TOPSIS综合评价法

score=xlsread('zhibiao2.xlsx','C2:J403');

maxsco=max(score);

minsco=min(score);

%归一化

%正向的稳定、供货能力、供货率

%反向相应时间

pos=[4,5,6,7,8];

neg=[1,2,3];%反向列

score(:,pos)=(score(:,pos)-minsco(pos))./(maxsco(pos)-minsco(pos));

score(:,neg)=(max(neg)-score(:,neg))./(maxsco(neg)-minsco(neg));

for i=1:size(score,2)

score(:,i)=score(:,i)./sqrt(sum(score(:,i).\*score(:,i)));%正向化矩阵标准化

end

smax=max(score);

smin=min(score);

for j=1:size(score,1)

distp(j)=sqrt(sum((smax-score(j,:)).^2));%D+

distm(j)=sqrt(sum(score(j,:)-smin).^2);%D-

end

newscore=distm./(distp+distm);

D=[distp' distm'];

[sd,dex50]=sort(newscore,'descend');

ans=[sd' dex50'];

original=xlsread('zhibiao2.xlsx','C2:J403');

final50=original(dex50,:);%提取前50各项指标

自适应滤波

function w=waveadj(y,k,N)%自适应滤波算法

T=length(y);

w=1/N\*ones(1,N);%初始化权重

wlast=ones(1,N);%初始化上一步权重

while sum(abs(w-wlast))>0.01 %当和上一步权重差距仍然较大时，再来一次

%一轮调整权重

for i=N+1:T

yhat=sum(w.\*y(i-N:i-1));%预测值

err=y(i)-yhat;%误差计算

wlast=w;

w=w+2\*k\*err\*y(i-N:i-1);%更新权重，用w'=w+2ke(t+1)y(t-i+1)

w=w/sum(w);%归一化

end

end

end

贯序算法

model:

!the multigoal;

!1st min factory;

!2nd min price;

sets:

amount/1..24/:uA,uB,uC,xA,xB,xC,SA,SB,SC,rdm;

!used amount in each month,the amount to buy in each amount,the amount of store in each month;

buy/1..50/:f,maxp,mid,TA,TB,TC;!the factories,the max production of factories,which type to buy,the rand use to adjust the max;

week/1..24/:w;!which week;

res(buy,week):mb,ans;!the amount to buy in the factory;

goal/1,2/:z,g;!the different goals,the best of z;

tylimit/1,2,3/:limit;!the limit in amount of each type;

soft\_variable/1..24/:dm,dp;!soft conditions;

endsets

data:

seed=123;

rdm=@qrand(seed);!use rand to change the max;

mid=@ole('D:\MCM2021\fact50.xlsx','mid');!the midiam;

TA=@ole('D:\MCM2021\fact50.xlsx','TA');

TB=@ole('D:\MCM2021\fact50.xlsx','TB');

TC=@ole('D:\MCM2021\fact50.xlsx','TC');!the type sell by each factory;

maxp=@ole('D:\MCM2021\fact50.xlsx','MAXP');!the max production of each factory;

limit=23406 42743 17053;!limit of A,B,C;

ctrl=1;!0 is z1,1 is z2;

g=? ?;!the z;

@ole('D:\MCM2021\pro1buyfinal.xlsx','buy')=ans;!the answer;

enddata

!object;

z(1)=@sum(buy:f);!the min of factory amount;

z(2)=1.2\*@sum(amount:xA)+1.1\*@sum(amount:xB)+@sum(amount:xC)+2\*@sum(amount:SA+SB+SC);!the min money to buy and store;

u=@sum(amount:SA+SB+SC);

min=(1-ctrl)\*z(1)+ctrl\*z(2);

!@sum(soft\_variable:dm);

!(1-ctrl)\*z(1)+ctrl\*z(2);

!conditions;

SA(1)=0;

SB(1)=0;

SC(1)=0;!innitial value;

z(1)=36;

@sum(soft\_variable:dm)<5338.147;!limit lack;

@for(amount(i)|i #ne# 1:SA(i)/0.6+SB(i)/0.66+SC(i)/0.72+dm(i)-dp(i)>28200\*2);!we should store for 2 weeks' production;

!for the first day;

xA(1)>uA(1);

xB(1)>uB(1);

xC(1)>uC(1);

@for(amount(i)|i #ne# 1:SA(i)=SA(i-1)+xA(i)-uA(i);SB(i)=SB(i-1)+xB(i)-uB(i);SC(i)=SC(i-1)+xC(i)-uC(i));!the recursive relationship of S;

@for(amount(j):uA(j)/0.6 + uB(j)/0.66 + uC(j)/0.72+dm(j)-dp(j)>28200);!the capability should be produced;

@for(amount(j):xA(j)+xB(j)+xC(j)<6000\*8);!the goods can be translated;

@for(buy:@bin(f));

!conditions;

@for(amount(j):xA(j)=@sum(buy(i):mb(i,j)\*f(i)\*TA(i));xB(j)=@sum(buy(i):mb(i,j)\*f(i)\*TB(i));xC(j)=@sum(buy(i):mb(i,j)\*f(i)\*TC(i)));

!each week the amount from each factory of each type is enough;

@for(res(i,j):mb(i,j)<rdm(j)\*maxp(i)+(1-rdm(j))\*(mid(i)+maxp(i))/2);!each factory has its limit;

@for(res(i,j):ans(i,j)=mb(i,j)\*f(i));

!soft;

!u+dm(1)-dp(1)=779384.8;

@for(goal(i):z(i)<g(i));

End

多阶段优化

model:

sets:

fact/1,2..36/:fbuy;!the amount bought from each company;

trans/1,2..8/:pbrk;!the translation company;

week/1,2..24/:w;!each week;

ans(fact,trans):plan,pwhe,fplan;!how much,whether;

supply(fact,week):s;!the amount in each week bought from each factory;

lost(trans,week):lp;!the lost portion of each translator in each week;

soft\_var/1..9/:dm,dp;!soft variables;

endsets

data:

s=@ole('D:\MCM2021\pro1buyfnl36.xlsx','buy');!the buying plan;

lp=@ole('D:\MCM2021\comp.xlsx','PL');!the loss;

ctrl=?;

!@ole('D:\MCM2021\pro1translation.xlsx','week14')=fplan;

enddata

!object;

min=@sum(soft\_var:dp);

!conditions;

!hard conditions;

@for(fact(i):fbuy(i)=s(i,ctrl));!the amount to buy in this week;

@for(trans(j):pbrk(j)=lp(j,ctrl));!the loss portions in this week;

@for(ans:@bin(pwhe));!whether 0-1;

@for(fact(i):@sum(trans(j):plan(i,j)\*pwhe(i,j))=fbuy(i));!all buy to be transfored;

@for(trans(j):@sum(fact(i):plan(i,j)\*pwhe(i,j))<6000);!one translator cannot afford more than 6000;

!soft\_condition;

@for(trans(j):@sum(fact(i):pwhe(i,j))+dm(j)-dp(j)=1);!one suppler best have one translator;

@sum(fact(i):@sum(trans(j):plan(i,j)\*pbrk(j)))+dm(9)-dp(9)=0;!the min lost;

@for(ans:fplan=plan\*pwhe);

End

问题三购买和运输方案

model:

sets:

!produce and buy;

week/1..24/:xA,xB,xC,SA,SB,SC,uA,uB,uC;!the amount to buy each week,the store in each week,the use in each week;

factory/1..50/:maxp,mid,TA,TB,TC;!factories max,midium production and its type;

buyplan(factory,week):x,rdm;!the buy plan in each factory in each week,the random change in each week;

!translation and loss;

translator/1..8/:tr;!the translators;

ploss(translator,week):lp;!the loss portion of each translator in each week;

tranplan(factory,translator,week):y,pwhe;!how much to translate and whether choose;

tplanw(factory,translator):tpw;!fixed week plan;

soft\_con/1..123/:dm,dp;!the soft conditions;

goal/1..5/:z;!the goalsin each preference;

endsets

data:

rdm=@qrand(123);

mid=@ole('D:\MCM2021\mid.xlsx','mid');!the midiam;

TA=@ole('D:\MCM2021\factoryinfo.xlsx','TA');

TB=@ole('D:\MCM2021\factoryinfo.xlsx','TB');

TC=@ole('D:\MCM2021\factoryinfo.xlsx','TC');!the type sell by each factory;

maxp=@ole('D:\MCM2021\factoryinfo.xlsx','MAXP');!the max production of each factory;

lp=@ole('D:\MCM2021\comp.xlsx','PL');!the loss;

enddata

!object;

min=10\*z(1)+5\*(z(2)+z(3))+3\*z(4);

z(1)=dp(97)+dp(98)+dp(99);!min price;

z(2)=@sum(week(j):dm(j)+dp(j+24));!more A less C;

z(3)=@sum(week(j):dp(j+48));!less lost;

z(4)=@sum(week(j):dm(j+99));!try ro store for 2 weeks;

!z(5)=@sum(week(j):dp(j+72));!one fact should try to have only one translator;

!conditions;

!hard conditions;

!production;

SA(1)=0;SB(1)=0;SC(1)=0;!innitial value;

@for(week(i)|i #ne# 1:SA(i)/0.6+SB(i)/0.66+SC(i)/0.72>28200\*2);!we should store for 2 weeks' production;

!for the first day;

xA(1)=uA(1);xB(1)=uB(1);xC(1)=uC(1);

@for(week(i)|i #ne# 1:SA(i)=SA(i-1)+xA(i)-uA(i);SB(i)=SB(i-1)+xB(i)-uB(i);SC(i)=SC(i-1)+xC(i)-uC(i));!the recursive relationship of S;

@for(week(j):xA(j)=@sum(factory(i):x(i,j)\*TA(i));xB(j)=@sum(factory(i):x(i,j)\*TB(i));xC(j)=@sum(factory(i):x(i,j)\*TC(i)));

@for(week(j):@for(factory(i):x(i,j)<(1-rdm(i,j))\*(maxp(i)+mid(i))/2+rdm(i,j)\*maxp(i)));!one factory has its limit;

!translation;

!@for(tranplan:@bin(pwhe));!whether choose this translator in this week;

@for(week(k):@for(translator(j):@sum(factory(i):y(i,j,k))<6000));!one translator cannot afford 60000;

@for(week(k):@for(factory(i):@sum(translator(j):y(i,j,k))=x(i,k)));!the goods should be translated;

!soft conditions;

!prices;

@for(week(j):uA(j)/0.6 + uB(j)/0.66 + uC(j)/0.72+dm(j+99)-dp(j+99)>28200);!the capability should be prepared;

1.2\*@sum(week:xA)+1.1\*@sum(week:xB)+@sum(week:xC)+dm(97)-dm(97)=528037.3;!bought fee;

@sum(week:SA+SB+SC)+dm(98)-dp(98)=779384.8;!store fee;

@sum(week:xA+xB+xC)+dm(99)-dp(99)=442676.8;!translation fee;

!more A,less C;

@for(week(j):xA(j)+dm(j)-dp(j)=28200\*0.6);

@for(week(j):xC(j)+dm(j+24)-dp(j+24)=0);

!min loss;

@for(week(k):@sum(tplanw(i,j):y(i,j,k)\*lp(j,k))+dm(k+48)-dp(k+48)=0);

!try to finish in one translator per fact;

@for(tranplan(i,j,k):pwhe(i,j,k)=@if(y(i,j,k)#gt#1,1,0));!calculate pwhe;

@for(week(k):@for(factory(i):@sum(translator(l):pwhe(i,l,k))+dm(k+72)-dp(k+72)=1));

End

问题四最大产能

model:

sets:

!produce and buy;

week/1..24/:xA,xB,xC,SA,SB,SC,uA,uB,uC,tmax;

!the amount to buy each week,the store in each week,the use in each week,the max week translation;

factory/1..119/:maxp,mid,TA,TB,TC;!factories max,midium production and its type;

stand/1..180/:maxpt,midt,TAt,TBt,TCt;

buyplan(factory,week):x,rdm;!the buy plan in each factory in each week,the random change in each week;

!translation and loss;

translator/1..8/:tr;!the translators;

ploss(translator,week):lp;!the loss portion of each translator in each week;

!soft\_conditions;

soft\_var/1..3/:dm,dp;

tranplan(factory,translator,week):y;!how much to translate;

endsets

data:

rdm=@qrand(123);

midt=@ole('D:\MCM2021\TOPSISall1.xlsx','mid');!the midiam;

TAt=@ole('D:\MCM2021\TOPSISall1.xlsx','TA');

TBt=@ole('D:\MCM2021\TOPSISall1.xlsx','TB');

TCt=@ole('D:\MCM2021\TOPSISall1.xlsx','TC');!the type sell by each factory;

maxpt=@ole('D:\MCM2021\TOPSISall1.xlsx','MAXP');!the max production of each factory;

lp=@ole('D:\MCM2021\comp.xlsx','PL');!the loss;

tmax=@ole('D:\MCM2021\maxtran.xlsx','tmax');!the max week tran;

@ole('D:\MCM2021\pro4buy1.xlsx','buy')=x;!get the answer in xlsx;

enddata

@for(factory(i):maxp(i)=maxpt(i);mid(i)=midt(i);TA(i)=TAt(i);TB(i)=TBt(i);TC(i)=TCt(i));

!object;

max=c-(0.2\*dm(1)+0.1\*(dm(2)+dp(3)));!max production;

!conditions;

!hard conditions;

!production;

SA(1)=0;SB(1)=0;SC(1)=0;!innitial value;

@for(week(i)|i #ne# 1:SA(i)/0.6+SB(i)/0.66+SC(i)/0.72+dm(1)-dp(1)=2\*c);!we should store for 2 weeks' production;

!for the first day;

xA(1)>uA(1);xB(1)>uB(1);xC(1)>uC(1);

@for(week(i)|i #ne# 1:SA(i)=SA(i-1)+xA(i)-uA(i);SB(i)=SB(i-1)+xB(i)-uB(i);SC(i)=SC(i-1)+xC(i)-uC(i));!the recursive relationship of S;

@for(week(j):xA(j)=@sum(factory(i):x(i,j)\*TA(i));xB(j)=@sum(factory(i):x(i,j)\*TB(i));xC(j)=@sum(factory(i):x(i,j)\*TC(i)));

@for(week(j):@for(factory(i):x(i,j)<maxp(i)\*rdm(i,j)+(mid(i)+maxp(i))/2\*rdm(i,j)));!one factory has its limit;

@for(week(j):uA(j)/0.6 + uB(j)/0.66 + uC(j)/0.72=c);!the capability should be prepared;

!more A,less C;

@for(week(j):xA(j)+dm(2)-dp(2)=0.6\*c);

@for(week(j):xC(j)+dm(3)-dp(3)=0);

@for(week(j):xA(j)+xB(j)+xC(j)<tmax(j));!max week tran;

遗传算法

global fbuy;%最省钱购买计划

global lossp;%损失率

fbuy=xlsread('D:\MCM2021\supply3.xlsx','A1:X50');

lossp=xlsread('D:\MCM2021\comp.xlsx','B2:Y9');

%function [x,fval]=genes(fun,lb,ub,epslon,conditions,index)

%遗传算法函数，输入适应度函数，下界，上界，精度，条件函数默认<=0，参数数组（最大进化代数，种群大小,交叉概率，变异概率）

%记住！最大进化数要够大！种群大小也是！主要是交叉概率在起作用！交叉概率要0.5以上

X=xlsread('D:\MCM2021\supply3.xlsx','A1:X50');

%%

week=3;%第一周

%%

index=[200,100,0.7,0.01];

G=index(1);%最大进化代数

NP=index(2);%种群大小

pexc=index(3);%交叉概率

pdiv=index(4);%变异概率

%上界是第k期对每一个供应商的购买量

ub=X(:,week);

ub=ub';

%下界是0

lb=zeros(1,size(X,1));

%%

%维度

L=50;

%我们采取8\*50的形式表示一次运输

fvalbest=zeros(1,G);%初始化最适应值

Xbest=zeros(G,8\*L);%初始化最优个体

k=1;%第一代

%产生初始种群

group=rand(NP\*8,L).\*(ub-lb);%产生NP个个体的基因,连着的8个代表一个个体

%计算NP个个体的适应度

fit=zeros(NP,1);

for k=1:G

%计算适应度

index=1:8;

for i=1:NP

temp=group(index+(i-1)\*8,:);

%调整group,使得运输公司加和更容易小于6000且单个供给方的运输公司尽量少

temp=adjust(temp,ub);

group(index+(i-1)\*8,:)=temp;

more=sum(temp,2)-6000;

punish=7\*length(find(more>0));%不超过最大运输量的限制

%if conditions ~= [] %如果有罚函数

%punish=conditions(indiv);%罚函数

%end

fit(i)=fitfun(temp,week)-punish;%实数数组计算适应度

end

%找到初始最适应个体

fvalbest(k)=max(fit);

maxplace=find(fit==max(fit));

bestmat=group(index+maxplace(1),:);

Xbest(k,:)=bestmat(:)';

%%

%赌轮盘复制个体

stand=cumsum(fit)./sum(fit);%计算概率标准

for i=1:NP

p=rand;%得到一次概率

flag=find(stand>p);%找到第一个大于p的

if double(isempty(flag))==1

ch=NP;

else

ch=flag(1);

end

newG((i-1)\*8+index,:)=group((ch-1)\*8+index,:);%复制

end

%%

%交叉

chosen=double(rand(1,NP)<pexc);%得到是否选择该个体的数组

chosenplace=find(chosen==1);%被选择的位置

if mod(length(chosenplace),2)==1%不是成对的

%从没被选中的挑一个

nchosenplace=find(chosen==0);

newch=round(1+(length(nchosenplace)-1)\*rand);

chosen(nchosenplace(newch))=1;

chosenplace=find(chosen==1);%更新一下

end

%%

%现在保证成对了

for i=1:2:length(chosenplace)

exc1=newG((chosenplace(i)-1)\*8+index,:);

exc2=newG((chosenplace(i+1)-1)\*8+index,:);%找到一对

excplace=double(rand(1,L)<pexc);%哪个供应商的运输方案交换，保证落入的范围还是符合标准

tempexc=exc1(:,find(excplace==1));%拷贝交换位置

tranplace=double(rand(1,8)<pexc);%运输商分配交换

tranexp=find(tranplace==1);

temptran=exc1(tranexp,:);

exc1(tranexp,:)=exc2(tranexp,:);

exc2(tranexp,:)=temptran;

exc1(:,find(excplace==1))=exc2(:,find(excplace==1));

exc2(:,find(excplace==1))=tempexc;%完成交换

%更改种族

newG((chosenplace(i)-1)\*8+index,:)=exc1;

newG((chosenplace(i+1)-1)\*8+index,:)=exc2;

end

%%

%变异

chdiv=double(rand(1,NP)<pdiv);

divplace=find(chdiv==1);%找到变异位置

for j=1:length(divplace)

%每一个变异个体，哪些位置变异

temp=newG((divplace(j)-1)\*8+index,:);

dex=find(sum(temp)>0);%被购买的公司序号

divpoint=find(double(rand(1,length(dex))<pdiv)==1);%找到变异非0位置

%实际变异位置

diverpt=dex(divpoint);

if double(isempty(diverpt))==0

newG((divplace(j)-1)\*8+index,diverpt)=rand(8,length(diverpt)).\*(ub(diverpt)-lb(diverpt));%变异

end

end

%%

group=newG;%更新种群

end

适应函数

function fit=fitfun(X,week)%目标函数

global fbuy;%最省钱购买计划

global lossp;%损失率

bplan=fbuy(:,week);

lp=lossp(:,week);

%损失

F1=sum(sum(lp.\*X))\*0.001;

%最好由一家运

temp=double(X~=0);%哪些有运

comp=sum(temp,2);%一个列向量，表示每家由几家运

F2=length(find(comp>1));%超过一家的量

fit=100/(F1+F2);%适应度

end