

Algorithm 1: solve

1 begin

2 从求解器里将求解值赋值给参数

3 objective ← cp.Maximize(obj), prob ← cp.Problem(objective, constrains)

4 prob.solve(solver←cp.GLPK_MI, qcp←True), df ← pd.DataFrame(x.value) 如果执行状态为 True

5 if status then

6 如果是最优解并且优先级标准数大于等于 1 if prob.status == 'optimal' and n ≥ 1 then

7 如果优先级标准数已经大于优先级排名的最大数

8 if n > max(rank) then

9 return None, 0, 0

10 end if

11 递归调用迭代，把解矩阵传给临时解

12 n ← n+1, prob_1, df_1, m_1 solver(aunt, order, timestamp, solver_mode, n, True, high_quality_aunt_id) 如果临时解为空则返回原矩阵解

13 if prob_1 == None then

14 return prob, df, np.sum(x.value)

15 else

16 如

17 end if

18 果临时解目标值大于原解优化值，则把临时解矩阵作为答案矩阵返回

19 if prob_1.value > prob.value then

20 return prob_1, df_1, m_1

21 else

22 如

23 end if

24 果原矩阵目标值大于临时解，则把原矩阵解作为答案返回

25 return prob, df, np.sum(x.value)

26 end if

27 如果无可行解，并且优先级标准数大于 1 则舍弃该方案

28 if prob.status == 'infeasible' and n > 1 then

29 return None, 0, 0

30 end if

31 如果无可行解，并且优先级标准数等于 1，则警告后将执行状态置为 False，然后进行再次迭代把解传给临时解并返回

32 if prob.status == 'infeasible' and n == 1 then

33 warnings.warn, prob_1, df_1, m_1 ← solver(aunt, order, timestamp, solver_mode, n, False, high_quality_aunt_id)

34 return prob_1, df_1, m_1

35 end if

36 else

37 返

38 end if

39 回原解矩阵 return prob, df, np.sum(x.value)

40 end

Algorithm 2: solve

```
1 begin
2   从求解器里将求解值赋值给参数
3   objective ← cp.Maximize(obj), prob ← cp.Problem(objective, constrains)
4   prob.solve(solver←cp.GLPK_MI, qcp←True), df ← pd.DataFrame(x.value)
5   if 执行状态为 True then
6     if 是最优解并且优先级标准数大于等于 l then
7       if 优先级标准数已经大于优先级排名的最大数 then
8         | 舍弃该方案
9       end if
10      递归调用迭代，把解矩阵传给临时解
11      n ← n+1, prob_1, df_1, m_1 solver(aunt, order, timestamp, solver_mode, n, True, high_quality_aunt_id)
12      if 临时解为空 then
13        | 返回原矩阵解
14      else
15        if 临时解目标值大于原解优化值 then
16          | 把临时解矩阵作为答案矩阵返回
17        else
18          | 如
19        end if
20        果原矩阵目标值大于临时解
21        把原矩阵解作为答案返回
22      end if
23    end if
24    if 无可行解，并且优先级标准数大于 l then
25      | 舍弃该方案
26    end if
27    则进行再次迭代把解传给临时解并返回
28    if 无可行解，并且优先级标准数等于 l then
29      | 发出警告并且将执行状态置为 False 再次递归迭代，然后返回迭代值
30    end if
31  else
32  end if
33  返回原解矩阵
34 end
```
