      
**数学模型课程**



**第四次作业**

|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 | 序号 |
| 陈宇 |  |  |
| 刘柏杉 |  |  |
| 孙莹莹 | 20204104 |  |

**问题一**

**一．问题重述**

商业银行的整存整取年利率：

|  |  |  |  |
| --- | --- | --- | --- |
| 一年期 | 二年期 | 三年期 | 五年期 |
| 1.750 | 2.250 | 2.750 | 2.750 |

现有1笔本金，准备30年后使用，若此期间利率不变，问应该采用怎样的存款方案？

题目+模型+程序+结果+分析

1. **问题假设**

1.本金为a，为1000元。

2.在这30年内，风险不会改变。

3.在这30年内，各种存款年利率不发生变化。

**三．符号规定**

1. Xij  表示第i年存入j年期的金额

2. a,b,c,d 分别表示一，二，三，五年期的年利率

1. Fi 表示第i年可用于分配存款的本金

**四**．**模型建立**

1. 目标函数

根据题目要求，即第30年所拥有的可用于存款的本金最大，即有



1. 约束条件

根据题目要求，每一年所存款的金额不能超过该年可用于支配的金额，同时存入的金额必须大于等于零。



3. 模型简化

1. 约束条件本为，但由于要使总收益达到最高，每一年都应该将所有可支配本金存储到银行，因此约束条件可以简化为



1. 考虑到最后四年不会投资五年期，最后两年不会投资三年期，最后一年不会投资两年期，可能会减少几个变量，但是由于如果减去这七个变量，并不会对于整个式子的规模减少很多，反而会使得A矩阵生成代码变得十分复杂，最终决定不减少七个变量，并且优化结果也不会有很大问题

**五．模型求解**

1. 不等式约束矩阵A矩阵生成代码

```c++

#include<iostream>

#include<vector>

#include<math.h>

#include<stdio.h>

using namespace std;

#define M 31

#define N 121

double A[M][N];

int table[5] = { 0,1,2,3,5 };

int b[31];

double rate[5] = { 0,1.750, 2.250, 2.750, 2.750 };

int main()

{

for (int i = 0; i < M; i++)

{

for (int j = 0; j < N; j++)

{

A[M][N] = 0;

}

}

for (int i = 0; i < M; i++)

{

b[i] = 0;

}

A[1][1] = 1; A[1][2] = 1; A[1][3] = 1; A[1][4] = 1;

b[1] = 1000;

//开始迭代构造A矩阵

for (int i = 1; i < M; i++)

{

for (int j = 1; j <= 4; j++)

{

A[i][(i - 1) \* 4 + j] = 1;

}

for (int j = 1; j <= 4; j++)

{

if (i - table[j] > 0)

{

A[i][(i - (table[j] + 1)) \* 4 + j] = -(1 + rate[j] \* table[j]\*0.01) \* A[i - table[j]][(i - (table[j]+1)) \* 4 + j];//

}

}

}

for (int i = 1; i <M; i++)

{

//cout << i << endl;

for (int j = 1; j < N; j++)

{

printf\_s("%10f ", A[i][j]);

}

cout << ";"<<endl;

}

}

```

**六．结果及分析**

**问题二**

**一．问题重述**

A公司面临破产，只余下100种物品，表1中给出了每种物品的数量，现有1000名公司债权人，表格中给出了债权人对不同物品的偏好（数值越大越喜欢），要求你们对这些资产进行处置，应该如何安排呢？

1. **模型假设**

1. 通过计算使得已有的多件商品使得所有商品让所有的债权人的喜好得到最优满足

2. 每一种商品每个债权人仅可以拿到一个

3. 每个债权人最多只能领取商品数为：所有商品数/债权人取整

**三．符号规定**

1. bj 第j个商品的数量

2. aij 第i个债权公司对第j个商品的喜好

3. xij 第i个债权公司是否收取第j个商品的决策变量

**四．模型建立**

1.目标函数

根据题目要求，我们的目标是最大程度地满足所有债权公司的喜好。因此目标函数为



1. 约束条件

约束条件首先是对于物品k,100个公司所拿k的总量要小于bk,即



其次，还需满足每个债权人最多只能领取商品数为：所有商品数/债权人取整，即



最后，因为我们将此题看作零一规划，所有xij的取值只能是0或1。综上，约束条件如下：



3.模型简化

1. 欲要使得偏好和最高两个约束条件可以直接使用等于号节约资源进行优化

**五．模型求解**

1.A约束矩阵生成代码

```c++

#include<iostream>

#include<math.h>

using namespace std;

#define M 1000

#define N 100

#define sum 1000

int A[N+M + 1][(M)\*(N)+1];

int b[2 \* M + 1];

int B[N + 1] = { 1,1,1,1,1,1 };

int main()

{

//每一经营商最多收购sum/10个

for (int i = 1; i < M + 1;i++)

{

for (int j = 1; j < N+1; j++)

{

A[i][(i - 1) \* N + j] = 1;

}

}

for (int i = 0; i < M + 1; i++)

{

b[i] = sum / 10;

}

//每一类商品数最多有B[i]个

for (int i = M + 1; i < N + M + 1; i++)

{

for (int j = 1; j < (M)+1; j++)

{

A[i][i-M+(j-1)\*N] = 1;

}

}

for (int i = M + 1; i < N + M + 1;i++)

{

// b[i] = B[i-M];

}

for (int i = 1; i < N + M + 1; i++)

{

for (int j = 1; j < (M) \* (N)+1; j++)

{

cout<<A[i][j] ;

}

cout << ";" << endl;

}

}

```

A矩阵

```c++

```

**六．结果及分析**