Задача: В городе площадью 1у.е. × 1у.е. 4 ресторана.

Необходимо расположить 2 продуктовых склада так, чтобы расстояние между ресторанами и складами было минимальным. Известны координаты ресторанов: $(x_1, y_1) \dots (x_4, y_4)$.

При этом склад 1 можно построить в 3й четверти „квадрата" города, а склад 2 в 1й четверти. Планировка города аналогична Манхэттенской.

Пусть склад 1 имеет координаты: (a_1, b_1)

склад 2 : (a_2, b_2)

d: distance

$$d(\text{склад}_i, \text{ресторан}_j) = |a_i - x_j| + |b_i - y_j|$$

Тогда
$$d = d(1,1) + \dots + d(1,4) + d(2,1) + \dots + d(2,4)$$
$$= |a_1 - x_1| + |b_1 - y_1| + \dots + |a_1 - x_4| + |b_1 - y_4| +$$
$$|a_2 - x_1| + |b_2 - y_1| + \dots + |a_2 - x_4| + |b_2 - y_4|$$

$$d \to \min_{a_1, b_1, a_2, b_2}$$

Пусть
$$|a_i - x_j| = m_{j+4I_{\{i=2\}}}$$
$$|b_i - y_j| = n_{j+4I_{\{i=2\}}}$$

$$\Rightarrow d = m_1 + n_1 + \dots + m_4 + n_4$$
$$+ m_5 + n_5 + \dots + m_8 + n_8$$

Если $|a_1 - x_1| = m_1$, то $\begin{cases} a_1 - x_1 \le m_1 \\ -a_1 + x_1 \le m_1 \end{cases}$

или $\begin{cases} a_1 - m_1 \le x_1 \\ -a_1 - m_1 \le -x_1 \end{cases}$

Таким образом, модель такая:

$$d = m_1 + \ldots + m_8 + n_1 + \ldots + n_8 \to \min_{\substack{a_1, b_1, m_1, \ldots, m_8 \\ a_2, b_2, n_1, \ldots, n_8}}$$

$$\begin{cases}
\begin{array}{ll}
a_1 - m_1 \le x_1 & b_1 - n_1 \le y_1 \\
\vdots & \vdots \\
a_1 - m_4 \le x_4 & b_1 - n_4 \le y_4 \\
-a_1 - m_1 \le -x_1 & -b_1 - n_1 \le -y_1 \\
\vdots & \vdots \\
-a_1 - m_4 \le -x_4 & -b_1 - n_4 \le -y_4 \\
a_2 - m_5 \le x_1 & b_2 - n_5 \le y_1 \\
\vdots & \vdots \\
a_2 - m_8 \le x_4 & b_2 - n_8 \le y_4 \\
-a_2 - m_5 \le -x_1 & -b_2 - n_5 \le -y_1 \\
\vdots & \vdots \\
-a_2 - m_8 \le -x_4 & -b_2 - n_8 \le -y_4
\end{array} \\
0 \le m_1, \ldots, m_8, n_1, \ldots, n_8 \le 1 \\
0 \le a_1, b_1 \le 0{,}5 \\
0{,}5 \le a_2, b_2 \le 1
\end{cases}$$

```
import scipy.optimize as opt
```

```
import numpy as np
```

```
coord = [
 [x1,y1],
 [x2,y2],
  [x3,y3],
  [x4,y4]
 ] = np.random.rand(4,2)
```

```
coord
```

```
     array([[0.04632109, 0.19339814],
            [0.23089857, 0.3883741 ],
            [0.25763443, 0.98877655],
            [0.00282756, 0.94341808]])
```

```
C = [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0]
```

```
lhs_ineq = [
            [-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0],
            [0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0],
            [0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0],
            [0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0],
            [-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0],
            [0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0],
            [0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0],
            [0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0],
            [0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0],
            [0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,1,0],
            [0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,1,0],
            [0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,1,0],
            [0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0],
            [0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,-1,0],
            [0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,-1,0],
            [0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,-1,0],
            [0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,1,0,0],
            [0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,1,0,0],
            [0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,1,0,0],
            [0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,1,0,0],
            [0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,-1,0,0],
            [0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,-1,0,0],
            [0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,-1,0,0],
            [0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,-1,0,0],
            [0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,1],
            [0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,1],
            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,1],
            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,1],
            [0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,-1],
            [0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,-1],
            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,-1],
            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,-1]
]


rhs_ineq = [x1,x2,x3,x4,-x1,-x2,-x3,-x4,x1,x2,x3,x4,-x1,-x2,-x3,-x4,y1,y2,y3,y4,-y1,-y2,-y3,-y4,y1,y2,y3,y4,-y1,-y2,-y3,-y4]


lhs_eq = np.zeros((1,20))


rhs_eq = [0]


bnds = [[0,1] for i in range(20)]


bnds[16:20] = [[0,0.5],[0,0.5],[0.5,1],[0.5,1]]


lin = opt.linprog(c=C, A_ub=lhs_ineq, b_ub=rhs_ineq,
               A_eq=lhs_eq, b_eq=rhs_eq, bounds=bnds,
              method="revised simplex")


print('distance: ',lin.fun)
print('solution found: ',lin.success)
print('variables optimal values: ',lin.x)
print('1st spot coordinates: (',lin.x[16],',',lin.x[17],')')
print('2nd spot coordinates: (',lin.x[18],',',lin.x[19],')')
```

```
distance:  4.602547460675344
solution found:  True
variables optimal values:  [0.         0.18457748 0.21131333 0.04349353 0.45367891 0.26910143
 0.24236557 0.49717244 0.19497596 0.         0.60040245 0.55504398
 0.75001993 0.55504398 0.04535847 0.         0.04632109 0.3883741
 0.5        0.94341808]
1st spot coordinates: ( 0.04632109282408681 , 0.38837410091720537 )
2nd spot coordinates: ( 0.5 , 0.9434180764696619 )
```

✓ 0 сек. выполнено в 22:08 ● ×