

Activity Based VRP With Automatic Link and Node Generation

Python 使用说明

2019.05

目录

1、问题说明.....	2
2、问题的底层网络结构.....	2
3. 模型说明.....	3
4. 输入说明.....	4
5. 输出说明.....	9
6. 程序操作说明.....	9

1、问题说明

本算法适用于两辆车辆基于乘客活动时间窗的 VRP 问题，将问题分解为乘客子问题和车辆子问题，基于商攀学长 space_time_network code，利用拉格朗日松弛算法和动态规划求解。

输入：乘客网中的节点信息、弧信息和乘客活动信息

车辆网中的节点信息、弧信息和车辆活动信息

输出：乘客和车辆的时空路径，上下界 GAP，乘客上界以及 volume 矩阵

约束：流平衡约束

目标：活动 benefit 最大

2、问题的底层网络结构

(1) 网络的节点

网络中有几种类型的节点：运输（物理）节点，乘客接、送节点以及车辆的车站节点。

乘客的时间窗表示在接、送节点（乘客的起始地和目的地）旁边的括号中。例如，接乘客 1 的时间窗是 4-7，那就是说，车辆必须在[4,7]这一时间段内接到乘客 1，否则乘客会取消这次服务。同样，车辆必须在[11,14]之间将乘客 1 送到节点 d_1 。

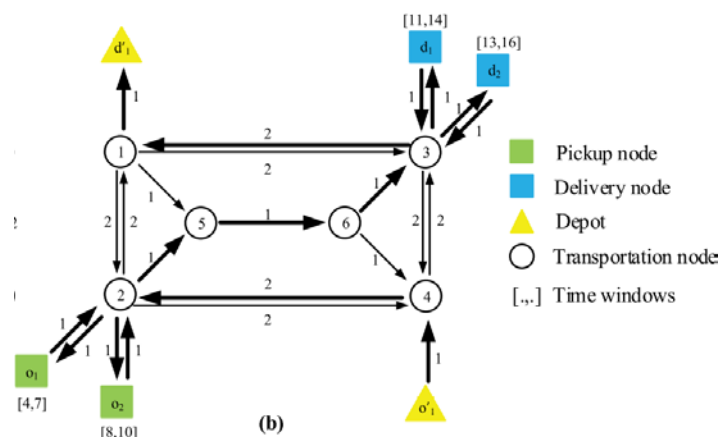


图 1 VRPPDTW 网络的节点

(2) 网络的弧

对于乘客子网而言，弧可分类 2 类：只有乘客能走的弧以及车和乘客都可以走的弧。

对于车辆子网而言，弧可以分为两类，只有车能走的弧以及车和乘客都可以走的弧。

此外弧也被分为等待弧、入库出库弧以及运输弧。

在运输弧上，cost 的更新与拉格朗日乘子有关，在其他两个弧上，cost 的更新与拉格朗日乘子无关。

需要额外注意的是，在该问题中我们不考虑运输的费用，因此在车辆子问题中 cost 用于描述车辆在某弧段上的 benefit。

3. 模型说明

Objective function:

$$\text{Min } Z = \sum_a \sum_{(i,j,t,t') \in E_{pa}} (c_{i,j,t,t'} \times x_{i,j,t,t'}^a) \quad (1)$$

Subject to:

Passenger flow balance constraint:

$$\sum_{i,t:(i,j,t,t') \in E_{pa}} x_{i,j,t,t'}^a - \sum_{i,t:(j,i,t',t) \in E_{pa}} x_{j,i,t',t}^a = \begin{cases} 1 & j = o(a), t' = DT(a) \\ -1 & j = d(a), t' = T \\ 0 & \text{otherwise} \end{cases} \quad \forall a \in A \quad (2)$$

Vehicle flow balance constraint:

$$\sum_{i,t:(i,j,t,t') \in E_{tr}} y_{i,j,t,t'}^q - \sum_{i,t:(j,i,t',t) \in E_{tr}} y_{j,i,t',t}^q = \begin{cases} 1 & j = o(q), t' = 1 \\ -1 & j = d(q), t' = T \\ 0 & \text{otherwise} \end{cases} \quad \forall q \in Q \quad (3)$$

Passenger flow and vehicle flow coupling constraint:

$$\sum_a x_{i,j,t,t'}^a \leq \sum_q y_{i,j,t,t'}^q \times Cap \quad \forall (i,j,t,t') \in E_{pa}^s \quad (4)$$

Minimum headway constraint for dispatching adjacent vehicles:

$$\sum_{(i,j,\tau,\tau') \in \Phi(i,j,t,t')} \sum_q y_{i,j,\tau,\tau'}^q \leq 1 \quad \forall (i,j,t,t') \in E_{tr}^s \quad (5)$$

Binary variable definition:

$$x_{i,j,t,t'}^a \in \{0, 1\} \quad (6)$$

$$y_{i,j,t,t'}^q \in \{0, 1\} \quad (7)$$

对约束 4 进行拆分得到约束 8 和 9:

$$x_{i,j,t,t'}^a \leq \sum_q y_{i,j,t,t'}^q \quad \forall a \in A, \forall (i,j,t,t') \in E_{pa}^s \quad (8)$$

$$\sum_a x_{i,j,t,t'}^a \leq Cap \quad \forall (i,j,t,t') \in E_{pa}^s \quad (9)$$

把约束 8 这一耦合约束利用拉格朗日松弛将原问题分解成两个问题:

问题 1 是针对人（把约束 9 也进行松弛）：

Objective function:

$$\begin{aligned} \text{Min } L_X = & \sum_a \sum_{(i,j,t,t') \in E_{pa}} \{c_{i,j,t,t'} \times x_{i,j,t,t'}^a\} + \sum_a \sum_{(i,j,t,t') \in E_{pa}^s} \{\pi_{i,j,t,t'}(a) \times x_{i,j,t,t'}^a\} + \\ & \sum_{(i,j,t,t') \in E_{pa}^s} \{\lambda_{i,j,t,t'} \times \sum_a x_{i,j,t,t'}^a\} = \sum_a \sum_{(i,j,t,t') \in E_{pa}} \{\hat{c}_{i,j,t,t'}^a \times x_{i,j,t,t'}^a\} \end{aligned} \quad (10)$$

Subject to:

Passenger flow balance constraint:

$$\sum_{i,t:(i,j,t,t') \in E_{pa}} x_{i,j,t,t'}^a - \sum_{i,t:(j,i,t',t) \in E_{pa}} x_{j,i,t',t}^a = \begin{cases} 1 & j = o(a), t' = DT(a) \\ -1 & j = d(a), t' = T \\ 0 & \text{otherwise} \end{cases} \quad \forall a \in A \quad (2)$$

$$x_{i,j,t,t'}^a \in \{0, 1\} \quad (6)$$

其中：

$$\hat{c}_{i,j,t,t'}^a = \begin{cases} c_{i,j,t,t'} + \pi_{i,j,t,t'}(a) + \lambda_{i,j,t,t'} & \forall (i,j,t,t') \in E_{pa}^s \\ c_{i,j,t,t'} & \forall (i,j,t,t') \in E_{pa}^w \cup E_{pa}^d \end{cases}$$

在求解这个问题 1 时用的是拉格朗日松弛，有两个拉格朗日乘子

问题 2 是针对车（对约束 5 进行增广拉格朗日，然后运用 ADMM 求解）：

Objective function:

$$\text{Min } L_Y(q) = \sum_{(i,j,t,t') \in E_{tr}} \hat{c}_{i,j,t,t'}^q \times y_{i,j,t,t'}^q \quad (17)$$

Subject to: Eqs. (3) and (7), where the general arc cost $\hat{c}_{i,j,t,t'}^q$ is defined by Eq. (15), and $y_{i,j,t,t'}^q$

is defined by Eq. (19).

$$\hat{c}_{i,j,t,t'}^q = \begin{cases} 0 & (i,j,t,t') \in E_{tr}^c \cup E_{tr}^h \cup E_{tr}^t \\ -\sum_a \pi_{i,j,t,t'}(a) + \mu_{i,j,t,t'} + \rho \{y_{i,j,t,t'}^q - 1/2\} & (i,j,t,t') \in E_{tr}^s, y_{i,j,t,t'}^q \neq 0 \\ -\sum_a \pi_{i,j,t,t'}(a) + \mu_{i,j,t,t'} & (i,j,t,t') \in E_{tr}^s, y_{i,j,t,t'}^q = 0 \end{cases} \quad (18)$$

$$y_{i,j,t,t'}^q = \sum_{(i,j,t,t') \in \Phi(i,j,\tau,\tau')} \sum_{q' \in Q/\{q\}} y_{i,j,t,t'}^{q'} \quad (19)$$

4. 输入说明

在该程序中，对于乘客的接送弧和虚拟接送节点采用自动生成的方式。由此数据分为初始输入数据，和运行后写入的数据。

①乘客

Table 1.Input passenger

passenger	from_node	to_node	departure	arrival	volume	available	initial	travel	travel_budget	departure	arrival	time_window
1	2	3	6	12		1 1;2;3;4;5	1		6	3	3	
2	2	3	9	15		1 1;2;3;4;5	1		6	2	3	

分别对应:

乘客ID	起点	终点	出发时间	到达时间	乘客数量	可行点	初始价值	预计旅行时间
------	----	----	------	------	------	-----	------	--------

针对乘客起点终点的修改放在了程序中，可行点 list 中添加虚拟节点也同样在程序中实现

②乘客弧

Table 2.original Input passenger link

link_id	from_node	to_node_id	type	cost	travel_time	coupled_train_link_id
1	1	2	1	0	2	1
2	2	1	1	0	2	2
3	1	3	1	0	2	3
4	3	1	1	0	2	4
5	2	4	1	0	2	5
6	4	2	1	0	2	6
7	2	5	1	0	1	7
8	3	4	1	0	2	8
9	4	3	1	0	2	9
10	1	5	1	0	1	10
11	5	6	1	0	1	11
12	6	3	1	0	1	12
13	6	4	1	0	1	13
14	1	1	2	0	1	-1
15	2	2	2	0	1	-1
16	3	3	2	0	1	-1
17	4	4	2	0	1	-1
18	5	5	2	0	1	-1
19	6	6	2	0	1	-1

分别对应:

弧ID	弧起点	弧终点	弧类型	费用	旅行时间	是否车人共用
-----	-----	-----	-----	----	------	--------

其中:

类型 1 表示：运输弧，类型 2 表示等待弧

是否车人共用中-1 表示仅人使用，其他数值表示车人共有

在运行过程中，虚拟弧的添加被写入 input link 文件

Table 3. revised Input passenger link

link_id	from_node	to_node_id	type	cost	travel_time	coupled_train_link_id
1	1	2	1	0	2	1
2	2	1	1	0	2	2
3	1	3	1	0	2	3
4	3	1	1	0	2	4
5	2	4	1	0	2	5
6	4	2	1	0	2	6
7	2	5	1	0	1	7
8	3	4	1	0	2	8
9	4	3	1	0	2	9
10	1	5	1	0	1	10
11	5	6	1	0	1	11
12	6	3	1	0	1	12
13	6	4	1	0	1	13
14	1	1	2	0	1	-1
15	2	2	2	0	1	-1
16	3	3	2	0	1	-1
17	4	4	2	0	1	-1
18	5	5	2	0	1	-1
19	6	6	2	0	1	-1
1002	501	2	100	0	1	1002
1012	2	501	100	0	1	1012
1052	501	501	2	0	1	1052
1003	3	601	200	0	1	1003
1013	601	3	200	0	1	1013
1053	601	601	2	0	1	1053
2002	502	2	100	0	1	2002
2012	2	502	100	0	1	2012
2052	502	502	2	0	1	2052
2003	3	602	200	0	1	2003
2013	602	3	200	0	1	2013
2053	602	602	2	0	1	2053

其中：

新增乘客 1 的接弧 1002、1012，送弧 1003、1013；乘客 2 接弧 2002、2012，送弧 2003、2013，其中四个虚拟节点新增等待弧 1052、1053、2052、2053。

命名规则为：乘客编号*1000+乘客虚拟节点所连接的节点编号

乘客编号*1000+10+乘客虚拟节点所连接的节点编号

等待弧：乘客编号*1000+50+乘客虚拟节点所连接的节点编号

Type 100 表示接乘客 200 表示送乘客 2 表示等待弧

③乘客节点

Table 3.Input passenger node

node_id	description
1	station1
2	station2
3	station3
4	station4
5	station5
6	station6

Table 4. Revised Input passenger node

node_id	description
1	station1
2	station2
3	station3
4	station4
5	station5
6	station6
501	
601	
502	
602	

其中

新增乘客 1 接节点 501，乘客 1 送节点 601，乘客 2 接节点 502，乘客 2 送节点 602

④车辆

Table 5.Input train

train_id	from_node	to_node_id	departure	arrival_t	volume	available_node_lsit
1	4	1	1	20	1	1;2;3;4;5;6
2	4	1	1	20	1	1;2;3;4;5;6

同理对于车辆起终点的修改和可行节点 list 的更新在程序中实现

⑤车辆弧

Table 6.Input train link

link_id	from_node	to_node_id	type	cost	travel_time	coupled_passenger_link_id
1	1	2	1	0	2	1
2	2	1	1	0	2	2
3	1	3	1	0	2	3
4	3	1	1	0	2	4
5	1	5	1	0	1	5
6	2	4	1	0	2	6
7	4	2	1	0	2	7
8	2	5	1	0	2	8
9	3	4	1	0	2	9
10	4	3	1	0	2	10
11	5	6	1	0	1	11
12	6	3	1	0	1	12
13	6	4	1	0	1	13
14	1	1	2	0	1	-1
15	2	2	2	0	1	-1
16	3	3	2	0	1	-1
17	4	4	2	0	1	-1
18	5	5	2	0	1	-1
19	6	6	2	0	1	-1

type 1 表示：运输弧，类型 2 表示等待

coupled_passenger_link_id 中-1 表示仅车使用，其他数值表示车人共有

Table 7.Revised Input train link

link_id	from_node	to_node_id	type	cost	travel_time	coupled_passenger_link_id	link_id
1	1	2	1	0	2	1	
2	2	1	1	0	2	2	
3	1	3	1	0	2	3	
4	3	1	1	0	2	4	
5	1	5	1	0	1	5	
6	2	4	1	0	2	6	
7	4	2	1	0	2	7	
8	2	5	1	0	2	8	
9	3	4	1	0	2	9	
10	4	3	1	0	2	10	
11	5	6	1	0	1	11	
12	6	3	1	0	1	12	
13	6	4	1	0	1	13	
14	1	1	2	0	1	-1	
15	2	2	2	0	1	-1	
16	3	3	2	0	1	-1	
17	4	4	2	0	1	-1	
18	5	5	2	0	1	-1	
19	6	6	2	0	1	-1	
1002	501	2	100	0	1	1002	
1012	2	501	100	0	1	1012	
1052	501	501	2	0	1	1052	
1003	3	601	200	0	1	1003	
1013	601	3	200	0	1	1013	
1053	601	601	2	0	1	1053	
2002	502	2	100	0	1	2002	
2012	2	502	100	0	1	2012	
2052	502	502	2	0	1	2052	
2003	3	602	200	0	1	2003	
2013	602	3	200	0	1	2013	
2053	602	602	2	0	1	2053	
10004	701	4	1000	0	1	10004	
10014	4	701	1000	0	1	10014	
10054	701	701	2	0	1	10054	
10001	1	801	2000	0	1	10001	
10011	801	1	2000	0	1	10011	
10051	801	801	2	0	1	10051	
20004	702	4	1000	0	1	20004	
20014	4	702	1000	0	1	20014	
20054	702	702	2	0	1	20054	
20001	1	802	2000	0	1	20001	
20011	802	1	2000	0	1	20011	
20051	802	802	2	0	1	20051	

其中

新增乘客 1、2 的接送弧及虚拟节点等待弧，与乘客网中的新增弧相同

新增车 1 出库 link 10004、10014 入库 link 10001、100011、车辆 2 出库 20004、20014 和 入库 20001、200011，此外还有虚拟节点等待弧 10054、10051、20054、20051

type 100 表示乘客接、200 表示乘客送、2 表示等待

type 1000 表示车出库、2000 表示车入库

⑥ 车辆节点

Table 8.Input train node

node_id	description
1	station1
2	station2
3	station3
4	station4
5	station5
6	station6

Table 8.Revised Input train node

node_id	description
1	station1
2	station2
3	station3
4	station4
5	station5
6	station6
501	
601	
502	
602	
701	
801	
702	
802	

其中新增乘客 1、2 的接送节点 501、601、502、602

新增车辆 1 出库节点 701、入库节点 801

车辆 2 出库节点 702、入库节点 802














5. 输出说明

Table 7 output summary

文件名	数据内容																					
out_put_passenger_results	<table><tr><th>passenger</th><th>node_sequence</th><th>time_sequence</th><th>volume</th><th>departure</th><th>arrival_t</th><th>total_travel_time</th></tr><tr><td>1</td><td>501;2;5;6;3;601;601</td><td>6;7;8;9;10;11;12</td><td>1</td><td>6</td><td>12</td><td>6</td></tr><tr><td>2</td><td>502;2;5;6;3;602;602</td><td>9;10;11;12;13;14;15</td><td>1</td><td>9</td><td>15</td><td>6</td></tr></table>	passenger	node_sequence	time_sequence	volume	departure	arrival_t	total_travel_time	1	501;2;5;6;3;601;601	6;7;8;9;10;11;12	1	6	12	6	2	502;2;5;6;3;602;602	9;10;11;12;13;14;15	1	9	15	6
passenger	node_sequence	time_sequence	volume	departure	arrival_t	total_travel_time																
1	501;2;5;6;3;601;601	6;7;8;9;10;11;12	1	6	12	6																
2	502;2;5;6;3;602;602	9;10;11;12;13;14;15	1	9	15	6																
out_put_train_results	<table><tr><th>train_id</th><th>node_sequence</th><th>time_sequence</th><th>volume</th><th>departure</th><th>arrival_t</th><th>total_travel_time</th></tr><tr><td>1</td><td>701;4;2;502;2;501;501;501;501;2;502;2;502;2;501;2;1;801</td><td>1;2;4;5;6;7;8;9;10;11;12;13;14;15;16;17;19;20</td><td>1</td><td>1</td><td>20</td><td>19</td></tr><tr><td>2</td><td>702;4;2;502;2;501;501;501;501;2;502;2;502;2;501;2;1;802</td><td>1;2;4;5;6;7;8;9;10;11;12;13;14;15;16;17;19;20</td><td>1</td><td>1</td><td>20</td><td>19</td></tr></table>	train_id	node_sequence	time_sequence	volume	departure	arrival_t	total_travel_time	1	701;4;2;502;2;501;501;501;501;2;502;2;502;2;501;2;1;801	1;2;4;5;6;7;8;9;10;11;12;13;14;15;16;17;19;20	1	1	20	19	2	702;4;2;502;2;501;501;501;501;2;502;2;502;2;501;2;1;802	1;2;4;5;6;7;8;9;10;11;12;13;14;15;16;17;19;20	1	1	20	19
train_id	node_sequence	time_sequence	volume	departure	arrival_t	total_travel_time																
1	701;4;2;502;2;501;501;501;501;2;502;2;502;2;501;2;1;801	1;2;4;5;6;7;8;9;10;11;12;13;14;15;16;17;19;20	1	1	20	19																
2	702;4;2;502;2;501;501;501;501;2;502;2;502;2;501;2;1;802	1;2;4;5;6;7;8;9;10;11;12;13;14;15;16;17;19;20	1	1	20	19																
out_put_upper_bound	暂无																					
Out_put_gap	暂无																					
Out_put_volume	详见输出																					

6. 程序操作说明

第一步，将初始数据文件夹中的六个输入文件复制到当前文件夹，并替换当前文件夹与之同名的六个文件。

 初始数据	2019/5/9 18:51	文件夹	
 input_passenger.xlsx	2019/4/23 20:45	Microsoft Excel 工...	9 KB
 input_passenger_link.xlsx	2019/5/9 18:40	Microsoft Excel 工...	6 KB
 input_passenger_node.xlsx	2019/5/9 18:40	Microsoft Excel 工...	6 KB
 input_train.xlsx	2019/4/23 20:45	Microsoft Excel 工...	9 KB
 input_train_link.xlsx	2019/5/9 18:40	Microsoft Excel 工...	10 KB
 input_train_node.xlsx	2019/5/9 18:40	Microsoft Excel 工...	6 KB
 output_gap.xlsx	2019/5/9 18:40	Microsoft Excel 工...	5 KB
 output_passenger_results.xlsx	2019/5/9 18:40	Microsoft Excel 工...	5 KB
 output_train_results.xlsx	2019/5/9 18:40	Microsoft Excel 工...	5 KB
 output_upper_bound.xlsx	2019/5/9 18:40	Microsoft Excel 工...	5 KB
 output_volume.csv	2019/5/9 18:40	Microsoft Excel 逗...	3 KB
 vrp_generate_network.py	2019/5/9 18:39	PY 文件	107 KB

第二步，打开 vrp_generate_network.py,并运行程序。

第三步，查看以 output 命名的输出文件，此时由于版本兼容问题，若出现无法打开文件的形式，请将文件后缀修改为**.xls**。

参考文献：

[1] Integrated optimization of urban rapid transit timetabling and circulation planning: A multi-commodity flow formulation and decomposition schemes based on coupled network representations[商攀师兄姚宇师姐尚未发表的论文]

.