# 数学建模第一次作业

# 黑白棋问题建模

## 

**摘要：**

黑白棋问题是指将任意颜色的黑白棋N个排成一圈，重复这样的操作：在相同颜色的棋子之间放一颗黑色棋子，在不同颜色的棋子之间放一颗白色棋子，并拿去原来的棋子。本文阐述了一种将棋子及其之间的操作进行抽象化表达，并利用编程工具分析棋子颜色的变化的方法。

**关键词：黑白棋 二进制 逻辑代数**

**正文：**

**一、问题重述**

任意拿出黑白两种颜色的棋子N个,排成一个圆圈.然后在两颗颜色相同的棋子中间放一颗黑色棋子,在两颗颜色不同的棋子中间放一颗白色棋子,放完后撤掉原来所放的棋子.再重复以上的过程,这样放下一圈后就拿走前次的一圈棋子,这样重复进行下去各棋子的颜色会怎样变化呢?

**二、模型建立**

结合数字逻辑、离散数学的知识，不难想到可以将棋子的颜色抽象为二进制布尔代数，即黑色棋子为1（True），白色棋子为0（False）；而由两颗棋子生成棋子的过程为：00或11生成1，01或10生成0，这恰巧对应了逻辑代数的同或运算（⊙）。

取一颗棋子的标号为1，顺时针对所有棋子进行标号到N。设j号和j+1号棋子经过一次变换生成的棋子为j号，且N号和1号生成的棋子为N号。由此可设表示经过第i次变换后j号棋子的颜色，

每一次变换的过程满足

**三、模型求解**

利用Matlab、Python、C/C++、Java等编程语言均可以有效求解该模型。在这里我选择利用Python进行求解，其编写较为简便，程序代码如下：

import random

def generate(n):

    chess=[]

    for i in range(0,n):

        chess.append(random.choice([True,False]))

    return chess

n=32

for t in range(0,100):

    chess=generate(n)

    for i in range(0,100):

        chessprev=chess.copy()

        temp=chess[0]

        for j in range(0,n-1):

            chess[j]=not chess[j]^chess[j+1]

        chess[n-1]=not chess[n-1]^temp

        if chess==chessprev:

            break

    print(chess)

该程序需要用户指定一个N，重复100次以下过程：利用generate函数生成一个随机棋子排列，对该排列进行逐个棋子同或的过程，直到排列不再变化或已重复了100次。输出最终得到的排列。

在N取不同值的情况下运行该程序，不难发现N取1，2，4，8，16，…时，程序输出的结果均为全黑棋排列，如下图所示：

对程序进行适当更改，使其输出100个任意序列得到的序列均相同的N，得：

即N=1，2，4，8，16，32，64，128时，可以通过数次变换得到全黑棋排列。

**四、结果分析**

模型求解得到的结果表明，当N=2n（n=0,1,2,…）时，经过数次重复后，各棋子均变为黑色棋子。当然，由于此模型的求解仅针对了N在200以内、100个生成的随机序列和500次变换，其得出的结论并不是严谨可证的，而是利用大量数据的模拟得到的。

事实上，可以通过同或的特性来证明这一结论。对于某一能变换为全黑色的序列，我们考察一颗棋子颜色变化的情况。当棋子发生变化而其他棋子不变，由同或运算可知和会发生变化。一直变化下去，我们会发现2n-1次变换后，相关联的2n颗棋子颜色恰巧全部发生变化。这时，如果N恰巧等于2n，我们就得到了一个全为白色的序列，其在下次变换之后变为全黑色。而我们知道，当初始棋子全为黑色时，其无论如何变换均为黑色，而每个初始排列都可以在此基础上经过数次调整得到，故N=2n时，任意排列都可以在经过2n次变换后变为全黑色。

**五、模型评价**

该数学模型将棋子之间的变换抽象为布尔代数中的同或运算，这样就可以直接用编程语言中提供的运算符进行求解。当然，也可以将黑白棋分别抽象为1和-1，由于1\*1=1，（-1）\*（-1）=1，1\*（-1）=-1，（-1）\*1=-1，可以利用乘法代替棋子之间的变换，也是一种行之有效的方法。

**结论：**

经过有效的模型建立和求解，我们得出，当N=2n（n=0,1,2,…）时，经过至多2n次重复后，各棋子均变为黑色棋子。而N取其他值时，棋子颜色的变化不能找到普遍规律。

**参考文献：**

[1]数字逻辑（第3版）.鲍家元等[M]高等教育出版社,2011.6

[2]离散数学（第3版）.陈建明等[M]西安交通大学出版社,2012.1

**附：同或运算和异或运算**

同或运算的逻辑功能是：两输入值相同时输出为1，反之输出为0；异或运算的逻辑功能是：两输入值不同时输出为0，反之输出为1；同或和异或恰为反函数关系。Python中^运算符表示异或运算，not运算符表示取反，我们可以用not a^b来计算同或运算。

**附：更改后源代码**

import random

def generate(n):

    chess=[]

    for i in range(0,n):

        chess.append(random.choice([True,False]))

    return chess

for n in range(1,200):

    cheeses=[]

    for t in range(0,100):

        chess=generate(n)

        for i in range(0,500):

            chessprev=chess.copy()

            temp=chess[0]

            for j in range(0,n-1):

                chess[j]=not chess[j]^chess[j+1]

            chess[n-1]=not chess[n-1]^temp

            if chess==chessprev:

                break

        cheeses.append(chess)

    temp=0

    for t in range(0,99):

        if cheeses[t]!=cheeses[t+1]:

            temp=1

            break

    if temp==0:

        print(n)

        print(cheeses[0])