
Algorithm 1: 网格划分

```
1 输入: Dataset={data_to_grid 被划分的数据集, (r,c) 网格大小}
2 过程: 函数 grid(Dataset)
3 根据 X、Y 极值分别生成间隔数为 r, c 的区间 X_range、Y_range
4 存储数据集的位置信息
5 dis_x = [], dis_y = []
6 for data_to_grid 的每一行 i do
7     for X_range 的每一个区间 j do
8         if data_to_grid 的第 i 行的 X 属于第 j 个区间 then
9             dis_x[i]=j
10        end if
11    end for
12    for Y_range 的每一个区间 k do
13        if data_to_grid 的第 i 行的 Y 属于第 k 个区间 then
14            dis_y[i]=k
15        end if
16    end for
17 end for
18 更新 data_to_grid 位置信息
19 data_to_grid['district_x'] = dis_x
20 data_to_grid['district_y'] = dis_y
21 输出: Dataset
```

Algorithm 2: 网格扩大化

```
1 输入: iter_num 网格扩大次数, gridshape 网格参数, f(gridshape, iter_num) 网格扩大函数
2 过程: 函数 enlarge_gridshape(iter_num)
3 if iter_num = 0 then
4   return 网格参数 gridshape
5 else
6   if 所有订单均已分配 then
7     size = (1, 1)
8     return size
9   else
10    确保网格能够平滑扩大:
11    r, c = f(gridshape, iter_num)
12    if  $r \leq 0$  then
13      r = 1
14    end if
15    if  $c \leq 0$  then
16      c = 1
17    end if
18    size = (r, c), return size
19  end if
20 end if
21 输出: size 网格参数
```

Algorithm 3: 递归求解

```
1 输入: Dataset={aunt 待分配阿姨数据集,order 待分配订单数据集,n 递归深度,status 求解状态,n_max 最大递归深度}
2 初始值: n = 1, status = True
3 过程: 函数 solver(Dataset)
4  $a_n, o_n$  分别表示待分配阿姨和订单的个数
5 使用 cvxpy 库对 MIP 问题求解得到原问题 prob 和解 x
6  $k = \sum_{i=1}^{a_n} \sum_{j=1}^{o_n} x_{ij}$ 
7 if status then
8     如果求解状态开启
9     if prob 存在最优解 and  $n \geq 1$  then
10         if  $n > n\_max$  then
11             防止无限递归
12             return None, 0, 0
13         end if
14         递归深度改变, 其他参数不变
15         prob_1, x_1, k_1 = solver(*, n+1, *)
16         if prob_1 = None then
17             return prob, x, k
18         else
19             if prob_1 的目标值 > prob 的目标值 then
20                 return prob_1, x_1, k_1
21             else
22                 return prob, x, k
23             end if
24         end if
25     end if
26     if prob 无可行解 and  $n > 1$  then
27         return None, 0, 0
28     end if
29     if prob 无可行解 and  $n = 1$  then
30         求解状态改变, 其他参数不变
31         prob_1, x_1, k_1 = solver(*, status = False, *)
32         return prob_1, x_1, k_1
33     end if
34 else
35     return prob, x, k
36 end if
37 输出: prob 问题参数,x 解矩阵,k 分配订单个数
```
