数值最优化课程 project
车辆路径规划中的分配问题

晏浩洋　徐荻

2021 年 1 月 3 日

1　前言

本篇 report 附件 *code.zip* 中含有 10 个文件

vrp_task.txt 为源数据文件，*data.mat* 为预处理后的 mat 数据文件，*taxi.m*，*solver*1.*m*，*solver*2.*m* 为 matlab 代码文件,*solution*1.*mat*，*solutionh*5.*mat*，*solutionh*7.*mat*，*solutionh*9.*mat*，*solutionh*11.*mat* 为求解结果文件

taxi.m 可以将源数据的 txt 文件预处理为 0-1 变量储存为 mat 形式，方便后续求解，输出结果即为 *data.mat*

*solver*1.*m* 是利用 matlab 自带整数型线性规划求解函数 *intlinprog* 构建的求解器

*solver*2.*m* 是只利用 matlab 线性规划求解函数 *linprog* 和 *branch & bound*(分支定界法) 思路构建的求解器，为本篇 report 的主要介绍部分，其中分支层数 h 可以作为参数根据实际情况调整，h 越大，求解结果越精确，求解速度越慢

*solution*1.*mat* 为在 *data.mat* 基础上 *solver*1.*m* 的求解结果，其中变量 *opt* 为 120*1 向量，即对应 120 个 task 求得的最低成本，变量 x 为 120*1 元胞，每个元胞内为长度为 Scores 的仅含 0,1 的向量，即对应 task 的最优方案，1 即代表该行被采纳到最优方案中

*solutionh*5.*mat*，*solutionh*7.*mat*，*solutionh*9.*mat*，*solutionh*11.*mat* 为在 *data.mat* 基础上 *solver*2.*m* 取分支层数 $h = 5, 7, 9, 11$ 时对应的求解结果，变量 *opt*，变量 x 含义同上

2　数据预处理

本部分对应代码文件 *taxi.m*

对于源数据中总共 120 个 task，每一个 task 创建对应 Sccores*3 的元胞，3 列信息分别代表订单 ID，司机 ID，分配成本，依次命名为 O，V，c。根据 Orders 和 Vehicles 数量分别构建 Orders*1 和 Vehicles*1 的全零向量，赋值给每一行的 O，V

利用 $fget1$ 函数逐行读取源数据，将此行 O，V 的第订单 ID，司机 ID 个元素修改为 1，即表示该行对应的订单和司机

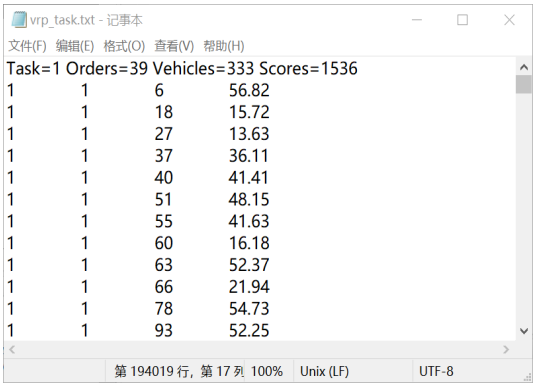
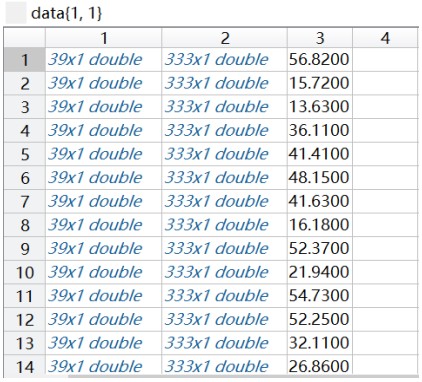


图 1: 源数据文件 *vrp_task.txt* 格式



	1	2	3	4
1	*39x1 double*	*333x1 double*	56.8200	
2	*39x1 double*	*333x1 double*	15.7200	
3	*39x1 double*	*333x1 double*	13.6300	
4	*39x1 double*	*333x1 double*	36.1100	
5	*39x1 double*	*333x1 double*	41.4100	
6	*39x1 double*	*333x1 double*	48.1500	
7	*39x1 double*	*333x1 double*	41.6300	
8	*39x1 double*	*333x1 double*	16.1800	
9	*39x1 double*	*333x1 double*	52.3700	
10	*39x1 double*	*333x1 double*	21.9400	
11	*39x1 double*	*333x1 double*	54.7300	
12	*39x1 double*	*333x1 double*	52.2500	
13	*39x1 double*	*333x1 double*	32.1100	
14	*39x1 double*	*333x1 double*	26.8600	

图 2: 预处理后的数据格式

3 建模

原问题即可表述为

$$\begin{aligned} min \quad & c^T x \\ s.t. \quad & o^T x = 1 \quad \forall o \in O \\ & v^T x \leq 1 \quad \forall v \in V \\ & x \in \{0,1\} \quad \forall x \end{aligned} \tag{1}$$

4 整数型线性规划求解器

本部分对应代码文件 *solver*1.*m*

利用 matlab 自带整数型线性规划求解函数 *intlinprog*，可直接对原问题求解

求解总耗时 5.074s，调用 *intlinprog* 函数 120 次，平均每个 task 耗时 0.042s，除 task109 无解之外，其余均求得最优解

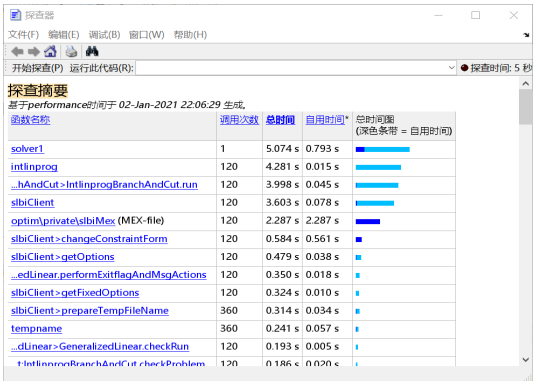
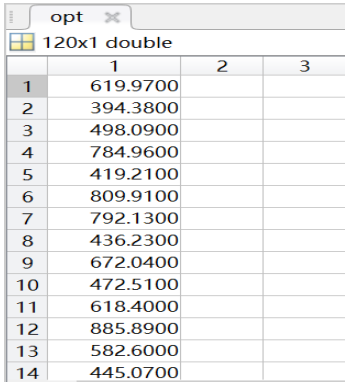


图 3: 整数型线性规划求解器耗时



	1	2	3
1	619.9700		
2	394.3800		
3	498.0900		
4	784.9600		
5	419.2100		
6	809.9100		
7	792.1300		
8	436.2300		
9	672.0400		
10	472.5100		
11	618.4000		
12	885.8900		
13	582.6000		
14	445.0700		

图 4: 整数型线性规划求解器输出结果

5 分支定界求解

本部分对应代码文件 $solver2.m$

由于本次 project 要求不能使用现有整数型线性规划求解器，因此尝试采用分支定界法结合已有的线性规划函数 $linprog$ 自行编写求解器

本求解器总体思路为：先将原问题中的整数约束 $x \in \{0,1\}$ 松弛为 $0 \leq x \leq 1$, 求解该线性规划问题，对于解中非整数的 x_i，分别讨论其为 0 和 1 的情况，继续求解线性规划，如果该子问题无解或解超出已有最优值，则放弃该支；如果有解但依然无整数解，则继续分支讨论；如果出现整数解且该解对应成本小于已有最优值，则更新最优值

该求解器伪代码如下，具体代码请参考代码文件 $solver2.m$

```
对于每一个task
    松弛掉原问题的整数约束，求解线性规划问题
    判定，如果无解，则终止
    对于解与0或1相差小于10^−5的，近似为0,1
    找到解中的非整数的位置及个数，命名为undef和noun
    如果noun=0，即为整数解，最优解即为该解
    否则，将undef的第一位放进def，将最优值设为正无穷
    创建5棵高度为h的空的二叉树，即长度为2^h−1向量，对应编号从上到下，从左到右排列，命名及含义分别为
        issol，是否有解树，0表示无整数解，1表示有整数解，2表示无解，3表示解超过最优值上界
        def，已确定订单编号树(仅在无整数解节点赋值)
        xdef，已确定订单取0还是1树(在所有子节点赋值)
        opti，最优值树(仅在有整数解节点赋值)
        xxi，解树(在有解节点赋值)
    对于每一个不是根节点的节点k
        判定其父节点编号
        如果issol(父节点)=0
            如果该子节点为其父节点的左儿子,则不采纳其父节点的undef
                其xdef在父节点基础上增加元素0
                删掉O，V，c对应的def(父节点)列
                对于约束右边的1，减去O或V的第(def(父节点)∗xdef(k))
                求解该线性规划子问题
                判定，如果无解，则终止此支
                对于解与0或1相差小于10^−5的，近似为0,1
                找到解中的非整数的位置及个数，命名为undef和noun
                将已确定的def插入回本次整数解,得到xxi(k)
                该方案成本opti(k)=c∗xxi(k)
                如果该方案成本大于已有最优值，则终止此支
                如果该方案为整数解
                    如果该方案成本小于已有最优值
                        更新已有最优值，最优解
                    终止此支
                如果该方案不为整数解
                    将undef第一位放进def(注意该处undef坐标是在已删除def列之后的，要换回原来坐标)
                    继续分支
            如果该子节点为其父节点的右儿子,则采纳其父节点的undef
                其xdef在父节点基础上增加元素1
                其余同上
```

取分支层数 $h = 5, 7, 9, 11$ 时分别测试该求解器，结果如下

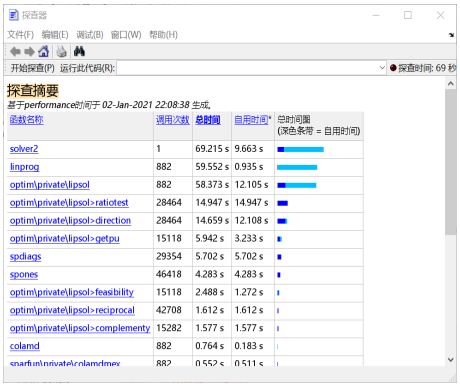


图 5: 分支定界法，h=5

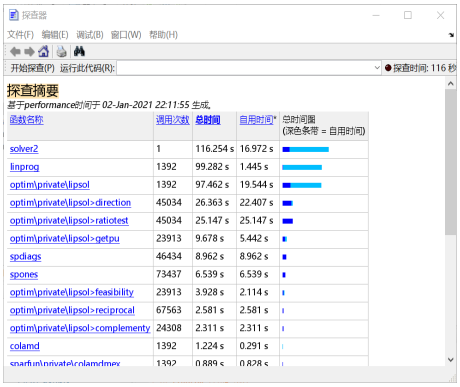


图 6: 分支定界法，h=7

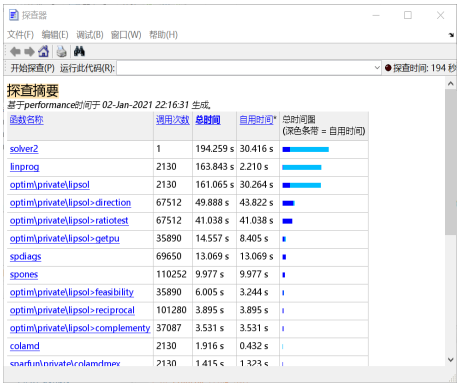


图 7: 分支定界法，h=9

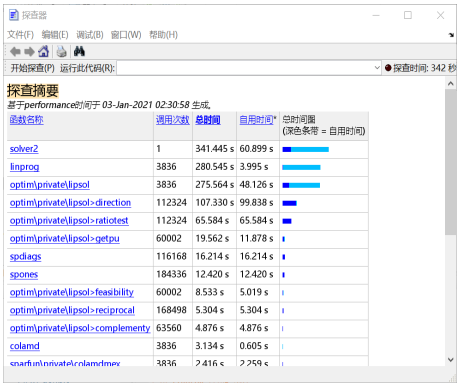


图 8: 分支定界法，h=11

将分支定界法求解结果与整数型线性规划求解器求解结果对比

h=5 时，求解总耗时 69.215s，调用 *linprog* 函数 882 次，平均每个 task 耗时 0.577s，120 个 task 中 3 个无解，其余 117 个 task 中 99 个求得最优解，平均每个 task 成本高出最优值 0.323 元

h=7 时，求解总耗时 116.254s，调用 *linprog* 函数 1392 次，平均每个 task 耗时 0.969s，120 个 task 中 1 个无解 (与整数型线性规划求解器结果相同)，其余 119 个 task 中 108 个求得最优解，平均每个 task 成本高出最优值 0.259 元

h=9 时，求解总耗时 194.259s，调用 *linprog* 函数 2130 次，平均每个 task 耗时 1.619s，120 个 task 中 1 个无解 (与整数型线性规划求解器结果相同)，其余 119 个 task 中 115 个求得最优解，平均每个 task 成本高出最优值 0.040 元

h=11 时，求解总耗时 341.445s，调用 *linprog* 函数 3836 次，平均每个 task 耗时 2.845s，120 个 task 中 1 个无解 (与整数型线性规划求解器结果相同)，其余 119 个 task 中 117 个求得最优解，平均每个 task 成本高出最优值 0.023 元