**多维背包问题——纯代码版**

#include <stdio.h>

int n,m;//物品数量n，物品属性个数m

int i,j,s,d;

double c[100];//各个属性的约束

double v[100];//各个物品的价值　value

double w[100][100];//各个物品的重量　weight

double cw[100];//当前背包重量　current weight

double cp = 0.0;//当前背包中物品总价值　current value

double bestp = 0.0;//当前最大价值best price

//回溯函数

void backtrack(int x)

{ //i用来指示到达的层数（第几步，从0开始），同时也指示当前选择完了几个物品

double bound(int x);

if(x>n) //递归结束的判定条件

{

bestp = cp;

return;

}

//如若左子节点可行，则直接搜索左子树;

//对于右子树，先计算上界函数，以判断是否将其减去

for(i=1;i<=n;i++){

for(j=1;j<=m;j++){

if(cw[s]+w[i][j]<=c[d])//将物品i放入背包,搜索左子树

{

cw[s]+=w[i][j];//同步更新当前背包的重量

cp+=v[i];//同步更新当前背包的总价值

d++;

}

}

backtrack(x+1);//深度搜索进入下一层

cw[s]-=w[i][j];//回溯复原

cp-=v[i];//回溯复原

}

if(bound(x+1)>bestp)//如若符合条件则搜索右子树

backtrack(x+1);

}

//计算上界函数，功能为剪枝

double bound(int x)

{ //判断当前背包的总价值cp＋剩余容量可容纳的最大价值<=当前最优价值

for(d=1;d<=m;d++){

s+=1;

double leftw= c[d]-cw[s];//剩余背包容量

double b = cp;//记录当前背包的总价值cp,最后求上界

//以物品单位重量价值递减次序装入物品

while(x<=n && w[i][j]<=leftw)

{

leftw-=w[i][j];

b+=v[i];

x++;

}//装满背包

if(x<=n)

b+=v[i]/w[i][j]\*leftw;

return b;//返回计算出的上界

}

}

int main(){

int x;

int a[100][100];

printf("请输入物品数和约束数的值：\n");

scanf("%d%d",&n,&m);

printf("请输入每个物品的价值:\n");

for(i=1;i<=n;i++){

scanf("%d",&v[i]);

}

printf("请输入约束条件的属性值：\n");

for(i=1;i<=m;i++){

for(j=1;j<=n;j++)

scanf("%d",&a[i][j]);

}

for(i=1;i<=m;i++){

for(j=1;j<=n;j++);

}

printf("请输入每个约束条件的约束值：\n");

for(d=1;d<=n;d++){

scanf("%d",&c[d]);

}

backtrack(1);

printf("最大价值为：%lf\n",bestp);

return;

}