遗传算法解决背包问题：

//评价函数就是物品的价值之和，但要注意一旦物品的重量大于背包的重量，那么该条染色体的幸存概率为0

//基因就是每一个物品是否选择，这里默认有10条染色体在比较，并且每一条染色体上的第i个基因就是代表第i个物品是否选择

//突变就是随机选择的染色体的随机位置由0变1,由1变0

//另外建议把变异的概率调高点，否则有可能会陷入局部最优

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

#include <time.h>

#define num 10//种群的大小

#define pm 0.1//变异的概率

#define pc 0.8//交配的概率

#define MAX 100//最大迭代次数

int n,m;//n为物品种类，m为背包容积大小

int v[10];//每一个物品的价值

int w[10];//每一个物品的重量

typedef struct node

{

int beibao[10];//描述每一个物品的状态

int adapt;//表示这种装入方法的适应值

double p;//幸存的概率

}node;

node group[num],grouptemp[num];

void groupproduce()

{

int i,j,t,wsum;

srand(time(NULL));

for (i=0;i<num;i++)

{

wsum=0;

for (j=0;j<n;j++)

{

t=rand()%100;

if (t<=49)

group[i].beibao[j]=0;

else

group[i].beibao[j]=1;

wsum=wsum+group[i].beibao[j]\*w[j];

}

if (wsum>m)

i--;

}

printf("初始状态为:\n");

for (i=0;i<num;i++)

{

for (j=0;j<n;j++)

printf("%d",group[i].beibao[j]);

printf("\n");

}

}

void pingjia()

{

int i,j,vsum,wsum,sum;

sum=0;

for (i=0;i<num;i++)

{

vsum=0;//物品的总价值

wsum=0;//物品的总容积（只要超过m一律认为适应性为0)

for (j=0;j<n;j++)

{

vsum=vsum+group[i].beibao[j]\*v[j];

wsum=wsum+group[i].beibao[j]\*w[j];

}

if (wsum>m)

group[i].adapt=0;

else

group[i].adapt=vsum;

sum=sum+group[i].adapt;

}

for (i=0;i<num;i++)

group[i].p=group[i].adapt\*1.0/sum;//计算幸存概率

for (i=0;i<num;i++)

printf("第%d条染色体价值为%d,幸存概率为%.4f\n",i,group[i].adapt,group[i].p);

}

void xuanze()

{

int i,j,temp;

double gradient[num];

double t;

int xuan[num];

for (i=0;i<num;i++)

gradient[i]=0.0;

gradient[0]=group[0].p;

for (i=1;i<num;i++)

gradient[i]=gradient[i-1]+group[i].p;

srand(time(NULL));

for (i=0;i<num;i++)

{

t=rand()%100;

t=t/100;

for (j=0;j<num;j++)

if (gradient[j]>t)

{

xuan[i]=j;

break;

}

}

for (i=0;i<num;i++)

{

for (j=0;j<n;j++)

grouptemp[i].beibao[j]=group[i].beibao[j];

grouptemp[i].adapt=group[i].adapt;

grouptemp[i].p=group[i].p;

}

for (i=0;i<num;i++)

{

temp=xuan[i];

group[i].adapt=grouptemp[temp].adapt;

group[i].p=grouptemp[temp].p;

for (j=0;j<n;j++)

group[i].beibao[j]=grouptemp[temp].beibao[j];

}

}

void jiaopei()

{

int i,j,t,c,d;

double jiaopeip[num];

int jiaopeiflag[num],temp,temp1,temp2,duan;

srand(time(NULL));

t=0;

for (i=0;i<num;i++)

{

jiaopeip[i]=rand()%100;

jiaopeip[i]=jiaopeip[i]/100;

if (jiaopeip[i]<pc)

jiaopeiflag[t]=i;

t++;

}

t=t/2\*2;

srand(time(NULL));

c=0;

d=1;

printf("%d\n",t);

for (i=0;i<t/2;i++)

{

temp1=jiaopeiflag[c];

temp2=jiaopeiflag[d];

c=c+2;

d=d+2;

duan=rand()%num;//随机断点

for (i=0;i<duan;i++)//断点之前的部分交换基因

{

temp=group[temp1].beibao[i];

group[temp1].beibao[i]=group[temp2].beibao[i];

group[temp2].beibao[i]=temp;

}

}

}

void bianyi()

{

int i,j,t;

double bianyip[num];

int bianyiflag[num];

srand(time(NULL));

memset(bianyiflag,0,sizeof(bianyiflag));

for (i=0;i<num;i++)

{

bianyip[i]=rand()%100;

bianyip[i]=bianyip[i]/100;

if (bianyip[i]<pm)

bianyiflag[i]=1;

}

srand(time(NULL));

for (i=0;i<num;i++)

if (bianyiflag[i]==1)

{

t=rand()%num;//随机生成变异基因

group[i].beibao[t]=1-group[i].beibao[t];//0变为1,1变为0

}

}

void shuaixuan()

{

int i,j;

double max=group[0].p;

for (i=1;i<num;i++)

if (max<group[i].p)

{

max=group[i].p;

j=i;

}

printf("最优选择为：\n");

for (i=0;i<n;i++)

printf("%d ",group[j].beibao[i]);

printf("\n");

printf("最多获利为:%d\n",group[j].adapt);

}

int main()

{

int i,t;

printf("请输入背包种类n\n");

scanf("%d",&n);

printf("请输入每一个背包的重量和价值\n");

for (i=0;i<n;i++)

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

printf("请输入背包容积\n");

scanf("%d",&m);

groupproduce();//生成初始种群

t=0;

while(t<MAX)

{

pingjia();

xuanze();

jiaopei();

bianyi();

t++;

}

pingjia();//最后还要进行一次评价，因为之前交配时更改过beibao（与adapt不对应)

shuaixuan();

return 0;

}

