function PVal = operation1\_1\_1(K) %针对一道工序的情况

%INPUT K IS AN ARRAY WHICH HAS 50 ELEMNTS,WHICH WE WANT

%M THE AMOUNT OF MACHINES N THE AMOUNT OF THINGS T THE ARRAY OF OPTIME

%E THE MATRIAX OF WAYTIME

%TO THE ORIGINNAL NUMBER OF EACH MACHINE

M=8;

N=50;

K=K(51:100);

%T=[572 577 572 577 572 577 572 577];

T=[588 591 588 591 588 591 588 591];

%T=[610 615 610 615 610 615 610 615];

E=[53 53 73 73 86 86 99 99

56 56 76 76 89 89 102 102

73 73 53 53 73 73 86 86

76 76 56 56 76 76 89 89

86 86 73 73 53 53 73 73

89 89 76 76 56 56 76 76

99 99 86 86 73 73 53 53

102 102 89 89 76 76 56 56

];

CO=[27146 28331 27275 27331 27404 27460 27533 28229

];

EO=2;

PVal=zeros(2,N);

C=zeros(1,N);

fmindex=zeros(1,M);

for j=1:M

for i=1:N

if K(i)==j

fmindex(j)=i;

break;

end

end

end

i=1;

temp=[CO(K(i)) E(EO,K(i))];

C(i)=max(temp)+T(K(i));

PVal(2,i)=max(temp);

for i=2:N

if isempty(find(fmindex==i, 1))

for j=i-1:-1:1

if K(j)==K(i)

x=j;

break;

end

end

temp=[C(x) C(i-1)-T(K(i-1))+E(K(i-1),K(i))];

C(i)=max(temp)+T(K(i));

PVal(2,i)=max(temp);

else

temp=[CO(K(i)),C(i-1)-T(K(i-1))+E(K(i-1),K(i))];

C(i)=max(temp)+T(K(i));

PVal(2,i)=max(temp);

end

end

PVal(1,:)=C;

end

function PVal = operation2\_1\_1(K) %针对两道工序的情况

M=8; %机器数

N=40;%工序数

n=K(1:40);

m=K(41:80);

T=[482 487 482 487 482 487 482 487;

209 214 209 214 209 214 209 214];

%T=[428 431 428 431 428 431 428 431

% 406 409 406 409 406 409 406 409];

%T=[310 315 310 315 310 315 310 315

% 530 535 530 535 530 535 530 535];

E=[27 27 45 45 59 59 73 73

32 32 50 50 64 64 78 78

45 45 27 27 45 45 59 59

50 50 32 32 50 50 64 64

59 59 45 45 27 27 45 45

64 64 50 50 32 32 50 50

73 73 59 59 45 45 27 27

78 78 64 64 50 50 32 32

]; %不清洗时间

CO=[99999999 21325 24650 24195 24513 24695 24023 21588

];

EO=6;

C=zeros(1,N);

PVal=zeros(2,N);

fmindex=zeros(1,M);

for j=1:M

for i=1:N

if m(i)==j

fmindex(j)=i;

break;

end

end

end

i=1;

temp=[CO(m(i)) E(EO,m(i))];

C(i)=max(temp)+T(1,m(i));

PVal(2,i)=max(temp);

for i=2:N

for k=i-1:-1:1 %判断当前加工的是第几步

if n(k)==n(i)

nstep=1;

nmachine=k; %保留第一步工序的索引

else

nstep=0;

nmachine=i;

end

end

if isempty(find(fmindex==i, 1)) %判断机器之前有没有用过，先是已经用过的

for j=i-1:-1:1

if m(j)==m(i) %找到上一次用这个机器，判断这个机器是否用完了

x=j;

break;

end

end

if nstep==0 % first step

temp=[C(x) C(i-1)-T(m(i-1))+E(m(i-1),m(i))];

C(i)=max(temp)+T(1,m(i));

PVal(2,i)=max(temp);

else

if m(nmachine)==m(i-1) %在此考虑三种绕路取物品的问题

if nmachine~=i-1

temp=[C(x) C(i-1)-T(m(i-1))+E(m(i-1),m(i))]; %先被安排了另外的任务

else

temp=[C(x) C(i-1)+E(m(i-1),m(i))]; %未被安排另外的任务

end

else

temp=[C(x) C(i-1)-T(m(i-1))+E(m(i-1),m(nmachine))+E(m(nmachine),m(i)) C(nmachine)+E(m(nmachine),m(i))];

end

C(i)=max(temp)+T(2,m(i));

PVal(2,i)=max(temp);

end

else %这是第一次使用该机器

if nstep==0 % first step

temp=[CO(m(i)) C(i-1)-T(m(i-1))+E(m(i-1),m(i))];

C(i)=max(temp)+T(1,m(i));

PVal(2,i)=max(temp);

else

if m(nmachine)==m(i-1) %在此考虑三种绕路取物品的问题

if nmachine~=i-1

temp=[CO(m(i)) C(i-1)-T(m(i-1))+E(m(i-1),m(i))]; %先被安排了另外的任务

else

temp=[CO(m(i)) C(i-1)+E(m(i-1),m(i))]; %未被安排另外的任务

end

else

temp=[CO(m(i)) C(i-1)-T(m(i-1))+E(m(i-1),m(nmachine))+E(m(nmachine),m(i)) C(nmachine)+E(m(nmachine),m(i))];

end

C(i)=max(temp)+T(2,m(i));

PVal(2,i)=max(temp);

end

end

end

PVal(1,:)=C;

End

%主函数

%% 清空环境

clc;clear

%% 下载数据

%load scheduleData Jm T JmNumber

%工序 时间

%% 基本参数

NIND=40; %个体数目

MAXGEN=500; %最大遗传代数

GGAP=0.9; %代沟

XOVR=0.8; %交叉率

MUTR=0.2; %变异率

gen=0; %代计数器

%PNumber 工件个数 MNumber 工序个数

%[PNumber MNumber]=size(Jm);

PNumber=20;MNumber=2;JmNumber=8;

trace=zeros(2, MAXGEN); %寻优结果的初始值

WNumber=PNumber\*MNumber; %工序总个数 80

%% 初始化

Number=zeros(1,PNumber); % PNumber 工件个数

for i=1:PNumber

Number(i)=MNumber; %MNumber工序个数

end

% 代码2层，第一层工序，第二层机器

Chrom=zeros(NIND,2\*WNumber);

for j=1:NIND

WPNumberTemp=Number;

for i=1:WNumber

%随机产成工序

val=unidrnd(PNumber);

while WPNumberTemp(val)==0

val=unidrnd(PNumber);

end

%第一层代码表示工序

Chrom(j,i)= val;

WPNumberTemp(val)=WPNumberTemp(val)-1;

%第2层代码表示机器

%随机产成工序机器

% www = unidrnd(JmNumber);

if mod(i,JmNumber)==0 %通过启发式思想生成顺序安排的机器，几乎一直是最优

www=JmNumber;

else

www=mod(i,JmNumber);

end

Chrom(j,i+WNumber)=www;

end

for i=1:WNumber-1 %杜绝同一机器加工两次

for k=i+1:WNumber

if Chrom(j,i)==Chrom(j,k)

while Chrom(j,i+WNumber)==Chrom(j,k+WNumber)

Chrom(j,k+WNumber)=unidrnd(JmNumber);

end

end

end

end

end

%计算目标函数值

[PVal ObjV P S]=cal(Chrom,PNumber,JmNumber);

%% 循环寻找

while gen<MAXGEN

%分配适应度值

FitnV=ranking(ObjV); %自动排序，给出最适合活下来的染色体

%选择操作

SelCh=select('rws', Chrom, FitnV, GGAP);

%交叉操作

SelCh=across(SelCh,XOVR,PNumber,JmNumber);

%变异操作只变异机器

SelCh=aberranceJm(SelCh,MUTR,PNumber,JmNumber);

%计算目标适应度值

[PVal ObjVSel P S]=cal(SelCh,PNumber,JmNumber);

%重新插入新种群

[Chrom ObjV] =reins(Chrom, SelCh,1, 1, ObjV, ObjVSel);

%代计数器增加

gen=gen+1;

%保存最优值

trace(1, gen)=min(ObjV);

trace(2, gen)=mean(ObjV);

% 记录最佳值

if gen==1

Val1=PVal;

Val2=P;

MinVal=min(ObjV);%最小时间

STemp=S;

end

%记录 最小的工序

if MinVal> trace(1,gen)

Val1=PVal;

Val2=P;

MinVal=trace(1,gen);

STemp=S;

end

end

% 当前最佳值

PVal=Val1; %工序时间

P=Val2; %工序

S=STemp; %调度基因含机器基因

%% 描绘解的变化

figure(1)

plot(trace(1,:));

hold on;

plot(trace(2,:),'-.');grid;

legend('解的变化','种群均值的变化');

%% 显示最优解

figure(2);

MP=S(1,PNumber\*MNumber+1:PNumber\*MNumber\*2);

for i=1:WNumber

val= P(1,i);

a=(mod(val,100)); %工序

b=((val-a)/100); %工件

Temp=[1:JmNumber];

mText=Temp(MP(1,i));

x1=PVal(1,i);

x2=PVal(2,i);

y1=mText-1;

y2=mText;

PlotRec(x1,x2,mText);

PlotRec(PVal(1,i),PVal(2,i),mText);

hold on;

fill([x1,x2,x2,x1],[y1,y1,y2,y2],[1-1/b,1/b,b/PNumber]);

text((x1+x2)/2,mText-0.25,num2str(P(i)),'FontSize',8);

end

clc

PVal

MP

co=zeros(1,JmNumber);

usingtime=[16870 8980 17352 7792 16324 7792 15360 4870

];

for i=1:JmNumber

for j=WNumber:-1:1

if MP(j)==i

co(i)=PVal(1,j); %给出结束时间索引

break;

end

end

end

for i=1:JmNumber

for j=WNumber:-1:1

if MP(j)==i

usingtime(i)=usingtime(i)+PVal(1,j)-PVal(2,j);

end

end

end

co

eo=MP(WNumber)

usingtime

function [PVal ObjV P S]=cal(Chrom,PNumber,JmNumber) %适应度计算函数

% 功能说明： 根据基因群,计算出个群中每个个体的调度工序时间，

% 保存最小时间的调度工序和调度工序时间

% 输入参数：

% Chrom 为基因种群

% PNumber 工件数量

% JmNumber 机器number

% 输出参数:

% PVal 为最佳调度工序时间

% P 为最佳输出的调度工序

% ObjV 为群中每个个体的调度工序时间

% S 为最佳输出的调度基因

%初始化

NIND=size(Chrom,1);

ObjV=zeros(NIND,1);

% 工序个数

MNumber=2;

for i=1:NIND

%取一个个体

S=Chrom(i,:);

%根据基因，计算调度工序

P= calP(S,PNumber);

%根据调度工序，计算出调度工序时间

% PVal=caltime(S,P,JmNumber,T,Jm);

PVal=operation2\_1\_1(S);

%取完成时间

MT=max(PVal);

TVal=max(MT);

%保存时间

ObjV(i,1)=TVal;

%初始化

if i==1

Val1=PVal;

Val2=P;

MinVal=ObjV(i,1);

STemp=S;

end

%记录 最小的调度工序时间、最佳调度工序时间 最佳输出的调度工序

if MinVal> ObjV(i,1)

Val1=PVal;

Val2=P;

MinVal=ObjV(i,1);

STemp=S;

end

end

%最佳调度工序时间 最佳输出的调度工序

PVal=Val1;

P=Val2;

S=STemp;