# 1初始解生成算法——两阶段生成算法

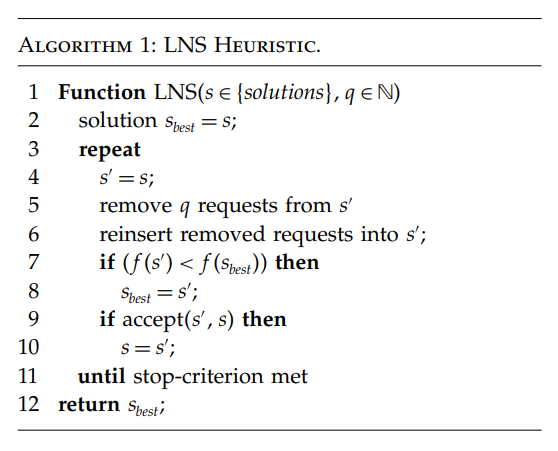
## 1.1第一阶段——最小化车辆的使用

（1）使用连续插入的方法生成初始解

（2）测试能否直接移除一条 route ，将其订单安排在其他route里，直至不能再直接移除route

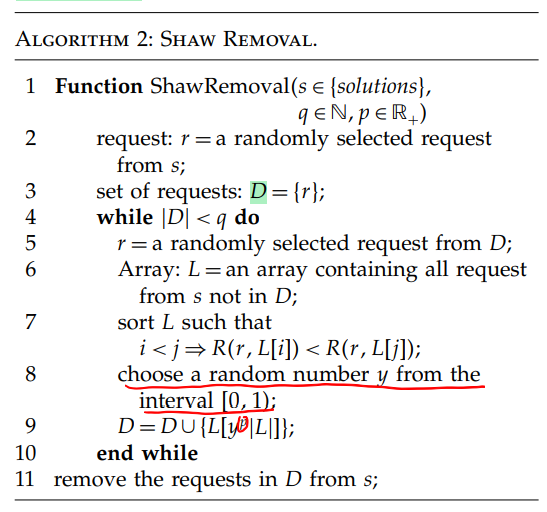
## 1.2第二阶段——最小化其他目标（车辆的总行驶距离等）

使用传统的LNS算法优化第二目标



# 2 需求移除算子 request removal

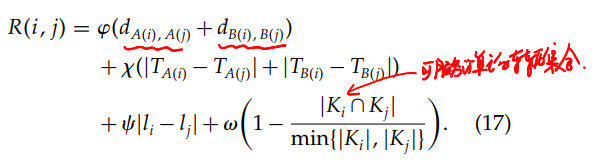
## 2.1 Shaw Removal Heuristic



（1）函数输入：一个可行解（solution），移除订单数量（q）,移除订单随机程度（p）

（2）函数输出：移除订单集合（request removed）,破坏后的解（destroyed solution）

（3）R(i,j)



A(i),B(i)表示request i的pickup节点和delivery节点

Ti 表示在可行解solution中节点i被访问的时间

li 为request i 的重量

Ki为可以服务request i 的车辆集合

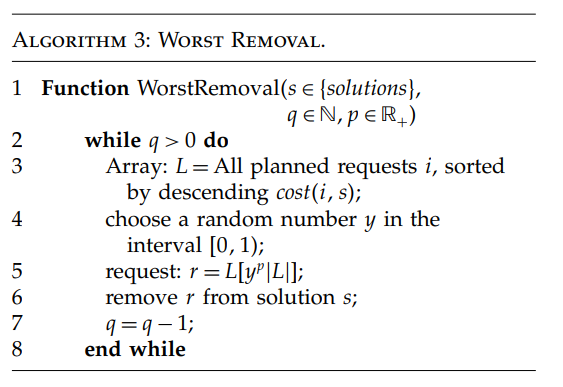
## 2.2 random removal

随机移除q个订单，可以认为是2.1中p=1的特例

（1）函数输入：一个可行解（solution），移除订单数量（q）

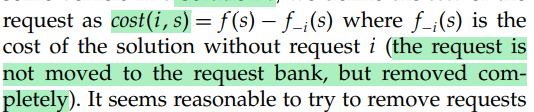
（2）函数输出：移除订单集合（request removed）,破坏后的解（destroyed solution）

2.3 worst removal



（1）函数输入：一个可行解（solution），移除订单数量（q）,移除订单随机程度（p）

（2）函数输出：移除订单集合（request removed）,破坏后的解（destroyed solution）

（3）

# 3 需求插入算子 insert requests

## 3.1 Basic Greedy Heuristic