岸到达右岸
此渡河方案不可行

第5次渡河:人，米，从左岸到达右岸
第6次渡河:人，猫，从右岸回到左岸
第7次渡河:人，鸡，从左岸到达右岸
第8次渡河:人，鸡，从右岸回到左岸
此渡河方案不可行

第8次渡河:人，米，从右岸回到左岸
此渡河方案不可行

第7次渡河:人，猫，从左岸到达右岸
此渡河方案不可行

第6次渡河:人，米，从右岸回到左岸
此渡河方案不可行

第6次渡河:人，从右岸回到左岸
第7次渡河:人，鸡，从左岸到达右岸
渡河成功

第4次渡河:人，猫，从右岸回到左岸
此渡河方案不可行

第3次渡河:人，米，从左岸到达右岸
第4次渡河:人，鸡，从右岸回到左岸
第5次渡河:人，鸡，从左岸到达右岸
```

```
此渡河方案不可行

第5次渡河:人，猫，从左岸到达右岸
第6次渡河:人，猫，从右岸回到左岸
此渡河方案不可行

第6次渡河:人，米，从右岸回到左岸
第7次渡河:人，鸡，从左岸到达右岸
第8次渡河:人，鸡，从右岸回到左岸
此渡河方案不可行

第8次渡河:人，猫，从右岸回到左岸
此渡河方案不可行

第7次渡河:人，米，从左岸到达右岸
此渡河方案不可行

第6次渡河:人，从右岸回到左岸
第7次渡河:人，鸡，从左岸到达右岸
渡河成功

第4次渡河:人，米，从右岸回到左岸
此渡河方案不可行

第3次渡河:人，从左岸到达右岸
此渡河方案不可行
```

从运行结果可以看出，共有两种运送方案：

1、 人先带鸡过河，然后人再回来，把米带过河，然后把鸡运回河岸，人再把猫带过河，最后人回来把鸡带过去。

2、 人先带鸡过河，然后人再回来，把猫带过河，然后把鸡运回河岸，人再把米带过河，最后人回来把鸡带过去。

8.5、收获与不足：

复习和加深了对 matlab 的学习，见识了 MATLAB 7.10.0(R2010a)的厉害，引发了我对数学建模的强大兴趣。而且，利用 MATLAB 可以非常容易而且很直观地得出问题的解决方案，从而在不断深化对问题认识的同时，把问题普遍化。但是这个问题的求解过程是请教了师姐才得以完成的。

九、“商人过河”的算法步骤和 matlab 求解过程

另外，由于本次数学建模的启发，我们组根据老师上课讲的“商人过河”的算法步骤，用 matlab 编程出程序，求解过程如下：

9.1、模型建立：

此问题可视为一个 多步决策

多步决策:决策过程难以一次完成，而要分步优化，最后获取一个全局最优方案的决策方法称为多步决策。问题：每一步就是一次渡河，每次渡河就是一次状态转移。

用三维变量(x,y,z)表示状态：

x ------商人数　　x 的取值范围：{0，1，2，3}；

y ------随从数　　y 的取值范围：{0，1，2，3}；

z ------船　　z 的取值范围：{0,1}。

那么安全状态（可取向量）可表示为：

$$\text{安全状态}=\begin{cases} y=0,1,2,3 & x=0,3 \\ y=x & x=1,2 \end{cases}$$

安全状态：商人们安全是指在两岸都安全，故当 x=0,3 时，y=0,1,2,3，而当 x=1,2 时,此岸要求 x≥y,对岸要求 3-x≥3-y,综合即 x=y；

安全状态：商人们安全是指在两岸都安全，故当 x=0,3 时，y=0,1,2,3，而当 x=1,2 时,此岸要求 x≥y,对岸要求 3-x≥3-y,综合即 x=y；

(3,3,1)	(3,2,1)	(3,1,1)	(2,2,1)	(3,0,1)	(0,3,1)	(0,2,1)	(1,1,1)	(0,1,1)
(3,2,0)	(3,1,0)	(2,2,0)	(3,0,0)	(0,3,0)	(0,2,0)	(1,1,0)	(0,1,0)	(0,0,0)

这就是此问题的数学模型。

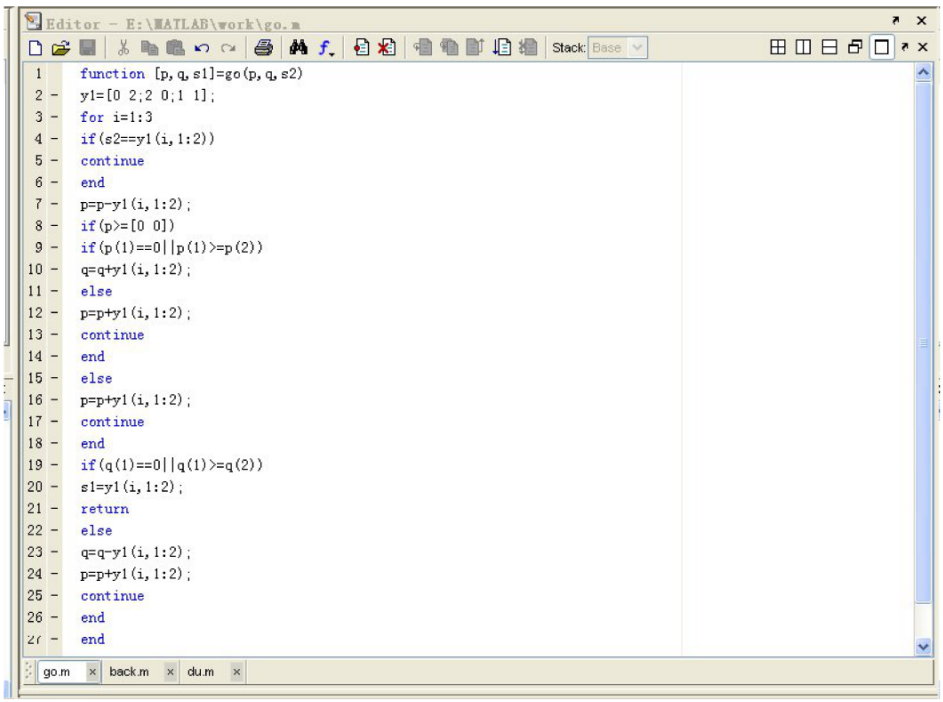
9.2、模型求解：

这样问题要求由(3，3，1)变到(0，0，0)的一条道路。

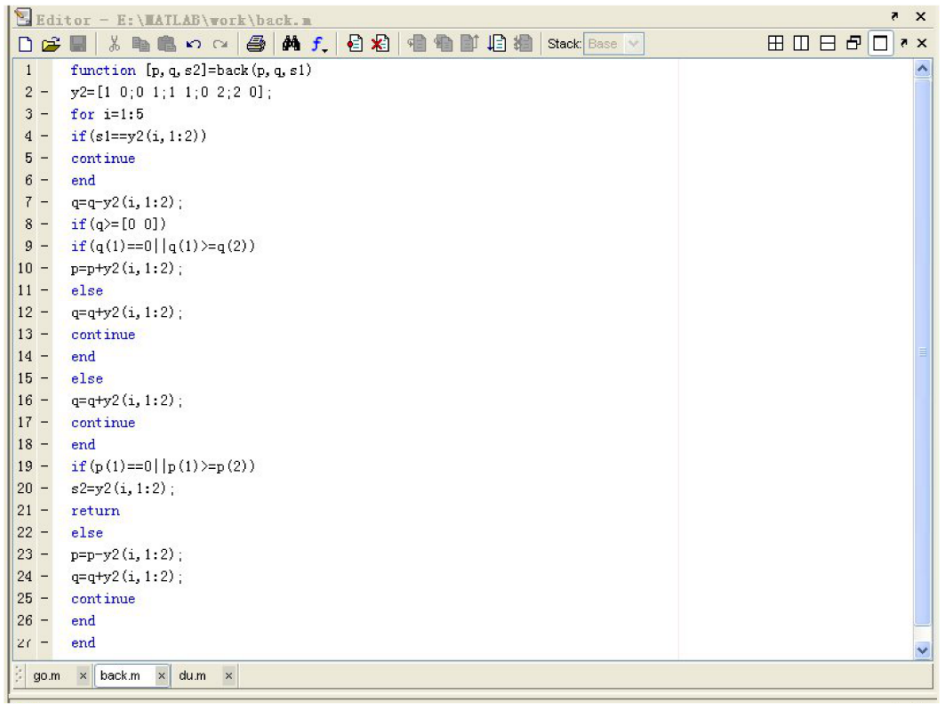
根据题意，状态转移时要满足一定的规则：

1. Z 从 1 变为 0 与从 0 变为 1 交替进行；
2. 当 Z 从 1 变为 0 时，即船从此岸到对岸，此岸人数减少 1 或 2 个；
   即$(x,y,1)\rightarrow(u,v,0)$时，$u\leqslant x$，$v\leqslant y$，$u+v=x+y-1$ or $u+v=x+y-2$；
3. 当 Z 从 0 变为 1 时，即船从对岸到此岸，此岸人数增加 1 或 2 个；
   即$(x,y,0)\rightarrow(u,v,1)$时，$u\geqslant x$，$v\geqslant y$，$u+v=x+y+1$ or $u+v=x+y+2$；
4. 不重复已出现过的状态，如$(3,3,1)\rightarrow(3,1,0)\rightarrow(3,3,1)$；

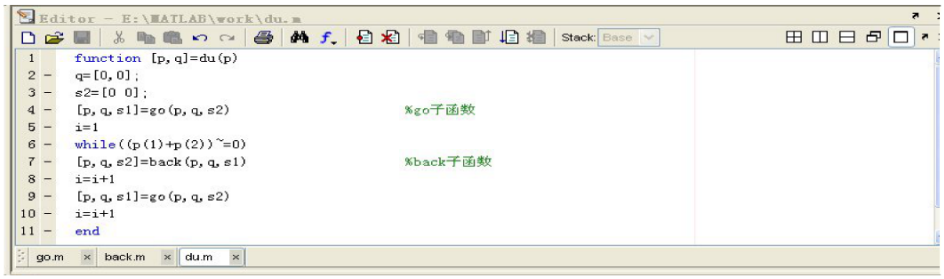
9.3、matlab 求解：



```
function [p,q,s1]=go(p,q,s2)
y1=[0 2;2 0;1 1];
for i=1:3
if(s2==y1(i,1:2))
continue
end
p=p-y1(i,1:2);
if(p>=[0 0])
if(p(1)==0||p(1)>=p(2))
q=q+y1(i,1:2);
else
p=p+y1(i,1:2);
continue
end
else
p=p+y1(i,1:2);
continue
end
if(q(1)==0||q(1)>=q(2))
s1=y1(i,1:2);
return
else
q=q-y1(i,1:2);
p=p+y1(i,1:2);
continue
end
end
```



```
function [p,q,s2]=back(p,q,s1)
y2=[1 0;0 1;1 1;0 2;2 0];
for i=1:5
if(s1==y2(i,1:2))
continue
end
q=q-y2(i,1:2);
if(q>=[0 0])
if(q(1)==0||q(1)>=q(2))
p=p+y2(i,1:2);
else
q=q+y2(i,1:2);
continue
end
else
q=q+y2(i,1:2);
continue
end
if(p(1)==0||p(1)>=p(2))
s2=y2(i,1:2);
return
else
p=p-y2(i,1:2);
q=q+y2(i,1:2);
continue
end
end
```



```
function [p,q]=du(p)
q=[0,0];
s2=[0 0];
[p,q,s1]=go(p,q,s2)                    %go子函数
i=1
while((p(1)+p(2))~=0)
[p,q,s2]=back(p,q,s1)                  %back子函数
i=i+1
[p,q,s1]=go(p,q,s2)
i=i+1
end
```

输入下面的语句将出现结果：（运行结果和预期结果一样）

```
>> p=[3 3];
[p,q]=du(p)
```

……

十、参考文献：

10.1、周义仓、赫孝良　《数学建模实验》，西安交通大学出版社。

10.2、姜启源、谢金星、叶俊　《数学模型》（第四版），高等教育出版社。

10.3、刘锋　　《数学建模 》　南京大学出版社　　2005 年 9 月等。