import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import matplotlib.gridspec as gridspec

import time,random

plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签

plt.rcParams['axes.unicode\_minus'] = False # 用来正常显示负号

# 第一问

#原始数据

data1 = pd.read\_excel(r'附件1 近5年402家供应商的相关数据.xlsx',sheet\_name='企业的订货量（m³）')

data2 = pd.read\_excel(r'附件1 近5年402家供应商的相关数据.xlsx',sheet\_name='供应商的供货量（m³）')

#企业订货量

temp1 = data1.set\_index('供应商ID').drop(['材料分类'],axis=1)

#供应商供货量

temp2 = data2.set\_index('供应商ID').drop(['材料分类'],axis=1)

#订货量与供货量的差值

temp3 = temp2 - temp1

#材料分类系数

Dtype = list(data1['材料分类'])

Ddata = [0.95,0.86,0.79]

D = []

for i in Dtype:

if i == 'A':

D.append(Ddata[0])

elif i == 'B':

D.append(Ddata[1])

elif i == 'C':

D.append(Ddata[2])

else:

continue

#构造趋势函数

def f(x):

if x < 0:

return 4-4\*(0.8)\*\*x

elif x >= 0:

return (0.5)\*\*x

else:

pass

#画趋势图

plt.figure()

x=list(np.linspace(-1,1))

y = [f(i) for i in x]

plt.plot(x,y)

plt.xlabel('t')

plt.ylabel('y')

plt.savefig('qushi.jpg')

plt.show()

#供货量与订货量的差值比企业订货量

temp4=temp3/temp1

temp4 = temp4.fillna(0)

#计算供货商每周得分

roworder = 0

ttp = []

for row in temp3.itertuples():

lineorder = 0

rowlist = []

for i in list(row)[1:]:

#计算供应商每周对应综合得分

tp = f(np.array(temp4)[roworder][lineorder])\*abs(np.array(temp2)[roworder][lineorder])\*D[roworder]

rowlist.append(tp)

lineorder +=1

roworder +=1

ttp.append(rowlist)

#计算供货商总得分

d1 = []

for i in ttp:

d1.append(sum(i))

#构造供货商总得分表

df1 = pd.DataFrame(d1)

#构造问题一供货商综合得分表

ttpp = ttp.copy()

count = 0

for i in ttpp:

ct = 0

i.append(Dtype[count])

i.append(d1[count])

count +=1

index = temp1.index.values

columns = data1.columns.values[2:]

index = list(index)

columns = list(columns)

columns.append('材料分类')

columns.append('得分')

w1\_defen = pd.DataFrame(ttpp,index=index,columns=columns)

#w1\_defen.to\_csv('问题一供货商得分表.csv')

#供货商综合得分表排序后得到最优的50个供货商

df1\_data = df1.sort\_values(by=[0],ascending=False).head(50)

#最优的50家供货商得到的订购量

tptp = []

for i in list(df1\_data.index):

tptp.append(pd.DataFrame(list(data1.iloc[i])).T)

dff1 = pd.concat(tptp)

#dff1.to\_csv('dff1.csv')

#最优的50家供货商提供的供货量

tptp = []

for i in list(df1\_data.index):

tptp.append(pd.DataFrame(list(data2.iloc[i])).T)

dff1 = pd.concat(tptp)

#dff1.to\_csv('df1.csv')

#第二问

#第二问

#50家供应商的一周总平均提供

temp5 = dff1.set\_index(0).drop([1],axis=1)

tp = temp5.std(axis=1)#具体方差表

tmp = temp5.median(axis=1)#具体中位数

index = list(tp.index)

fc\_values = list(tp)

zvs\_values = list(tmp)

#方差中位数图

plt.figure(figsize=(25, 10), dpi=60)

x = np.linspace(1,len(index))

y1 = fc\_values#方差

y2 = zvs\_values#中位数

plt.plot(x,y1,x,y2)

plt.show()

#突变稳定初筛选

tubian\_list = []

wending\_list = []

for i in fc\_values:

if i >= 300:#方差大于300

addres = fc\_values.index(i)

if zvs\_values[addres] <= 100:#中位数小于100

tubian\_list.append(index[addres])

else:

wending\_list.append(index[addres])

else:

addres = fc\_values.index(i)

wending\_list.append(index[addres])

#处理类型成本

Dtype50 = []

for i in np.array(dff1):

Dtype50.append(i[1])

Ddata50 = [0.6,0.66,0.72]

D50 = []

for i in Dtype50:

if i == 'A':

D50.append(Ddata50[0])

elif i == 'B':

D50.append(Ddata50[1])

elif i == 'C':

D50.append(Ddata50[2])

else:

continue

#已订购的平均值

you0 = list((temp5 == 0).astype(int).sum(axis=1))

sum\_temp5 = list(temp5.sum(axis=1).values)

fei0 = [240 - i for i in you0]

mean\_temp5 = np.array(sum\_temp5)/np.array(fei0)

#按顺序提取最低限度的供应商

index = list(dff1[0])

pj\_values = list(mean\_temp5)

sum\_cn = 0

ct = 0

count = 0

pjcn = []

ljcn = []

for i in mean\_temp5:

sum\_cn = sum\_cn + (i/D50[ct])

ljcn.append(sum\_cn)

pjcn.append((i/D50[ct]))

if sum\_cn < 28200:

count+=1

ct+=1

continue

else:

break

gys = list(dff1[0].values)[0:count+1]

cpt = Dtype50[0:count+1]

gys23 = pd.DataFrame([gys,cpt,pjcn,ljcn],index=['供应商ID','材料分类','供应商平均产能','累计产能']).T.set\_index('供应商ID')

#gys23.to\_excel('选择的23个供应商.xlsx')

#看看原始23家的供货商供货量图

#for i in gys:

#if i in tubian\_list

tpw = []

for i in gys:

tpw.append(np.array(temp5.T[i]))

biao23 = pd.DataFrame(tpw)

count = 0

for j in tpw:

y = tpw[count]

x = biao23.columns.values

plt.figure()

plt.plot(x,y)

plt.xlabel(gys[count])

plt.show()

time.sleep(2)

count+=1

#biao23.to\_excel('选择的23个供应商具体数据.xlsx')

#第一周随机序列

def first\_suiji():

#23家分类

tubian\_id = ['S140','S139','S307','S395']

yichang\_id = ['S201','S348','S330','S308','S151']

teshu\_id = ['S108']

wending\_id = []

wending\_type=[]

for i in gys:

if (i not in yichang\_id )and (i not in tubian\_id) and (i not in teshu\_id):

wending\_id.append(i)

wending\_type.append(gys23.loc[i]['材料分类'])

#异常值取行与去除异常值后加入稳定值

yic = []

yic\_type = []

for j in yichang\_id:

addres = gys.index(j)

yic.append(np.array(biao23.loc[addres]))

yic\_type.append(gys23.loc[j]['材料分类'])

for k in yic:

for m in k:

if m > 6000:

addres = list(k).index(m)

k[addres] = np.median(k)#异常值使用中位数

yic\_suiji\_xulie = []

for t in yic:

yic\_max=t.max()

yic\_min=t.min()

yic\_suiji\_xulie.append(np.random.randint(yic\_max\*0.65,yic\_max, 1))

#特殊值取行

addres = gys.index(teshu\_id[0])

tes = np.array(biao23.loc[addres])

#稳定值取行

wen = []

for j in wending\_id:

addres = gys.index(j)

wen.append(np.array(biao23.loc[addres]))

#稳定值取随机序列

wen\_up = []

wen\_down = []

for i in wen:

wen\_up.append(i.max())

wen\_down.append(i.max()\*0.65)

wen\_suiji\_xulie = []

for i in wen\_up:

addres = wen\_up.index(i)

wen\_suiji\_xulie.append(np.random.randint(wen\_down[addres],i, 1))

#总的稳定值随机序列

wenzong = wen\_suiji\_xulie+yic\_suiji\_xulie

wenzong\_type = wending\_type+yic\_type

for i in yichang\_id:

wending\_id.append(i)

#特殊值取随机序列后加上最大值

text = []

tq = []

te\_type = gys23.loc[teshu\_id[0]]['材料分类']

for i in tes:

if i != tes[list(tes).index(tes.max())]:

text.append(i)

tq.append(i)

else:

text.append('s')

text[text.index('s')] = np.array(tq).mean()

tes\_up = np.array(text).max()

tes\_down = np.array(text).min()

te\_suiji\_xulie = np.random.randint(tes\_up\*0.65,tes\_up, 24)

tes\_max = list(tes).index(tes.max())

#te\_suiji\_xulie[random.randint(0,24)] = tes\_max

#te\_suiji = te\_suiji\_xulie[random.randint(0,23)]

#三个突变值取最大值的两个周

tb = []

tb\_type = []

for j in tubian\_id:

tb\_type.append(gys23.loc[j]['材料分类'])

tp = np.linspace(0,0,24)

addres = np.random.randint(0,24, 2)

if (addres[0] == 0) or (addres[1] == 23):

tb.append(np.random.randint(5999,6000, 1))

else:

tb.append(0)

return wenzong,tb,np.array([6000]),wenzong\_type,tb\_type,te\_type,[wending\_id,tubian\_id,teshu\_id]

#后面几周随机序列

def suiji():

#23家分类

tubian\_id = ['S140','S139','S307','S395']

yichang\_id = ['S201','S348','S330','S308','S151']

teshu\_id = ['S108']

wending\_id = []

wending\_type=[]

for i in gys:

if (i not in yichang\_id )and (i not in tubian\_id) and (i not in teshu\_id):

wending\_id.append(i)

wending\_type.append(gys23.loc[i]['材料分类'])

#异常值取行与去除异常值后加入稳定值

yic = []

yic\_type = []

for j in yichang\_id:

addres = gys.index(j)

yic.append(np.array(biao23.loc[addres]))

yic\_type.append(gys23.loc[j]['材料分类'])

for k in yic:

for m in k:

if m > 6000:

addres = list(k).index(m)

k[addres] = np.median(k)#异常值使用中位数

yic\_suiji\_xulie = []

for t in yic:

yic\_max=t.max()

yic\_min=t.min()

yic\_suiji\_xulie.append(np.random.randint(yic\_max\*0.2,yic\_max\*0.7, 1))

#特殊值取行

addres = gys.index(teshu\_id[0])

tes = np.array(biao23.loc[addres])

#稳定值取行

wen = []

for j in wending\_id:

addres = gys.index(j)

wen.append(np.array(biao23.loc[addres]))

#稳定值取随机序列

wen\_up = []

wen\_down = []

for i in wen:

wen\_up.append(i.max()\*0.7)

wen\_down.append(i.max()\*0.2)

wen\_suiji\_xulie = []

for i in wen\_up:

addres = wen\_up.index(i)

wen\_suiji\_xulie.append(np.random.randint(wen\_down[addres],i, 1))

#总的稳定值随机序列

wenzong = wen\_suiji\_xulie+yic\_suiji\_xulie

wenzong\_type = wending\_type+yic\_type

for i in yichang\_id:

wending\_id.append(i)

#特殊值取随机序列

text = []

tq = []

te\_type = gys23.loc[teshu\_id[0]]['材料分类']

for i in tes:

if i != tes[list(tes).index(tes.max())]:

text.append(i)

tq.append(i)

else:

text.append('s')

text[text.index('s')] = np.array(tq).mean()

tes\_up = np.array(text).max()

tes\_down = np.array(text).min()

te\_suiji\_xulie = np.random.randint(500,1000, 24)

tes\_max = list(tes).index(tes.max())

#te\_suiji\_xulie[random.randint(0,24)] = tes\_max

te\_suiji = te\_suiji\_xulie[random.randint(0,23)]

#三个突变值取最大值的两个周

tb = []

tb\_type = []

#for i in tubian\_id:

# addres = gys.index(i)

# tubian.append(np.array(biao23.loc[addres]))

for j in tubian\_id:

tb\_type.append(gys23.loc[j]['材料分类'])

tp = np.linspace(0,0,24)

addres = np.random.randint(0,24, 2)

if (addres[0] == 0) or (addres[1] == 23):

tb.append(np.random.randint(2000,6000, 1))

else:

tb.append(0)

return wenzong,tb,np.array(te\_suiji),wenzong\_type,tb\_type,te\_type,[wending\_id,tubian\_id,teshu\_id]

#第一周

result1 = []

mubiao1\_list = []

P1 = []

for i in range(1000):

suiji\_list = first\_suiji()

count +=0

t2 = []

t2.append(int(suiji\_list[2]))

suiji\_data = suiji\_list[0]+suiji\_list[1]+t2

suiji\_type = suiji\_list[3]+suiji\_list[4]

suiji\_type.append(suiji\_list[5])

gh = pd.DataFrame([suiji\_type,suiji\_data]).T

A = gh[gh[0] == 'A'][1].values.sum()

B = gh[gh[0] == 'B'][1].values.sum()

C = gh[gh[0] == 'C'][1].values.sum()

if ((A+B+C) <= 48000) and ((A/0.6+B/0.66+C/0.72)>=(28200\*2)):

P1.append(A/0.6+B/0.66+C/0.72)

mubiao1 = 1.2\*A+1.1\*B+C

mubiao1\_list.append(mubiao1)

result1.append(suiji\_list)

else:

continue

addres = mubiao1\_list.index(min(mubiao1\_list)[0])

fliter = []

for i in result1[addres][0]:

fliter.append(int(i))

for i in result1[addres][1]:

fliter.append(i)

fliter.append(int(result1[addres][2]))

fliter.append(int(P1[addres]))

fliter.append(int(min(mubiao1\_list)))

fliter1 = fliter.copy()

#后面几周

#筛选对应材料的供货商

p = int(P1[addres])

fliter\_list = []

P\_list = [int(P1[addres])]

CT = 0

for i in range(2,25):

result2 = []

mubiao2\_list = []

Pn = []

p=P\_list[CT]

for i in range(500):

suiji\_list = suiji()

count +=0

t2 = []

t2.append(int(suiji\_list[2]))

suiji\_data = suiji\_list[0]+suiji\_list[1]+t2

suiji\_type = suiji\_list[3]+suiji\_list[4]

suiji\_type.append(suiji\_list[5])

gh = pd.DataFrame([suiji\_type,suiji\_data]).T

A = gh[gh[0] == 'A'][1].values.sum()

B = gh[gh[0] == 'B'][1].values.sum()

C = gh[gh[0] == 'C'][1].values.sum()

if ((A+B+C) <= 48000) and ((A/0.6+B/0.66+C/0.72)>=(28200\*2)-p):

Pn.append(A/0.6+B/0.66+C/0.72)

mubiao2 = 1.2\*A+1.1\*B+C

mubiao2\_list.append(mubiao1)

result2.append(suiji\_list)

else:

continue

addres = mubiao2\_list.index(min(mubiao2\_list)[0])

fliter = []

for i in result2[addres][0]:

fliter.append(i[0])

for i in result2[addres][1]:

tmp = []

try:

for j in i:

if j != 0:

tmp.append(j[0])

else:

tmp.append(j)

for k in tmp:

fliter.append(k)

except:

fliter.append(i)

fliter.append(int(result2[addres][2]))

fliter.append(int(Pn[addres]))

fliter.append(int(min(mubiao2\_list)))

fliter\_list.append(fliter)

P\_list.append(int(Pn[addres]))

CT +=1

#整体订购方案

t = [fliter1]

for i in fliter\_list:

t.append(i)

ttp = []

for i in suiji\_list[-1]:

for j in i:

ttp.append(j)

ttp.append('原料对应产能')

ttp.append('原材料费用')

dinggoufangan=pd.DataFrame(t,columns=ttp,index=list(np.linspace(1,24,24))).T

t = []

for i in dinggoufangan.values:

ttttt = []

for j in i:

ttttt.append(int(j))

t.append(np.array(ttttt))

dinggoufangan = pd.DataFrame(np.array(t),index=dinggoufangan.index.values,columns=dinggoufangan.columns.values)

dinggoufangan.to\_excel('订购草案.xlsx')

#dinggoufangan

#转换填表格式

temp7 = pd.read\_excel(r'订购草案.xlsx')

t23 = temp7.iloc[0:23,:]

tp = []

for i in np.linspace(0,24,25):

tp.append(int(i))

t23.columns = tp

tp = []

for i in data1['供应商ID'].values:

c = [i]

for j in np.linspace(0,0,24):

c.append(int(j))

tp.append(c)

s402 = pd.DataFrame(tp)

ct = 0

for i in t23[0].values:

index\_23 = int(i[-3:])-1

s402.iloc[index\_23,:] = t23.iloc[ct,:]

ct += 1

s402.to\_csv(r'订购草案终.csv')

temp7 = pd.read\_excel('订购草案.xlsx')

data3 = pd.read\_excel('附件2 近5年8家转运商的相关数据.xlsx')

temp6 = data3.set\_index('转运商ID')

tpq = []

for i in np.array(temp6):

ttp = []

for j in i:

if j != 0:

ttp.append(j\*0.01)

else:

continue

tpq.append(ttp)

ct = 1

for i in tpq:

x = np.linspace(0,len(i),len(i))

y = list(i)

fig = plt.figure(figsize=(8,11),dpi=98)

na=plt.subplot(4,1,1)

na.plot(x,y,label='转运商{}'.format(ct))

na.legend()

ct+=1

plt.show()

#fig.savefig('zhuanyun.jpg')

#计算转运商信息

tp = []

for i in np.linspace(1,8,8):

tp.append(int(i))

tpq\_mean = []

for i in tpq:

tpq\_mean.append(np.array(i).mean())

plt.figure(dpi=95)

plt.ylabel('平均转运损耗')

plt.xlabel('转运商')

plt.plot(tp,tpq\_mean)

plt.savefig('zhunayunpj.jpg')

plt.show()

sh\_p = pd.DataFrame([tp,tpq\_mean]).T.sort\_values(by=1)

#随机转运的转运商选择结果 24

def suiji\_zhuanyun(count):

zy1 = []

for i in range(0,count):

zy1.append(int(np.random.randint(1,9,1)))

return zy1

#随机损耗 1~8

def suiji\_sunhao():

ss = []

tpq\_suiji = []

for i in tpq:

tpq\_suiji.append(random.sample(list(np.array(i)),1)[0])

return tpq\_suiji

#对应的概率

sjz = suiji\_zhuanyun(23)

sh = suiji\_sunhao()

temp = []

for i in sjz:

temp.append(sh[i-1])

c=0

q = []

for i in temp7.values[0:-2]:#.sort\_values(by=k-1)[::-1][k-1]

t = list(i)

t.append(c)

q.append(t)

c+=1

temp1 = pd.DataFrame(q)

pm = sh\_p[0].values.copy()

tp111 = [] #记录选中供货商值

tp222 = [] #记录选中供货商和

tp333 = [] #记录供货商名字

tp444 = [] #记录转运商名字

#tp555 = 0 #记录检索到哪一位置供应商了

tp666 = [] #记录供货商供货量

for k in range(2,26):

ghs\_tp = temp1.sort\_values(k-1)[::-1].iloc[:,k-1]#供货商数据 从大到小

ghs\_mz = temp1.sort\_values(k-1)[::-1].iloc[:,0].values#供货商名字 从大到小

#zys\_tp = temp1.sort\_values(k-1)[::-1].iloc[:,-1]#对应损失率

tp11 = [] #记录选中供货商值

tp22 = [] #记录选中供货商和

tp33 = [] #记录供货商名字

tp44 = [] #记录转运商名字

#tp55 = 0 #记录检索到哪一位置供应商了

tp66 = [] #记录供货商供货量

sjz\_t = sjz.copy()

pm = sh\_p[0].values.copy()

for j in pm:

#try:

tp1 = [] #记录选中供货商值

tp2 = 0 #记录选中供货商和

tp3 = [] #记录供货商名字

tp4 = [] #记录转运商名字

#tp5 = 0 #记录检索到哪一位置供应商了

tp6 = [] #记录供货商供货量

count = 0

yz = []

for i in ghs\_mz:

try:

yz.append([i,ghs\_tp[count]])

except:

yz.append([i,0])

count+=1

for i in yz:

tp2 +=i[1]

if tp2 <= 6000:

tp3.append(i[0])

tp6.append(i[1])

else:

tp2 -=i[1]

tp5+=1

tp5 = j

ghs\_tp = [x for x in ghs\_tp if x not in tp6]

pm = pm[1:]

ghs\_mz = [x for x in ghs\_mz if x not in tp3]

tp11.append(tp1)

tp22.append([tp2])

tp33.append(tp3)

tp44.append(tp4)

#tp55.append([tp5])

tp66.append(tp6)

#except:

#continue

tp111.append(tp11)

tp222.append([tp22])

tp333.append(tp33)

tp444.append(tp44)

#tp555.append([tp55])

tp666.append(tp66)

kkk = []

for k in range(24):

pm = sh\_p[0].values.copy()

t = []

for j in pm:

tp = []

for i in range(8):

name = tp333[k][i]

shuju = tp666[k][i]

tp.append(pd.DataFrame([name,shuju],index=[0,j]).T.set\_index(0))

t.append(tp)#3 6 2 8 4 1 7 5

#0 1 2 3 4 5 6 7

kkk.append(pd.concat([t[0][5],t[0][2],t[0][0],t[0][4],t[0][7],t[0][1],t[0][6],t[0][3]],axis=1))

result = pd.concat(kkk,axis=1)

result.to\_csv('转运方案2终.csv')

temp7 = pd.read\_excel('订购草案.xlsx')

t23 = temp7.iloc[0:23,:]

tp = []

for i in np.linspace(0,24,25):

tp.append(int(i))

t23.columns = tp

tp = []

for i in data2['供应商ID'].values:

c = [i]

for j in np.linspace(0,0,24):

c.append(int(j))

tp.append(c)

s402 = pd.DataFrame(tp)

ct = 0

for i in t23[0].values:

index\_23 = int(i[-3:])-1

s402.iloc[index\_23,:] = t23.iloc[ct,:]

ct += 1

s402.to\_csv('转运方案2终终.csv')

#第三问

t = dff1[0:46] #.set\_index(0)

t1 = pd.concat([t[t[1] == 'A'],t[t[1] == 'B'],t[t[1] == 'C']])

#取每行平均值

t2 = t1.set\_index(0).drop([1],axis=1)

you0 = list((t2 == 0).astype(int).sum(axis=1))

sum\_t2 = list(t2.sum(axis=1).values)

fei0 = [240 - i for i in you0]

mean\_t2 = np.array(sum\_t2)/np.array(fei0)

index = list(t1.index.values)

pj\_values = list(mean\_t2)

sum\_cn = 0

ct = 0

count = 0

pjcn = []

ljcn = []

for i in mean\_t2:

sum\_cn = sum\_cn + (i/D50[ct])

ljcn.append(sum\_cn)

pjcn.append((i/D50[ct]))

if sum\_cn < 28200:

count+=1

ct+=1

continue

else:

break

gys = t2.index.values[0:len(pjcn)]

cpt = t1[1].values[0:len(gys)]

gyswz = pd.DataFrame([gys,cpt,pjcn,ljcn],index=['供应商ID','材料分类','供应商平均产能','累计产能']).T.set\_index('供应商ID')

#gyswz

#看看原始31家的供货商供货量图

'''

#for i in gys:

#if i in tubian\_list

tpw = []

for i in gys:

tpw.append(np.array(t2.T[i]))

biao31 = pd.DataFrame(tpw)

count = 0

for j in tpw:

y = tpw[count]

x = biao31.columns.values

plt.figure()

plt.plot(x,y)

plt.xlabel(gys[count])

plt.show()

time.sleep(2)

count+=1

'''

# 分步式考虑

#第一周随机序列

gys = list(gyswz.index.values)

def first\_suiji\_3():

#31家分类

tubian\_id = ['S140','S307','S395','S139']

yichang\_id = ['S151','S308','S330','S348','S201','S143']

teshu\_id = ['S108']

wending\_id = []

wending\_type=[]

for i in gys:

if (i not in yichang\_id )and (i not in tubian\_id) and (i not in teshu\_id):

wending\_id.append(i)

wending\_type.append(gyswz.loc[i]['材料分类'])

#异常值取行与去除异常值后加入稳定值

yic = []

yic\_type = []

for j in yichang\_id:

addres = gys.index(j)

yic.append(np.array(biao31.loc[addres]))

yic\_type.append(gyswz.loc[j]['材料分类'])

for k in yic:

for m in k:

if m > 2500:

addres = list(k).index(m)

k[addres] = np.median(k)#异常值使用中位数

yic\_suiji\_xulie = []

for t in yic:

yic\_max=t.max()

yic\_min=t.min()

yic\_suiji\_xulie.append(np.random.randint(yic\_max\*0.65,yic\_max, 1))

#特殊值取行

addres = gys.index(teshu\_id[0])

tes = np.array(biao31.loc[addres])

#稳定值取行

wen = []

for j in wending\_id:

addres = gys.index(j)

wen.append(np.array(biao31.loc[addres]))

#稳定值取随机序列

wen\_up = []

wen\_down = []

for i in wen:

wen\_up.append(i.max())

wen\_down.append(i.max()\*0.65)

wen\_suiji\_xulie = []

for i in wen\_up:

addres = wen\_up.index(i)

wen\_suiji\_xulie.append(np.random.randint(wen\_down[addres],i, 1))

#总的稳定值随机序列

wenzong = wen\_suiji\_xulie+yic\_suiji\_xulie

wenzong\_type = wending\_type+yic\_type

for i in yichang\_id:

wending\_id.append(i)

#特殊值取随机序列后加上最大值

text = []

tq = []

te\_type = gyswz.loc[teshu\_id[0]]['材料分类']

for i in tes:

if i != tes[list(tes).index(tes.max())]:

text.append(i)

tq.append(i)

else:

text.append('s')

text[text.index('s')] = np.array(tq).mean()

tes\_up = np.array(text).max()

tes\_down = np.array(text).min()

te\_suiji\_xulie = np.random.randint(tes\_up\*0.65,tes\_up, 24)

tes\_max = list(tes).index(tes.max())

#te\_suiji\_xulie[random.randint(0,24)] = tes\_max

#te\_suiji = te\_suiji\_xulie[random.randint(0,23)]

#三个突变值取最大值的两个周

tb = []

tb\_type = []

#for i in tubian\_id:

# addres = gys.index(i)

# tubian.append(np.array(biao31.loc[addres]))

for j in tubian\_id:

tb\_type.append(gyswz.loc[j]['材料分类'])

tp = np.linspace(0,0,24)

addres = np.random.randint(0,24, 2)

if (addres[0] == 0) or (addres[1] == 23):

tb.append(np.random.randint(5999,6000, 1))

else:

tb.append(0)

return wenzong,tb,np.array([6000]),wenzong\_type,tb\_type,te\_type,[wending\_id,tubian\_id,teshu\_id]

#后面几周随机序列

def suiji\_3():

#31家分类

tubian\_id = ['S140','S307','S395','S139']

yichang\_id = ['S151','S308','S330','S348','S201','S143']

teshu\_id = ['S108']

wending\_id = []

wending\_type=[]

for i in gys:

if (i not in yichang\_id )and (i not in tubian\_id) and (i not in teshu\_id):

wending\_id.append(i)

wending\_type.append(gyswz.loc[i]['材料分类'])

#异常值取行与去除异常值后加入稳定值

yic = []

yic\_type = []

for j in yichang\_id:

addres = gys.index(j)

yic.append(np.array(biao31.loc[addres]))

yic\_type.append(gyswz.loc[j]['材料分类'])

for k in yic:

for m in k:

if m > 2500:

addres = list(k).index(m)

k[addres] = np.median(k)#异常值使用中位数

yic\_suiji\_xulie = []

for t in yic:

yic\_max=t.max()

yic\_min=t.min()

yic\_suiji\_xulie.append(np.random.randint(yic\_max\*0.2,yic\_max\*0.7, 1))

#特殊值取行

addres = gys.index(teshu\_id[0])

tes = np.array(biao31.loc[addres])

#稳定值取行

wen = []

for j in wending\_id:

addres = gys.index(j)

wen.append(np.array(biao31.loc[addres]))

#稳定值取随机序列

wen\_up = []

wen\_down = []

for i in wen:

wen\_up.append(i.max()\*0.7)

wen\_down.append(i.max()\*0.2)

wen\_suiji\_xulie = []

for i in wen\_up:

addres = wen\_up.index(i)

wen\_suiji\_xulie.append(np.random.randint(wen\_down[addres],i, 1))

#总的稳定值随机序列

wenzong = wen\_suiji\_xulie+yic\_suiji\_xulie

wenzong\_type = wending\_type+yic\_type

for i in yichang\_id:

wending\_id.append(i)

#特殊值取随机序列

text = []

tq = []

te\_type = gyswz.loc[teshu\_id[0]]['材料分类']

for i in tes:

if i != tes[list(tes).index(tes.max())]:

text.append(i)

tq.append(i)

else:

text.append('s')

text[text.index('s')] = np.array(tq).mean()

tes\_up = np.array(text).max()

tes\_down = np.array(text).min()

te\_suiji\_xulie = np.random.randint(500,1000, 24)

tes\_max = list(tes).index(tes.max())

#te\_suiji\_xulie[random.randint(0,24)] = tes\_max

te\_suiji = te\_suiji\_xulie[random.randint(0,23)]

#三个突变值取最大值的两个周

tb = []

tb\_type = []

#for i in tubian\_id:

# addres = gys.index(i)

# tubian.append(np.array(biao31.loc[addres]))

for j in tubian\_id:

tb\_type.append(gyswz.loc[j]['材料分类'])

tp = np.linspace(0,0,24)

addres = np.random.randint(0,24, 2)

if (addres[0] == 0) or (addres[1] == 23):

tb.append(np.random.randint(2000,6000, 1))

else:

tb.append(0)

return wenzong,tb,np.array(te\_suiji),wenzong\_type,tb\_type,te\_type,[wending\_id,tubian\_id,teshu\_id]

#随机转运的转运商选择

def suiji\_zhuanyun\_3():

zy1 = []

for i in range(0,31):

zy1.append(int(np.random.randint(1,9,1)))

return zy1

#第一周

result1 = []

mubiao1\_list = []

P1 = []

for i in range(1000):

suiji\_list = first\_suiji\_3()

count +=0

t2 = []

t2.append(int(suiji\_list[2]))

suiji\_data = suiji\_list[0]+suiji\_list[1]+t2

suiji\_type = suiji\_list[3]+suiji\_list[4]

suiji\_type.append(suiji\_list[5])

gh = pd.DataFrame([suiji\_type,suiji\_data]).T

A = gh[gh[0] == 'A'][1].values.sum()

B = gh[gh[0] == 'B'][1].values.sum()

C = gh[gh[0] == 'C'][1].values.sum()

if ((A+B+C) <= 48000) and ((A/0.6+B/0.66+C/0.72)>=(28200\*2)):

P1.append(A/0.6+B/0.66+C/0.72)

mubiao1 = 1.2\*A+1.1\*B+C

mubiao1\_list.append(mubiao1)

result1.append(suiji\_list)

else:

continue

addres = mubiao1\_list.index(min(mubiao1\_list)[0])

fliter = []

for i in result1[addres][0]:

fliter.append(int(i))

for i in result1[addres][1]:

fliter.append(i)

fliter.append(int(result1[addres][2]))

fliter.append(int(P1[addres]))

fliter.append(int(min(mubiao1\_list)))

fliter1 = fliter.copy()

#后面几周

#筛选对应材料的供货商

p = int(P1[addres])

fliter\_list = []

P\_list = [int(P1[addres])]

CT = 0

for i in range(2,25):

result2 = []

mubiao2\_list = []

Pn = []

p=P\_list[CT]

for i in range(200):

try:

suiji\_list = suiji\_3()

count +=0

t2 = []

t2.append(int(suiji\_list[2]))

suiji\_data = suiji\_list[0]+suiji\_list[1]+t2

suiji\_type = suiji\_list[3]+suiji\_list[4]

suiji\_type.append(suiji\_list[5])

gh = pd.DataFrame([suiji\_type,suiji\_data]).T

A = gh[gh[0] == 'A'][1].values.sum()

B = gh[gh[0] == 'B'][1].values.sum()

C = gh[gh[0] == 'C'][1].values.sum()

if ((A+B+C) <= 48000) and ((A/0.6+B/0.66+C/0.72)>=(28200\*2)-p):

Pn.append(A/0.6+B/0.66+C/0.72)

mubiao2 = 1.2\*A+1.1\*B+C

mubiao2\_list.append(mubiao1)

result2.append(suiji\_list)

else:

continue

except:

continue

addres = mubiao2\_list.index(min(mubiao2\_list)[0])

fliter = []

for i in result2[addres][0]:

fliter.append(i[0])

for i in result2[addres][1]:

tmp = []

try:

for j in i:

if j != 0:

tmp.append(j[0])

else:

tmp.append(j)

for k in tmp:

fliter.append(k)

except:

fliter.append(i)

fliter.append(int(result2[addres][2]))

fliter.append(int(Pn[addres]))

fliter.append(int(min(mubiao2\_list)))

fliter\_list.append(fliter)

P\_list.append(int(Pn[addres]))

CT +=1

#整体订购方案

t = [fliter1]

for i in fliter\_list:

t.append(i)

ttp = []

for i in suiji\_list[-1]:

for j in i:

ttp.append(j)

ttp.append('原料对应产能')

ttp.append('原材料费用')

dinggoufangan=pd.DataFrame(t,columns=ttp,index=list(np.linspace(1,24,24))).T

t = []

for i in dinggoufangan.values:

ttttt = []

for j in i:

ttttt.append(int(j))

t.append(np.array(ttttt))

dinggoufangan3 = pd.DataFrame(np.array(t),index=dinggoufangan.index.values,columns=dinggoufangan.columns.values)

dinggoufangan3.to\_csv('订购方案')

temp7 = dinggoufangan3.copy()

data3 = pd.read\_excel('附件2 近5年8家转运商的相关数据.xlsx')

temp6 = data3.set\_index('转运商ID')

tpq = []

for i in np.array(temp6):

ttp = []

for j in i:

if j != 0:

ttp.append(j\*0.01)

else:

continue

tpq.append(ttp)

ct = 1

for i in tpq:

x = np.linspace(0,len(i),len(i))

y = list(i)

fig = plt.figure(figsize=(8,11),dpi=98)

na=plt.subplot(4,1,1)

na.plot(x,y,label='转运商{}'.format(ct))

na.legend()

ct+=1

plt.show()

#fig.savefig('zhuanyun.jpg')

#计算转运商信息

tp = []

for i in np.linspace(1,8,8):

tp.append(int(i))

tpq\_mean = []

for i in tpq:

tpq\_mean.append(np.array(i).mean())

plt.figure(dpi=95)

plt.ylabel('平均转运损耗')

plt.xlabel('转运商')

plt.plot(tp,tpq\_mean)

plt.savefig('zhunayunpj.jpg')

plt.show()

sh\_p = pd.DataFrame([tp,tpq\_mean]).T.sort\_values(by=1)

#随机转运的转运商选择结果 24

def suiji\_zhuanyun(count):

zy1 = []

for i in range(0,count):

zy1.append(int(np.random.randint(1,9,1)))

return zy1

#随机损耗 1~8

def suiji\_sunhao():

ss = []

tpq\_suiji = []

for i in tpq:

tpq\_suiji.append(random.sample(list(np.array(i)),1)[0])

return tpq\_suiji

#对应的概率

sjz = suiji\_zhuanyun(23)

sh = suiji\_sunhao()

temp = []

for i in sjz:

temp.append(sh[i-1])

c=0

q = []

for i in temp7.values[0:-2]:#.sort\_values(by=k-1)[::-1][k-1]

t = list(i)

t.append(c)

q.append(t)

c+=1

temp1 = pd.DataFrame(q)

pm = sh\_p[0].values.copy()

tp111 = [] #记录选中供货商值

tp222 = [] #记录选中供货商和

tp333 = [] #记录供货商名字

tp444 = [] #记录转运商名字

#tp555 = 0 #记录检索到哪一位置供应商了

tp666 = [] #记录供货商供货量

for k in range(2,26):

ghs\_tp = temp1.sort\_values(k-1)[::-1].iloc[:,k-1]#供货商数据 从大到小

ghs\_mz = temp1.sort\_values(k-1)[::-1].iloc[:,0].values#供货商名字 从大到小

#zys\_tp = temp1.sort\_values(k-1)[::-1].iloc[:,-1]#对应损失率

tp11 = [] #记录选中供货商值

tp22 = [] #记录选中供货商和

tp33 = [] #记录供货商名字

tp44 = [] #记录转运商名字

#tp55 = 0 #记录检索到哪一位置供应商了

tp66 = [] #记录供货商供货量

sjz\_t = sjz.copy()

pm = sh\_p[0].values.copy()

for j in pm:

#try:

tp1 = [] #记录选中供货商值

tp2 = 0 #记录选中供货商和

tp3 = [] #记录供货商名字

tp4 = [] #记录转运商名字

#tp5 = 0 #记录检索到哪一位置供应商了

tp6 = [] #记录供货商供货量

count = 0

yz = []

for i in ghs\_mz:

try:

yz.append([i,ghs\_tp[count]])

except:

yz.append([i,0])

count+=1

for i in yz:

tp2 +=i[1]

if tp2 <= 6000:

tp3.append(i[0])

tp6.append(i[1])

else:

tp2 -=i[1]

tp5+=1

tp5 = j

ghs\_tp = [x for x in ghs\_tp if x not in tp6]

pm = pm[1:]

ghs\_mz = [x for x in ghs\_mz if x not in tp3]

tp11.append(tp1)

tp22.append([tp2])

tp33.append(tp3)

tp44.append(tp4)

#tp55.append([tp5])

tp66.append(tp6)

#except:

#continue

tp111.append(tp11)

tp222.append([tp22])

tp333.append(tp33)

tp444.append(tp44)

#tp555.append([tp55])

tp666.append(tp66)

kkk = []

for k in range(24):

pm = sh\_p[0].values.copy()

t = []

for j in pm:

tp = []

for i in range(8):

name = tp333[k][i]

shuju = tp666[k][i]

tp.append(pd.DataFrame([name,shuju],index=[0,j]).T.set\_index(0))

t.append(tp)#3 6 2 8 4 1 7 5

#0 1 2 3 4 5 6 7

kkk.append(pd.concat([t[0][5],t[0][2],t[0][0],t[0][4],t[0][7],t[0][1],t[0][6],t[0][3]],axis=1))

result = pd.concat(kkk,axis=1)

result.to\_csv('转运方案3同终.csv')

#格式转换

temp7 = result.copy()

t23 = temp7.iloc[0:23,:]

tp = []

for i in np.linspace(0,24,25):

tp.append(int(i))

t23.columns = tp

tp = []

for i in data2['供应商ID'].values:

c = [i]

for j in np.linspace(0,0,24):

c.append(int(j))

tp.append(c)

s402 = pd.DataFrame(tp)

ct = 0

for i in t23[0].values:

index\_23 = int(i[-3:])-1

s402.iloc[index\_23,:] = t23.iloc[ct,:]

ct += 1

s402.to\_csv('转运方案3终终.csv')

#分步目标值

cs\_type = list(gyswz['材料分类'].values)

suihaobiao = []

for i in range(24):

suihaobiao.append(suiji\_sunhao())

ck = suihaobiao[0]

for j in suihaobiao[1:]:

ck = ck + j

roworder = 0

rowlist = []

for row in fenbu.fillna(0).itertuples():

ttp = []

temp = np.array(row[1:])\*np.array(ck)+np.array(row[1:])

roworder+=1

rowlist.append(temp)

count = 0

tptp = []

for i in cs\_type:

if i == 'A':

tptp.append(rowlist[count].sum()\*1.2)

elif i == 'B':

tptp.append(rowlist[count].sum()\*1.1)

else:

tptp.append(rowlist[count].sum())

count+=1

#min(np.array(tptp))

sum(tptp)

# 同步式考虑

#第一周随机序列

gys = list(gyswz.index.values)

def first\_suiji\_3():

#31家分类

tubian\_id = ['S140','S307','S395','S139']

yichang\_id = ['S151','S308','S330','S348','S201','S143']

teshu\_id = ['S108']

wending\_id = []

wending\_type=[]

for i in gys:

if (i not in yichang\_id )and (i not in tubian\_id) and (i not in teshu\_id):

wending\_id.append(i)

wending\_type.append(gyswz.loc[i]['材料分类'])

#异常值取行与去除异常值后加入稳定值

yic = []

yic\_type = []

for j in yichang\_id:

addres = gys.index(j)

yic.append(np.array(biao31.loc[addres]))

yic\_type.append(gyswz.loc[j]['材料分类'])

for k in yic:

for m in k:

if m > 2500:

addres = list(k).index(m)

k[addres] = np.median(k)#异常值使用中位数

yic\_suiji\_xulie = []

for t in yic:

yic\_max=t.max()

yic\_min=t.min()

yic\_suiji\_xulie.append(np.random.randint(yic\_max\*0.65,yic\_max, 1))

#特殊值取行

addres = gys.index(teshu\_id[0])

tes = np.array(biao31.loc[addres])

#稳定值取行

wen = []

for j in wending\_id:

addres = gys.index(j)

wen.append(np.array(biao31.loc[addres]))

#稳定值取随机序列

wen\_up = []

wen\_down = []

for i in wen:

wen\_up.append(i.max())

wen\_down.append(i.max()\*0.65)

wen\_suiji\_xulie = []

for i in wen\_up:

addres = wen\_up.index(i)

wen\_suiji\_xulie.append(np.random.randint(wen\_down[addres],i, 1))

#总的稳定值随机序列

wenzong = wen\_suiji\_xulie+yic\_suiji\_xulie

wenzong\_type = wending\_type+yic\_type

for i in yichang\_id:

wending\_id.append(i)

#特殊值取随机序列后加上最大值

text = []

tq = []

te\_type = gyswz.loc[teshu\_id[0]]['材料分类']

for i in tes:

if i != tes[list(tes).index(tes.max())]:

text.append(i)

tq.append(i)

else:

text.append('s')

text[text.index('s')] = np.array(tq).mean()

tes\_up = np.array(text).max()

tes\_down = np.array(text).min()

te\_suiji\_xulie = np.random.randint(tes\_up\*0.65,tes\_up, 24)

tes\_max = list(tes).index(tes.max())

#te\_suiji\_xulie[random.randint(0,24)] = tes\_max

#te\_suiji = te\_suiji\_xulie[random.randint(0,23)]

#三个突变值取最大值的两个周

tb = []

tb\_type = []

#for i in tubian\_id:

# addres = gys.index(i)

# tubian.append(np.array(biao31.loc[addres]))

for j in tubian\_id:

tb\_type.append(gyswz.loc[j]['材料分类'])

tp = np.linspace(0,0,24)

addres = np.random.randint(0,24, 2)

if (addres[0] == 0) or (addres[1] == 23):

tb.append(np.random.randint(5999,6000, 1))

else:

tb.append(0)

return wenzong,tb,np.array([6000]),wenzong\_type,tb\_type,te\_type,[wending\_id,tubian\_id,teshu\_id]

#后面几周随机序列

def suiji\_3():

#31家分类

tubian\_id = ['S140','S307','S395','S139']

yichang\_id = ['S151','S308','S330','S348','S201','S143']

teshu\_id = ['S108']

wending\_id = []

wending\_type=[]

for i in gys:

if (i not in yichang\_id )and (i not in tubian\_id) and (i not in teshu\_id):

wending\_id.append(i)

wending\_type.append(gyswz.loc[i]['材料分类'])

#异常值取行与去除异常值后加入稳定值

yic = []

yic\_type = []

for j in yichang\_id:

addres = gys.index(j)

yic.append(np.array(biao31.loc[addres]))

yic\_type.append(gyswz.loc[j]['材料分类'])

for k in yic:

for m in k:

if m > 2500:

addres = list(k).index(m)

k[addres] = np.median(k)#异常值使用中位数

yic\_suiji\_xulie = []

for t in yic:

yic\_max=t.max()

yic\_min=t.min()

yic\_suiji\_xulie.append(np.random.randint(yic\_max\*0.2,yic\_max\*0.7, 1))

#特殊值取行

addres = gys.index(teshu\_id[0])

tes = np.array(biao31.loc[addres])

#稳定值取行

wen = []

for j in wending\_id:

addres = gys.index(j)

wen.append(np.array(biao31.loc[addres]))

#稳定值取随机序列

wen\_up = []

wen\_down = []

for i in wen:

wen\_up.append(i.max()\*0.7)

wen\_down.append(i.max()\*0.2)

wen\_suiji\_xulie = []

for i in wen\_up:

addres = wen\_up.index(i)

wen\_suiji\_xulie.append(np.random.randint(wen\_down[addres],i, 1))

#总的稳定值随机序列

wenzong = wen\_suiji\_xulie+yic\_suiji\_xulie

wenzong\_type = wending\_type+yic\_type

for i in yichang\_id:

wending\_id.append(i)

#特殊值取随机序列

text = []

tq = []

te\_type = gyswz.loc[teshu\_id[0]]['材料分类']

for i in tes:

if i != tes[list(tes).index(tes.max())]:

text.append(i)

tq.append(i)

else:

text.append('s')

text[text.index('s')] = np.array(tq).mean()

tes\_up = np.array(text).max()

tes\_down = np.array(text).min()

te\_suiji\_xulie = np.random.randint(500,1000, 24)

tes\_max = list(tes).index(tes.max())

#te\_suiji\_xulie[random.randint(0,24)] = tes\_max

te\_suiji = te\_suiji\_xulie[random.randint(0,23)]

#三个突变值取最大值的两个周

tb = []

tb\_type = []

#for i in tubian\_id:

# addres = gys.index(i)

# tubian.append(np.array(biao31.loc[addres]))

for j in tubian\_id:

tb\_type.append(gyswz.loc[j]['材料分类'])

tp = np.linspace(0,0,24)

addres = np.random.randint(0,24, 2)

if (addres[0] == 0) or (addres[1] == 23):

tb.append(np.random.randint(2000,6000, 1))

else:

tb.append(0)

return wenzong,tb,np.array(te\_suiji),wenzong\_type,tb\_type,te\_type,[wending\_id,tubian\_id,teshu\_id]

#随机转运的转运商选择

def suiji\_zhuanyun\_3():

zy1 = []

for i in range(0,31):

zy1.append(int(np.random.randint(1,9,1)))

return zy1

min\_2 = []

dinggou\_list3 = []

zhunayun\_list3 = []

for mmm in range(8):

try:

#第一周

result1 = []

mubiao1\_list = []

P1 = []

for i in range(30):

suiji\_list = first\_suiji\_3()

count +=0

t2 = []

t2.append(int(suiji\_list[2]))

suiji\_data = suiji\_list[0]+suiji\_list[1]+t2

suiji\_type = suiji\_list[3]+suiji\_list[4]

suiji\_type.append(suiji\_list[5])

gh = pd.DataFrame([suiji\_type,suiji\_data]).T

A = gh[gh[0] == 'A'][1].values.sum()

B = gh[gh[0] == 'B'][1].values.sum()

C = gh[gh[0] == 'C'][1].values.sum()

if ((A+B+C) <= 48000) and ((A/0.6+B/0.66+C/0.72)>=(28200\*2)):

P1.append(A/0.6+B/0.66+C/0.72)

mubiao1 = 1.2\*A+1.1\*B+C

mubiao1\_list.append(mubiao1)

result1.append(suiji\_list)

else:

continue

addres = mubiao1\_list.index(min(mubiao1\_list)[0])

fliter = []

for i in result1[addres][0]:

fliter.append(int(i))

for i in result1[addres][1]:

fliter.append(i)

fliter.append(int(result1[addres][2]))

fliter.append(int(P1[addres]))

fliter.append(int(min(mubiao1\_list)))

fliter1 = fliter.copy()

#后面几周

#筛选对应材料的供货商

p = int(P1[addres])

fliter\_list = []

P\_list = [int(P1[addres])]

CT = 0

for i in range(2,25):

result2 = []

mubiao2\_list = []

Pn = []

p=P\_list[CT]

for i in range(150):

suiji\_list = suiji\_3()

count +=0

t2 = []

t2.append(int(suiji\_list[2]))

suiji\_data = suiji\_list[0]+suiji\_list[1]+t2

suiji\_type = suiji\_list[3]+suiji\_list[4]

suiji\_type.append(suiji\_list[5])

gh = pd.DataFrame([suiji\_type,suiji\_data]).T

A = gh[gh[0] == 'A'][1].values.sum()

B = gh[gh[0] == 'B'][1].values.sum()

C = gh[gh[0] == 'C'][1].values.sum()

if ((A+B+C) <= 48000) and ((A/0.6+B/0.66+C/0.72)>=(28200\*2)-p):

Pn.append(A/0.6+B/0.66+C/0.72)

mubiao2 = 1.2\*A+1.1\*B+C

mubiao2\_list.append(mubiao1)

result2.append(suiji\_list)

else:

continue

addres = mubiao2\_list.index(min(mubiao2\_list)[0])

fliter = []

for i in result2[addres][0]:

fliter.append(i[0])

for i in result2[addres][1]:

tmp = []

try:

for j in i:

if j != 0:

tmp.append(j[0])

else:

tmp.append(j)

for k in tmp:

fliter.append(k)

except:

fliter.append(i)

fliter.append(int(result2[addres][2]))

fliter.append(int(Pn[addres]))

fliter.append(int(min(mubiao2\_list)))

fliter\_list.append(fliter)

P\_list.append(int(Pn[addres]))

CT +=1

#整体订购方案

t = [fliter1]

for i in fliter\_list:

t.append(i)

ttp = []

for i in suiji\_list[-1]:

for j in i:

ttp.append(j)

ttp.append('原料对应产能')

ttp.append('原材料费用')

dinggoufangan=pd.DataFrame(t,columns=ttp,index=list(np.linspace(1,24,24))).T

t = []

for i in dinggoufangan.values:

ttttt = []

for j in i:

ttttt.append(int(j))

t.append(np.array(ttttt))

dinggoufangan3 = pd.DataFrame(np.array(t),index=dinggoufangan.index.values,columns=dinggoufangan.columns.values)

dinggou\_list3.append(dinggoufangan3)

#转运方案

t31 = dinggoufangan3[0:31]

result2 = []

for k in range(24):

index = list(dinggoufangan3[0:31].index.values)

ttpp = list(t31.iloc[:,k].values)

ss\_ghs = pd.DataFrame([index,ttpp]).T.sort\_values(by=1).iloc[::-1]

ghsmd = ss\_ghs[0].values #供货商名单

zysmdv = sh\_p[0].values #转运商名字

ghsmdv= ss\_ghs[1].values #供货商从大到小排的供货量

ghslist = []

for i in zysmdv:

tp = 0

ghsl = []

ct = 0

ghsmdv\_tp = []

for j in ghsmdv:

tp += j

if tp <= 6000:

ghsl.append(ghsmd[ct])

ghsmdv\_tp.append(j)

ct +=1

ghslist.append(ghsl)

ghsmd = [x for x in ghsmd if x not in ghsl]

zysmdv = zysmdv[1:]

ghsmdv = [x for x in ghsmdv if x not in ghsmdv\_tp]

count = 0

ys = pd.DataFrame(index = ss\_ghs[0].values,columns=[1,2,3,4,5,6,7,8])

count2 = 0

for i in ghslist:

for j in i:

ys.loc[j][(sh\_p[0].values)[count]] = ss\_ghs[1].values[count2]

count2 +=1

count +=1

result2.append(ys)

chushi = result2[0]

for i in result2[1:]:

chushi = pd.concat([chushi,i],axis=1)

zhunayun\_list3.append(chushi)

#chushi

#一次循环目标值

cs\_type = list(gyswz['材料分类'].values)

suihaobiao = []

for i in range(24):

suihaobiao.append(suiji\_sunhao())

ck = suihaobiao[0]

for j in suihaobiao[1:]:

ck = ck + j

roworder = 0

rowlist = []

for row in chushi.fillna(0).itertuples():

ttp = []

temp = np.array(row[1:])\*np.array(ck)+np.array(row[1:])

roworder+=1

rowlist.append(temp)

count = 0

tptp = []

for i in cs\_type:

if i == 'A':

tptp.append(rowlist[count].sum()\*1.2)

elif i == 'B':

tptp.append(rowlist[count].sum()\*1.1)

else:

tptp.append(rowlist[count].sum())

count+=1

min\_2.append(sum(tptp))

except:

pass

address = min\_2.index(min(min\_2))

dinggou\_list3[address].to\_csv('wenti3\_ghs.csv')

zhunayun\_list3[address].to\_csv('wenti3\_zys.csv')

#第四题

df4\_data = df1.sort\_values(by=[0],ascending=False)#评分表

tptp = []

for i in list(df4\_data.index):

tptp.append(pd.DataFrame(list(data2.iloc[i])).T)

dff4 = pd.concat(tptp)#评分表对应供货商数据

df4\_std = []

for i in np.array(dff4.loc[:,2:]):

df4\_std.append(np.array(i).std())

#df4\_data

#构建所有数据表

count = 0

tp = []

for i in dff4.values:

t = list(i)

t.append(df4\_data.values[count][0])

t.append(df4\_std[count])

count += 1

tp.append(t)

origin4 = pd.DataFrame(tp)

#data4 = origin4.loc[:,2:241]

#origin4

#筛选突变进行平稳化处理

x4 = np.linspace(0,402,402)

fc4 = origin4[243].values

adr = []

count = 0

for i in fc4:

if i > 1300:#取出突变型

adr.append(count)

count +=1

#max ---> rand(0.66m~0.33m)/(m/6000)

ttp = []

for i in adr:

ttp.append(origin4.iloc[i])

cn = pd.DataFrame(ttp).iloc[:,2:242]

roworder = 0

ttp = []

max\_l4 = []

for row in np.array(cn):

max\_l4.append(row.max())

cout = 0

tp = []

for row in np.array(cn):

ct = 0

for i in row:

if i > 6000:

row[ct] = random.uniform(0.6\*max\_l4[cout],max\_l4[cout])/(max\_l4[cout]/6000)

ct +=1

cout +=1

tp.append(row)

values\_t = origin4.iloc[:,2:242]

#values\_t

#处理后的数据表

count = 0

tpv = []

for i in np.array(values\_t):

ct = 0

if count in cn.index.values:

tpv.append(tp[ct])

ct +=1

else:

tpv.append(i)

count+=1

temp = pd.DataFrame(tpv,index=values\_t.index.values,columns=values\_t.columns.values)

#temp

ghs4 = origin4.copy()

#处理类型成本

Dtype402 = []

for i in np.array(ghs4):

Dtype402.append(i[1])

Ddata3 = [0.6,0.66,0.72]

D402 = []

for i in Dtype402:

if i == 'A':

D402.append(Ddata3[0])

elif i == 'B':

D402.append(Ddata3[1])

elif i == 'C':

D402.append(Ddata3[2])

else:

continue

#取每行平均值

t4 = ghs4.set\_index(0).drop([1],axis=1)

you0 = list((t4 == 0).astype(int).sum(axis=1))

sum\_t4 = list(t4.sum(axis=1).values)

fei0 = [240 - i for i in you0]

mean\_t4 = np.array(sum\_t4)/np.array(fei0)

index = list(t1.index.values)

pj\_values = list(mean\_t4\*1.14)

ghs4.iloc[:,242] = pj\_values

ghs4 = ghs4.sort\_values(by=242)[::-1]

#ghs4 = pd.concat([ghs4[ghs4[1] == 'A'],ghs4[ghs4[1] == 'B'],ghs4[ghs4[1] == 'C']])

#ghs4

sum\_cn = 0

ct = 0

count = 0

pjcn = []

ljcn = []

for i in pj\_values:

sum\_cn = sum\_cn + i

ljcn.append(sum\_cn)

pjcn.append(i)

if sum\_cn < 48000:

count+=1

ct+=1

continue

else:

break

gys = t4.index.values[0:len(pjcn)]

cpt = ghs4[1].values[0:len(gys)]

gyslim = pd.DataFrame([gys,cpt,pjcn,ljcn],index=['供应商ID','材料分类','供应商平均供应能力','累计供应能力']).T.set\_index('供应商ID')

#gyslim = pd.concat([gyslim[gyslim['材料分类'] == 'A'],gyslim[gyslim['材料分类'] == 'B'],gyslim[gyslim['材料分类'] == 'C']])

gyslim.to\_csv('重要参数4.csv')

#14/15 6/5

def onew():

while True:

one\_w = []

for i in pjcn:

a = random.uniform((i\*0.8),(i\*1.3))

one\_w.append(a)

if sum(one\_w) > 48000:

return one\_w

#概率表

tq = []

for i in range(24):

for j in suiji\_sunhao():

tq.append(j)

kb = []

#订购方案

tp = []

for i in range(24):

tp.append(onew())

t286 = pd.DataFrame(tp,columns=gyslim.index.values).T

t286.to\_csv('订购4.csv')

temp7 = pd.read\_csv('订购4.csv')

data3 = pd.read\_excel('附件2 近5年8家转运商的相关数据.xlsx')

temp6 = data3.set\_index('转运商ID')

tpq = []

for i in np.array(temp6):

ttp = []

for j in i:

if j != 0:

ttp.append(j\*0.01)

else:

continue

tpq.append(ttp)

ct = 1

for i in tpq:

x = np.linspace(0,len(i),len(i))

y = list(i)

fig = plt.figure(figsize=(8,11),dpi=98)

na=plt.subplot(4,1,1)

na.plot(x,y,label='转运商{}'.format(ct))

na.legend()

ct+=1

plt.show()

#fig.savefig('zhuanyun.jpg')

#计算转运商信息

tp = []

for i in np.linspace(1,8,8):

tp.append(int(i))

tpq\_mean = []

for i in tpq:

tpq\_mean.append(np.array(i).mean())

plt.figure(dpi=95)

plt.ylabel('平均转运损耗')

plt.xlabel('转运商')

plt.plot(tp,tpq\_mean)

plt.savefig('zhunayunpj.jpg')

plt.show()

sh\_p = pd.DataFrame([tp,tpq\_mean]).T.sort\_values(by=1)

#随机转运的转运商选择结果 24

def suiji\_zhuanyun(count):

zy1 = []

for i in range(0,count):

zy1.append(int(np.random.randint(1,9,1)))

return zy1

#随机损耗 1~8

def suiji\_sunhao():

ss = []

tpq\_suiji = []

for i in tpq:

tpq\_suiji.append(random.sample(list(np.array(i)),1)[0])

return tpq\_suiji

#对应的概率

sjz = suiji\_zhuanyun(23)

sh = suiji\_sunhao()

temp = []

for i in sjz:

temp.append(sh[i-1])

c=0

q = []

for i in temp7.values[0:-2]:#.sort\_values(by=k-1)[::-1][k-1]

t = list(i)

t.append(c)

q.append(t)

c+=1

temp1 = pd.DataFrame(q)

pm = sh\_p[0].values.copy()

tp111 = [] #记录选中供货商值

tp222 = [] #记录选中供货商和

tp333 = [] #记录供货商名字

tp444 = [] #记录转运商名字

#tp555 = 0 #记录检索到哪一位置供应商了

tp666 = [] #记录供货商供货量

for k in range(2,26):

ghs\_tp = temp1.sort\_values(k-1)[::-1].iloc[:,k-1]#供货商数据 从大到小

ghs\_mz = temp1.sort\_values(k-1)[::-1].iloc[:,0].values#供货商名字 从大到小

#zys\_tp = temp1.sort\_values(k-1)[::-1].iloc[:,-1]#对应损失率

tp11 = [] #记录选中供货商值

tp22 = [] #记录选中供货商和

tp33 = [] #记录供货商名字

tp44 = [] #记录转运商名字

#tp55 = 0 #记录检索到哪一位置供应商了

tp66 = [] #记录供货商供货量

sjz\_t = sjz.copy()

pm = sh\_p[0].values.copy()

for j in pm:

#try:

tp1 = [] #记录选中供货商值

tp2 = 0 #记录选中供货商和

tp3 = [] #记录供货商名字

tp4 = [] #记录转运商名字

#tp5 = 0 #记录检索到哪一位置供应商了

tp6 = [] #记录供货商供货量

count = 0

yz = []

for i in ghs\_mz:

try:

yz.append([i,ghs\_tp[count]])

except:

yz.append([i,0])

count+=1

for i in yz:

tp2 +=i[1]

if tp2 <= 6000:

tp3.append(i[0])

tp6.append(i[1])

else:

tp2 -=i[1]

tp5+=1

tp5 = j

ghs\_tp = [x for x in ghs\_tp if x not in tp6]

pm = pm[1:]

ghs\_mz = [x for x in ghs\_mz if x not in tp3]

tp11.append(tp1)

tp22.append([tp2])

tp33.append(tp3)

tp44.append(tp4)

#tp55.append([tp5])

tp66.append(tp6)

#except:

#continue

tp111.append(tp11)

tp222.append([tp22])

tp333.append(tp33)

tp444.append(tp44)

#tp555.append([tp55])

tp666.append(tp66)

kkk = []

for k in range(24):

pm = sh\_p[0].values.copy()

t = []

for j in pm:

tp = []

for i in range(8):

name = tp333[k][i]

shuju = tp666[k][i]

tp.append(pd.DataFrame([name,shuju],index=[0,j]).T.set\_index(0))

t.append(tp)#3 6 2 8 4 1 7 5

#0 1 2 3 4 5 6 7

kkk.append(pd.concat([t[0][5],t[0][2],t[0][0],t[0][4],t[0][7],t[0][1],t[0][6],t[0][3]],axis=1))

result = pd.concat(kkk,axis=1)

result.to\_csv('转运方案4终.csv')

#格式转换

temp7 = result.copy()

qm = data2.iloc[:,0].values

qm\_list = [x[1:] for x in qm]

k\_index = [x[1:] for x in qm if x not in temp7.index.values]

temp = pd.DataFrame(qm).set\_index(0)

y\_index = [x[1:] for x in temp7.index.values]

for i in range(192):

temp[i] = None

k = 0

tp = []

for i in qm\_list:

if i in y\_index:

tp.append(temp7.iloc[k,:])

k+=1

else:

tp.append(temp.iloc[int(i)-1,:])

tpp = []

for i in tp:

tpp.append(np.array(i))

temp = pd.DataFrame(tpp)

temp.to\_csv('转运方案4终终.csv')

#转换填表格式

temp7 = pd.read\_csv(r'订购4.csv')

t23 = temp7.iloc[0:135,:]

tp = []

for i in np.linspace(0,24,25):

tp.append(int(i))

t23.columns = tp

tp = []

for i in data1['供应商ID'].values:

c = [i]

for j in np.linspace(0,0,24):

c.append(int(j))

tp.append(c)

s402 = pd.DataFrame(tp)

ct = 0

for i in t23[0].values:

index\_23 = int(i[-3:])-1

s402.iloc[index\_23,:] = t23.iloc[ct,:]

ct += 1

s402.to\_csv(r'订购4终.csv')

type\_z = data1['材料分类'].values

#平均产能

data1 = pd.read\_excel(r'附件1 近5年402家供应商的相关数据.xlsx',sheet\_name='企业的订货量（m³）')

type\_z = data1['材料分类'].values

z4 = pd.read\_csv('订购4终.csv')

z4 = z4.drop(['Unnamed: 0'],axis=1)

z4 = z4.set\_index('0')

#z4 = z4.T

Dtype = type\_z

Ddata = [1/0.6,1/0.66,1/0.72]

D = []

for i in Dtype:

if i == 'A':

D.append(Ddata[0])

elif i == 'B':

D.append(Ddata[1])

elif i == 'C':

D.append(Ddata[2])

else:

continue

t = []

for i in range(24):

t.append(D)

#temp = /24

z4 = pd.DataFrame(np.array(z4)\*np.array(t).T)

res = z4.apply(lambda x: x.sum(),axis=0).T

res.to\_csv('重要数据4.csv')

print('第四问平均产能：{}'.format(res.sum()/24))