



北京航空航天大学
BEIHANG UNIVERSITY

数学建模（一）

——分羊问题模型

学院：可靠性与系统工程学院

姓名：曹建钦

学号：20375177

分羊问题

摘要

本论文分析了分羊问题的其他两种情况——农民有四个儿子、邻居牵来两只羊，并提供了相应算法得出不定方程的全部可能结果。

问题一中，得到了农民有四个儿子，邻居牵来一只羊时分羊问题故事的全部讲法。

问题二中，得到了农民有三个儿子，邻居牵来两只羊时分羊问题故事的全部讲法。

关键词：分羊问题，不定方程

一、问题重述

问题一：农民有 n 只羊，大儿子分 $1/x$ ，二儿子分 $1/y$ ，三儿子分 $1/z$ ，小儿子分 $1/u$ ，如果邻居牵来 1 只羊参与分羊，则四个儿子能成功分 n 只羊，求出 x, y, z, u, n 的所有可能解。

问题二：农民有 n 只羊，大儿子分 $1/x$ ，二儿子分 $1/y$ ，小儿子分 $1/z$ ，如果邻居牵来 2 只羊参与分羊，则三个儿子能成功分 n 只羊，求出 x, y, z, n 的所有可能解。

二、问题分析

2.1 问题一的分析

想要得到所有可能得结果需要列方程求解。

题目给出条件如下：

一、农民有 n 只羊， n 为未知正整数

二、农民要求大儿子分 $1/x$ ，二儿子分 $1/y$ ，三儿子分 $1/z$ ，小儿子分 $1/u$ 。则 x, y, z, u 为 4 个未知的正整数，在这 4 个正整数中，因为 $1 > 1/x > 1/y > 1/z > 1/u$ ，所以 $1 < x < y < z < u$

三、再牵来 1 只羊后，羊就能够分配了，说明 x, y, z, u 都能整除 $n+1$

四、4 个儿子分过以后还剩 1 只羊

根据这些条件，可以找到等量关系列出方程。大儿子分得 $(n+1)/x$ 只，二儿子分得 $(n+1)/y$ 只，三儿子分得 $(n+1)/z$ 只，小儿子分得 $(n+1)/u$ 只，四个儿子共分得 n 只羊，则列出如下方程：

$$\frac{n+1}{x} + \frac{n+1}{y} + \frac{n+1}{z} + \frac{n+1}{u} = n$$

令 $w = n + 1$ ，则方程可化为：

$$\frac{1}{x} + \frac{1}{y} + \frac{1}{z} + \frac{1}{u} + \frac{1}{w} = 1$$

这里， x, y, z, u, w 都必须为正整数，而且还必须满足两个条件：

$$\textcircled{1} \quad x < y < z < u \leq w$$

$$\textcircled{2} \quad x, y, z, u \text{ 都能整除 } w$$

此为不定方程，有五个未知数，两个约束条件，方程不只有一个解。

2.2 问题二的分析

想要得到所有可能得结果需要列方程求解。

题目给出条件如下：

一、农民有 n 只羊， n 为未知正整数

二、农民要求大儿子分 $1/x$ ，二儿子分 $1/y$ ，小儿子分 $1/z$ 。则 x, y, z 为 3 个未知的正整数，在这 3 个正整数中，因为 $1 > 1/x > 1/y > 1/z$ ，所以 $1 < x < y < z$

三、再牵来 2 只羊后，羊就能够分配了，说明 x, y, z 都能整除 $n+2$

四、4 个儿子分过以后还剩 2 只羊

根据这些条件，可以找到等量关系列出方程。大儿子分得 $(n+2)/x$ 只，二儿子分得 $(n+2)/y$ 只，小儿子分得 $(n+2)/z$ 只，三个儿子共分得 n 只羊，则列出如下方程：

$$\frac{n+2}{x} + \frac{n+2}{y} + \frac{n+2}{z} = n$$

令 $w = n + 2$ ，则方程可化为：

$$\frac{1}{x} + \frac{1}{y} + \frac{1}{z} + \frac{2}{w} = 1$$

这里， x, y, z, w 都必须为正整数，而且还必须满足两个条件：

$$\textcircled{1} \quad x < y < z \leq w$$

$$\textcircled{2} \quad x, y, z \text{ 都能整除 } w$$

此为不定方程，有四个未知数，两个约束条件，方程不只有一个解。

三、符号说明

问题一和问题二的符号说明如表格 1、表格 2 所示

表格 1：问题一符号说明

| 符号 | 说明 |
|-----|------------|
| x | 大儿子分 $1/x$ |
| y | 二儿子分 $1/y$ |
| z | 三儿子分 $1/z$ |

| | |
|---|-----------|
| u | 小儿分 1/u |
| n | 农民拥有 n 只羊 |
| w | n+1 |

表格 2：问题二符号说明

| 符号 | 说明 |
|----|-----------|
| x | 大儿子分 1/x |
| y | 二儿子分 1/y |
| z | 小儿分 1/z |
| n | 农民拥有 n 只羊 |
| w | n+2 |

四、模型的建立与求解

4.1 问题一模型及求解

4.1.1 模型

$$\frac{1}{x} + \frac{1}{y} + \frac{1}{z} + \frac{1}{u} + \frac{1}{w} = 1$$

满足：

$$x < y < z < u \leq w$$

x, y, z, u 都能整除 w

求解 x, y, z, w。

4.1.2 模型的求解

解不定方程可以先根据一部分条件，选出符合要求的，然后再根据其他条件，淘汰不符合要求的，留下符合要求的，最后就可以把 x, y, z, u, w 全部求出来。

1、首先根据 x, y, z, u, w 都是正整数且有 $x < y < z < u \leq w$ ，可以得到：

$$\begin{cases} x > 1 \\ \frac{5}{x} > 1 \end{cases}$$

则 x 只能取 2,3,4

2、分别讨论当 $x = 2$, $x = 3$, $x = 4$ ，时：

$$\begin{cases} y > x \\ y > \frac{1}{1 - \frac{1}{x}} \\ y < \frac{4}{1 - \frac{1}{x}} \end{cases}$$

对于每个可能的 y （如 $x = 2$ 时， $2 < y < 8$ ，则 y 分别取 3,4,5,6,7）进行讨论 z 的取值范围：

$$\begin{cases} z > y \\ z > \frac{1}{1 - \frac{1}{x} - \frac{1}{y}} \\ z < \frac{3}{1 - \frac{1}{x} - \frac{1}{y}} \end{cases}$$

对于每个可能的 z （如 $x = 2$ ， $y = 3$ 时， $6 < z < 18$ ，则 z 分别取 7~17 的整数）进行讨论 u 的取值范围：

$$\begin{cases} u > z \\ u > \frac{1}{1 - \frac{1}{x} - \frac{1}{y} - \frac{1}{z}} \\ u \leq \frac{2}{1 - \frac{1}{x} - \frac{1}{y} - \frac{1}{z}} \end{cases}$$

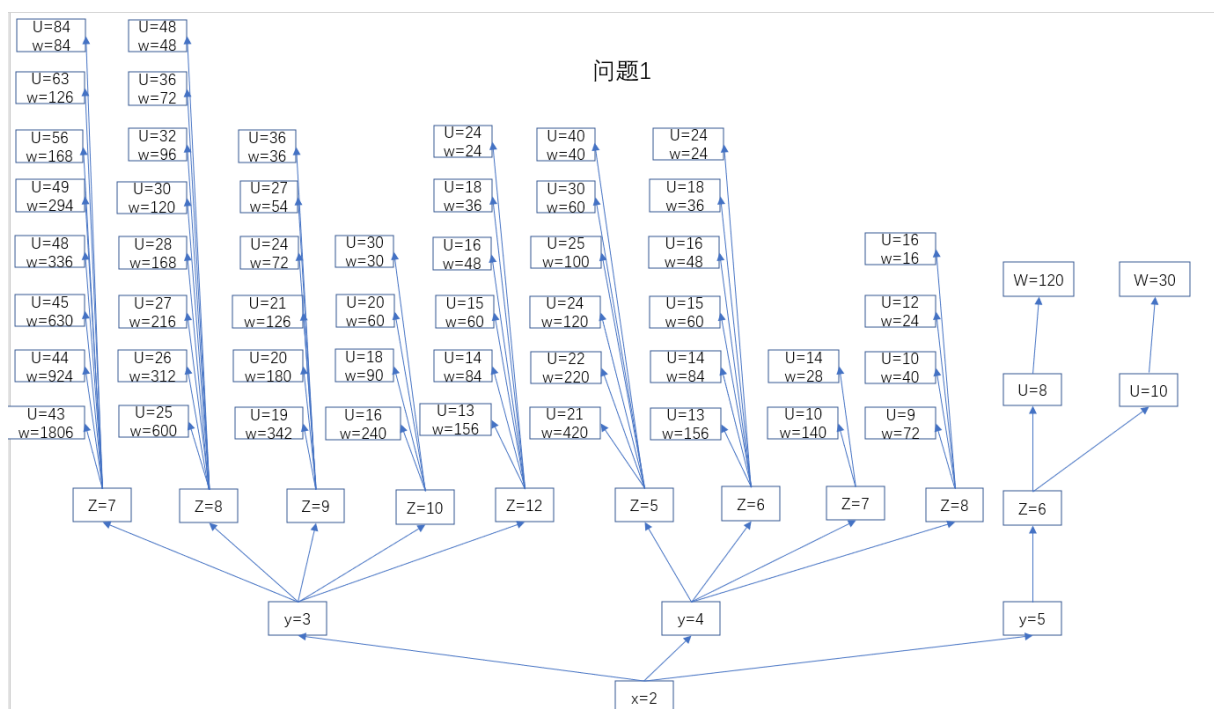
对于每个可能的 u （如 $x = 2$ ， $y = 3$ ， $z = 7$ 时 $42 < u \leq 84$ ，则 u 分别取 43~84 之间的整数）：

可由 $\frac{1}{x} + \frac{1}{y} + \frac{1}{z} + \frac{1}{u} + \frac{1}{w} = 1$ 计算出对应的 w （如 $x = 2$ ， $y = 3$ ， $z = 7$ ， $u = 43$ 时， $w = 1806$ ）

将按照这个步骤可以得到的数据依次进行检验，选出每个除以 x ， y ， z ， u 得到整数的 w 。

由于数据量太大，这里利用 Visual Studio 编写了以循环-判断结构为主体的程序，代码逻辑和上面分析讨论的逻辑相似（具体代码和运行结果见附录一）。

采用树结构描述结果如下：



此方程有 52 组解，故这个故事一共有 52 种讲法：

(x, y, z, u, n) 可以等于 (2, 3, 7, 43, 1805), (2, 3, 7, 44, 923), (2, 3, 7, 45, 629), (2, 3, 7, 48, 335), (2, 3, 7, 49, 293), (2, 3, 7, 56, 167), (2, 3, 7, 63, 125), (2, 3, 7, 84, 83), (2, 3, 8, 25, 599), (2, 3, 8, 26, 311), (2, 3, 8, 27, 215), (2, 3, 8, 28, 167), (2, 3, 8, 30, 119), (2, 3, 8, 32, 95), (2, 3, 8, 36, 71), (2, 3, 8, 48, 47), (2, 3, 9, 19, 341), (2, 3, 9, 20, 179), (2, 3, 9, 21, 125), (2, 3, 9, 24, 71), (2, 3, 9, 27, 53), (2, 3, 9, 36, 35), (2, 3, 10, 16, 239), (2, 3, 10, 18, 89), (2, 3, 10, 20, 59), (2, 3, 10, 30, 29), (2, 3, 12, 13, 155), (2, 3, 12, 14, 83), (2, 3, 12, 15, 59), (2, 3, 12, 16, 47), (2, 3, 12, 18, 35), (2, 3, 12, 24, 23), (2, 4, 5, 21, 419), (2, 4, 5, 22, 219), (2, 4, 5, 24, 119), (2, 4, 5, 25, 99), (2, 4, 5, 30, 59), (2, 4, 5, 40, 39), (2, 4, 6, 13, 155), (2, 4, 6, 14, 83), (2, 4, 6, 15, 59), (2, 4, 6, 16, 47), (2, 4, 6, 18, 35), (2, 4, 6, 24, 23), (2, 4, 7, 10, 139), (2, 4, 7, 14, 27), (2, 4, 8, 9, 71), (2, 4, 8, 10, 39), (2, 4, 8, 12, 23), (2, 4, 8, 16, 15), (2, 5, 6, 8, 119), (2, 5, 6, 10, 29)

4.2 问题二模型及求解

4.2.1 模型

$$\frac{1}{x} + \frac{1}{y} + \frac{1}{z} + \frac{2}{w} = 1$$

满足：

$$x < y < z \leq w$$

x, y, z 都能整除 w

求解 x, y, z, w 。

4.2.2 模型的求解

同样为不定方程的求解问题，求解步骤与问题一类似：

1、首先根据 x, y, z, w 都是正整数且有 $x < y < z \leq w$ ，可以得到：

$$\begin{cases} x > 1 \\ \frac{5}{x} > 1 \end{cases}$$

则 x 只能取 2,3,4

2、分别讨论当 $x = 2, x = 3, x = 4$ 时：

$$\begin{cases} y > x \\ y > \frac{1}{1 - \frac{1}{x}} \\ y < \frac{4}{1 - \frac{1}{x}} \end{cases}$$

对于每个可能的 y （如 $x = 2$ 时， $2 < y < 8$ ，则 y 分别取 3,4,5,6,7）进行讨论 z 的取值范围：

$$\begin{cases} z > y \\ z > \frac{1}{1 - \frac{1}{x} - \frac{1}{y}} \\ z \leq \frac{3}{1 - \frac{1}{x} - \frac{1}{y}} \end{cases}$$

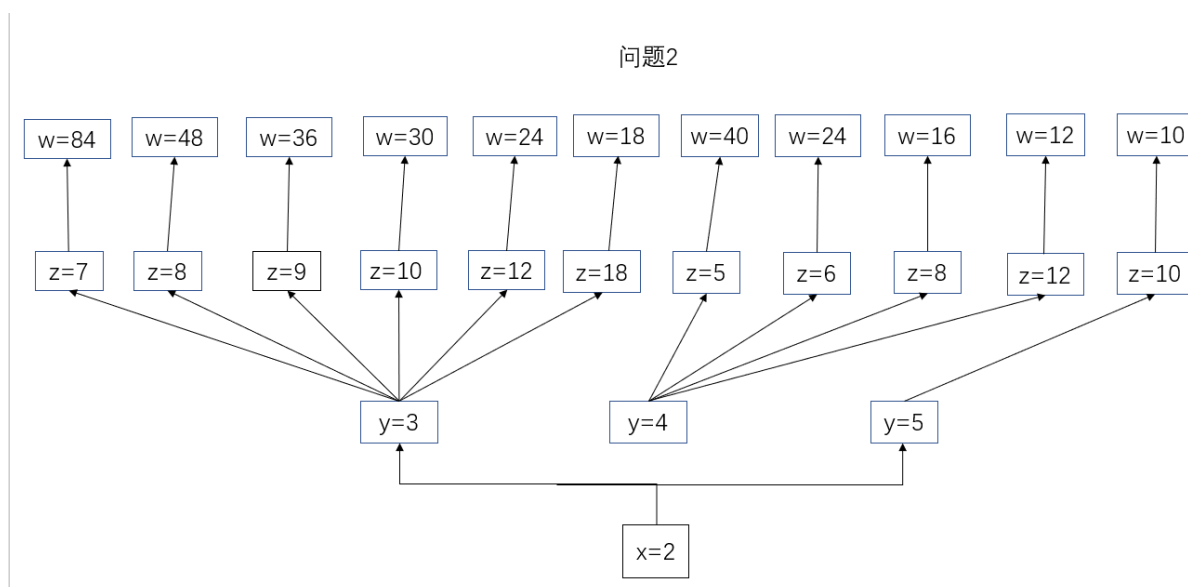
对于每个可能的 z （如 $x = 2, y = 3$ 时， $6 < z \leq 18$ ，则 z 分别取 7~18 的整数）：

可由 $\frac{1}{x} + \frac{1}{y} + \frac{1}{z} + \frac{2}{w} = 1$ 计算出对应的 w （如 $x = 2, y = 3, z = 7$ 时， $w = 84$ ）

将按照这个步骤可以得到的数据依次进行检验，选出每个除以 x, y, z 得到整数的 w 。

这里同样利用 Visual Studio 编写了以循环-判断结构为主体的程序，代码逻辑和上面分析讨论的逻辑相似（具体代码和运行结果见附录二）。

采用树结构描述结果如下：



此方程一共有 11 组，故这个故事有 11 种讲法：

(x, y, z, n) 可以为：(2, 3, 7, 82)，(2, 3, 8, 46)，(2, 3, 9, 34)，(2, 3, 10, 28)，(2, 3, 12, 22)，(2, 3, 18, 16)，(2, 4, 5, 38)，(2, 4, 6, 22)，(2, 4, 8, 14)，(2, 4, 12, 10)，(2, 5, 10, 8)

五、模型的评价、改进与推广

5.1 模型的优点

该模型为分羊问题建立了数学模型——不定方程的求解，并提供了对应代码对求解步骤进行自动化。

5.2 模型的缺点

可以看出问题一和问题二代码重复性很高，同时，有几个儿子意味着需要几次循环，但循环次数过多会导致算法复杂度指数级增长，严重影响运行速度。

5.3 模型的改进

可以将问题一、问题二的解题代码整合，对外只提供儿子数量和邻居牵来几只羊的接口，从而减少重复性，提高算法的适用性。

不定方程形式类似于埃及分数求和，现有迭代深搜的算法可以借鉴以优化求解过程。

5.4 模型的推广

这两个问题分别改动了最原始的分羊问题：“从前有个农民，他有 17 只羊。他要把羊分给 3 个儿子。他说：大儿子分一半，二儿子分 $\frac{1}{3}$ ，小儿子分 $\frac{1}{9}$ ，但不允许把羊杀死或者卖掉。怎么分？”中问题设计中的变量——农民儿子的数量和分羊问题答案：邻

居牵来几只羊。

进一步地，可以设定任意数量儿子和牵来几只羊参与分羊作为问题的答案，并通过更改算法中的循环层数和参数得到相应的分羊比例。这样分羊问题就可以彻底解决。

同时，此模型也为类似的不定方程求解问题提供了一种通用的分析思路和解法。

附录一（问题一算法及结果）

```
#include<iostream>
using namespace std;
double max(double a, double b)
{
    if (a > b)
    {
        return a;
    }
    else
    {
        return b;
    }
}
//向下取整
double quzheng(double a)
{
    if (a == int(a)) return int(a);
    else if (a<int(a + 1) && a>int(a + 1) - 1e-8) return int(a + 1);
    else if (a > int(a) && a < int(a) + 1e-8) return int(a);
    else return int(a);
}
int doubleisint(double a, double b)
{
    if (fabs(a / b - (int)(a / b)) < 1e-8)
        return 1;
    else if (fabs(a / b - 1 - (int)(a / b)) < 1e-8)
        return 1;
    else if (fabs(a / b + 1 - (int)(a / b)) < 1e-8)
        return 1;
    else
        return 0;
}
int main()
{
    double x, y, z, u, w;
    cout << "问题一解法" << endl;
    for (x = 2; x < 5; x++)
    {
        double y1 = 1 / (1 - 1 / x);
        double y2 = 4 / (1 - 1 / x);
        for (y = max(quzheng(y1) + 1, x + 1); y < quzheng(y2); y++)
        {
            double z1 = 1 / (1 - 1 / x - 1 / y);
            double z2 = 3 / (1 - 1 / x - 1 / y);
            for (z = max(quzheng(z1) + 1, y + 1); z < quzheng(z2); z++)
            {
                double u1 = 1 / (1 - 1 / x - 1 / y - 1 / z);
                double u2 = 2 / (1 - 1 / x - 1 / y - 1 / z);
                for (u = max(quzheng(u1) + 1, z + 1); u <= quzheng(u2); u++)
                {
```

```

        w = 1 / (1 - 1 / x - 1 / y - 1 / z - 1 / u);
        if (doubleisint(w, x) && doubleisint(w, y) && doubleisint(w,
z) && doubleisint(w, u))
        {
            cout << "(" << x << " " << y << " " << z << " " << u << "
" << w << ")" << endl;
        }
    }
}
}
system("pause");
return 0;
}

```

输出结果:

问题一解法

```

(2 3 7 43 1806) (2 3 7 44 924) (2 3 7 45 630) (2 3 7 48 336) (2 3 7 49 294) (2 3
7 56 168) (2 3 7 63 126)
(2 3 7 84 84) (2 3 8 25 600) (2 3 8 26 312) (2 3 8 27 216) (2 3 8 28 168) (2 3 8
30 120) (2 3 8 32 96)
(2 3 8 36 72) (2 3 8 48 48) (2 3 9 19 342) (2 3 9 20 180) (2 3 9 21 126) (2 3 9
24 72) (2 3 9 27 54)
(2 3 9 36 36) (2 3 10 16 240) (2 3 10 18 90) (2 3 10 20 60) (2 3 10 30 30) (2 3
12 13 156) (2 3 12 14 84)
(2 3 12 15 60) (2 3 12 16 48) (2 3 12 18 36) (2 3 12 24 24) (2 4 5 21 420) (2 4
5 22 220) (2 4 5 24 120)
(2 4 5 25 100) (2 4 5 30 60) (2 4 5 40 40) (2 4 6 13 156) (2 4 6 14 84) (2 4 6
15 60) (2 4 6 16 48)
(2 4 6 18 36) (2 4 6 24 24) (2 4 7 10 140) (2 4 7 14 28) (2 4 8 9 72) (2 4 8 10
40) (2 4 8 12 24)
(2 4 8 16 16) (2 5 6 8 120) (2 5 6 10 30)

```

附录二（问题二算法及结果）

```

#include<iostream>
using namespace std;
double max(double a, double b)
{
    if (a > b)
    {
        return a;
    }
    else
    {
        return b;
    }
}
//向下取整
double quzheng(double a)
{
    if (a == int(a)) return int(a);
    else if (a<int(a + 1) && a>int(a + 1) - 1e-8) return int(a + 1);
}

```

```

        else if (a > int(a) && a < int(a) + 1e-8) return int(a);
        else return int(a);
    }
    int doubleisint(double a, double b)
    {
        if (fabs(a / b - (int)(a / b)) < 1e-8)
            return 1;
        else if (fabs(a / b - 1 - (int)(a / b)) < 1e-8)
            return 1;
        else if (fabs(a / b + 1 - (int)(a / b)) < 1e-8)
            return 1;
        else
            return 0;
    }
    int main()
    {
        double x, y, z, w;
        cout << "问题二解法" << endl;
        for (x = 2; x < 5; x++)
        {
            double y1 = 1 / (1 - 1 / x);
            double y2 = 4 / (1 - 1 / x);
            for (y = max(quzheng(y1) + 1, x + 1); y < quzheng(y2); y++)
            {
                double z1 = 1 / (1 - 1 / x - 1 / y);
                double z2 = 3 / (1 - 1 / x - 1 / y);
                for (z = max(quzheng(z1) + 1, y + 1); z <= quzheng(z2); z++)
                {
                    w = 2 / (1 - 1 / x - 1 / y - 1 / z);
                    if (doubleisint(w, x) && doubleisint(w, y) && doubleisint(w, z))
                    {
                        cout << "(" << x << " " << y << " " << z << " " << w << ")"
<< endl;
                    }
                }
            }
        }
        system("pause");
        return 0;
    }

```

输出结果:

```

问题二解法
(2 3 7 84)
(2 3 8 48)
(2 3 9 36)
(2 3 10 30)
(2 3 12 24)
(2 3 18 18)
(2 4 5 40)
(2 4 6 24)
(2 4 8 16)
(2 4 12 12)
(2 5 10 10)

```