最优化算法中的粒子群算法。

总目标有两个：

一共有两个变量，也就是我们需要求解得到的值。

储备过量点9个，储备不足点3个

1. ：储备过量点*Si*的单位运输成本

表示储备过量点*Si*的单位运输成本，那么表示*S1*的单位运输成本

表示一类，然后从储备过量点到储备不足点

transCost = np.array([[[5, 5, 5],

[6, 6, 6],

[7, 7, 7],

[7, 7, 7],

[4, 4, 4],

[5, 5, 5],

[5, 5, 5],

[7, 7, 7],

[6, 6, 6]],

[[5, 5, 5],

[6, 6, 6],

[7, 7, 7],

[7, 7, 7],

[4, 4, 4],

[5, 5, 5],

[5, 5, 5],

[7, 7, 7],

[6, 6, 6]],

[[5, 5, 5],

[6, 6, 6],

[7, 7, 7],

[7, 7, 7],

[4, 4, 4],

[5, 5, 5],

[5, 5, 5],

[7, 7, 7],

[6, 6, 6]]

])

timeMatrix = np.array([[15, 10, 28],

[33, 16, 24],

[40, 24, 29],

[35, 30, 25],

[21, 15, 12],

[16, 21, 19],

[13, 23, 24],

[18, 29, 22],

[24, 35, 29]])

t\_ij

timeMatrix = np.array([[15, 10, 28],

[33, 16, 24],

[40, 24, 29],

[35, 30, 25],

[21, 15, 12],

[16, 21, 19],

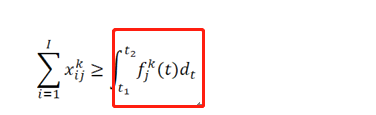
[13, 23, 24],

[18, 29, 22],

[24, 35, 29]])

下面的两个是同一个

C:\Users\CHENAN~1\AppData\Local\Temp\WeChat Files\5ea170b6911c017bf6931e12458c0e2.png

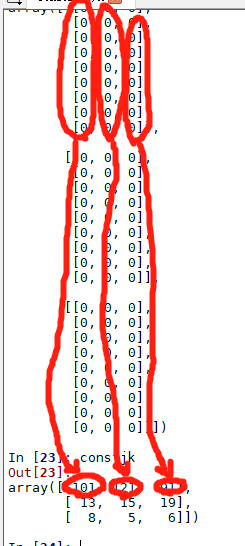


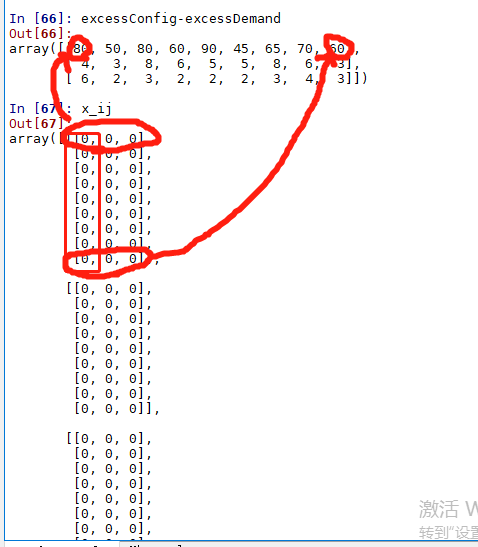
决策变量：

x\_{ij}^k 是要的输出

yi是是否启动某个过量点的0-1变量

利用传统PSO算法对算例进行求解





这个与随机性也有一定的影响。