**匈牙利算法python代码：**

from scipy.optimize import linear\_sum\_assignment

cost =np.array([[4,1,3],[2,0,5],[3,2,2]])

row\_ind,col\_ind=linear\_sum\_assignment(cost)

print(row\_ind)#开销矩阵对应的行索引

print(col\_ind)#对应行索引的最优指派的列索引

print(cost[row\_ind,col\_ind])#提取每个行索引的最优指派列索引所在的元素，形成数组

print(cost[row\_ind,col\_ind].sum())#数组求和

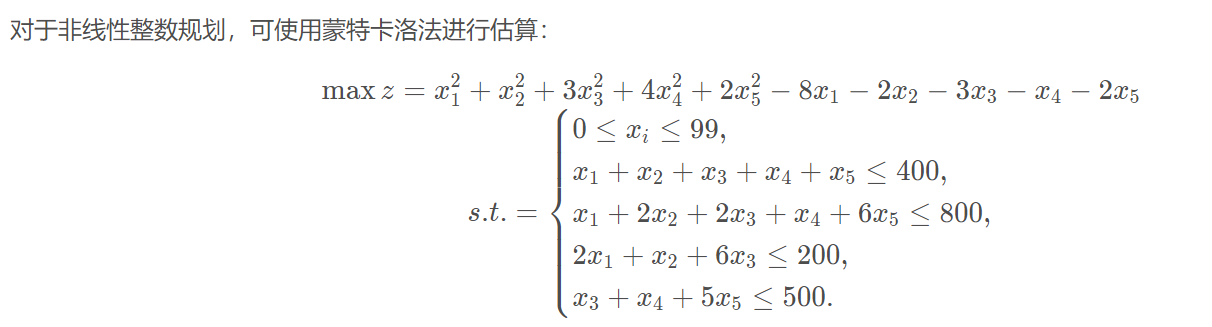
# 输出：

# [0 1 2]

# [1 0 2]

# [1 2 2]

# 5



求解的Python代码

import numpy as np

def check(x):

if x.sum() > 400:

return False

if x[0]+2\*x[1]+2\*x[2]+x[3]+6\*x[4] > 800:

return False

if 2\*x[0]+x[1]+6\*x[2]>200:

return False

if x[2]+x[3]+5\*x[3]>200:

return False

return True

def get\_radom():

x = np.random.randint(100, size=5)

while not check(x):

x = get\_radom()

return x

lim = 10\*\*6

ans = -1

for i in range(lim):

num = get\_radom()

ans = max(ans, num.all())

if i % 10000 == 0:

print(i)

print('ans=' + ans)