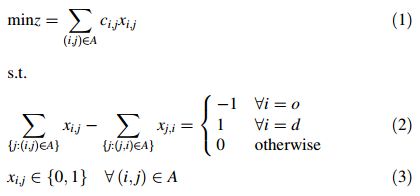
# 一、经典VRP问题的定义+分类

一类是为乘客（货物） 设计单程的路线，一条路径上全为接人（货） back haul 或全为送人（货） line haul。一类是服务乘客或者货物，将其从不同的original node运送到其destination node，在其起讫点对应有服务需求的time window，不失一般性都为第二类。There are two classes of the vehicle routing problem (VRP): (1) designing line haul services for customers from the depot and back haul services for customers to the depot, and (2) transporting passengers or goods between specific origins and destinations with possible requested time windows.

# 二、最短路问题分类

## 1、带有流平衡的最短路问题（[17]）



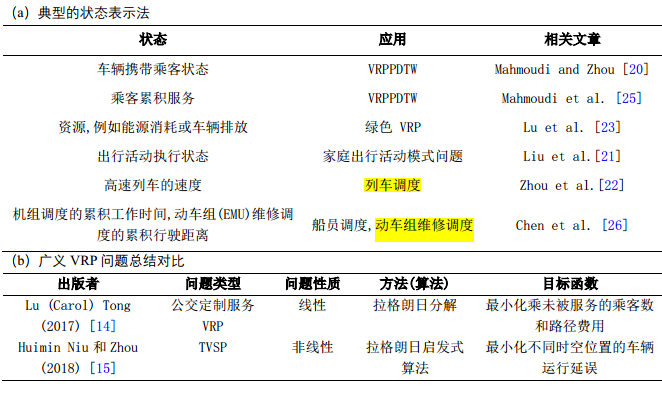
解决该最短路的方法主要有两个（[17]），一个是label setting算法（不可以有负回路，相当于Dijkstra算法），一个是label correcting算法（可以有负回路PDM算法、可以有负权但是不含负回路的逐次逼近算法）

## 2、时间-space的最短路问题（[18]、[19]）

## 3、三维网络

（time-space-state）（[20]、[21]、[[22] joint optimization of train timetables and speed profiles]、）或高维网络（[14]、[24]）的最短路问题





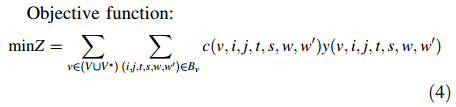
# 三、Space-Time-State-Based model

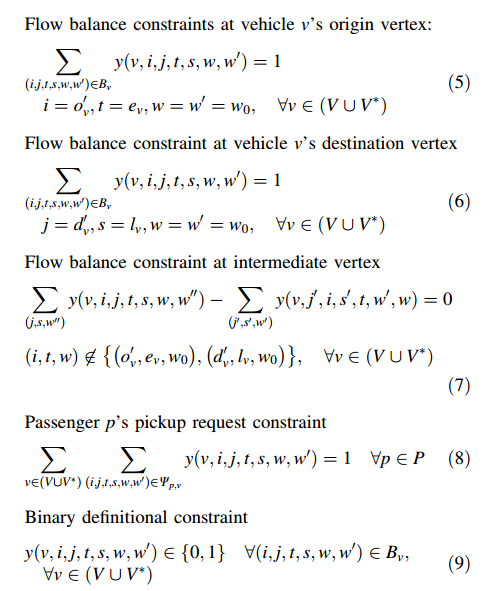
物理网络构成为（N,M），其中i，j∈N，（i，j）∈M。

时空网络构成为（E,A），其中（i,t），(j,s)∈E，（i,j,t,s）∈A

STT网络构成为（V,A）,其中（i,t,w），(j,s,w’)∈V，（i,j,t,s,w,w’）∈A

## 1、模型





## 2、 小案例-Gams求解

所构建的虚拟网络如图6所示，圆圈内数字表示节点编号，连接线上的数字表示旅行时间。



车辆k=1,2；乘客p=1,2；时间戳t=1,2,…,30；点（虚拟路网）i=1,2,…,12；i\_source(i)=7； i\_sink(i)=8；输入信息：

表2 输入数据集

|  |  |  |
| --- | --- | --- |
| 编号 | 数据内容 | 参数名称 |
| 1 | 物理节点信息 | i/i\_source(i)/ i\_sink(i) |
| 2 | 路网连接 | arcs(i,j,t,s)，waiting\_arcs(i,j,t,s)，travel\_cost(k,i,j,t,s) |
| 3 | 时间窗信息 | win(t,n) |
| 4 | 乘客信息 | p/ p\_origin(i,t)/ p\_destination(i,t) |
| 5 | 车辆信息 | k |
| 6 | 状态信息 | w，w\_trans(w,wp) |

下面对输入数据量进行详述。

（1）i/i\_source(i)/ i\_sink(i)

i为节点数， i\_source(i)为所出车场，i\_sink(i)为所入车场。

（2）win(t,n),是为确定p\_origin(i,t)/ p\_destination(i,t)服务

t表示时间窗所包含的各时间点，n表示时间窗序号，win(t,n)的值表示该时间窗对应的节点。时间窗t取值如图中方括号所示。



（3）arcs(i,j,t,s) /waiting\_arcs(i,j,t,s)

表示时空弧（运输弧和等待弧）是否可行。其中包括起止节点编号，已经走行该弧段所需时间。对于可行的时空弧其值取为1。

travel\_cost(k,i,j,t,s)

可行的时空弧的费用，运输弧和时间成正比（此处比例为1），等待弧费用为0.5.

（4） p/ p\_origin(i,t)/ p\_destination(i,t)

p为乘客数量，p\_origin(i,t)和 p\_destination(i,t)分别表示在时间t、虚拟点i处的可行行为，当p\_origin(i,t)=1代表可以在该点该时刻接乘客1，当p\_origin(i,t)=2代表可以在该点该时刻接乘客2；当p\_destination (i,t)=-1代表可以在该点该时刻送乘客1，当p\_ destination (i,t)=-2代表可以在该点该时刻送乘客2；

（5）k

其值为车辆数目。

（6）w/ w\_trans(w,wp)

可行状态和状态转移方程。

set w /0,1,2,1\_2/;

parameter w\_trans(w,wp) TRANSITION MATRIX/

0. 1 1

0. 2 2

1. 0 -1

1. 1\_2 2

2. 0 -2

2. 1\_2 1

1\_2. 1 -2

1\_2. 2 -1

/;

w\_trans(w,w) = 0;

**关键：**根据(3)、（4）、（6）确定出可行的时空状态转移弧和可行的每个乘客的pick-up弧和drop-off弧。

parameter arcs\_w(i,j,t,s,w,wp) link w transition ;

\* add space-time-state arcs

arcs\_w(i,j,t,s,w,w)$(trans\_arcs(i,j,t,s)) =1;

arcs\_w(i,j,t,s,w,w)$(waiting\_arcs(i,j,t,s)) =1;

arcs\_w(i,j,t,s,w,wp)$(p\_origin(i,t)=1 and (w\_trans(w,wp) eq p\_origin(i,t))and trans\_arcs(i,j,t,s))= 1;

arcs\_w(i,j,t,s,w,wp)$(p\_origin(i,t)=2 and (w\_trans(w,wp) eq p\_origin(i,t))and trans\_arcs(i,j,t,s))= 1;

arcs\_w(i,j,t,s,w,wp)$(p\_destination(i,t)= -1 and (w\_trans(w,wp) eq p\_destination(i,t))and trans\_arcs(i,j,t,s))= 1;

arcs\_w(i,j,t,s,w,wp)$(p\_destination(i,t)= -2 and (w\_trans(w,wp) eq p\_destination(i,t))and trans\_arcs(i,j,t,s))= 1;

parameter arcs\_pickup(p,i,j,t,s,w,wp);

arcs\_pickup(p,i,j,t,s,w,wp)$((p\_origin(i,t)=value\_p(p)) and (w\_trans(w,wp) eq p\_origin(i,t)) and arcs(i,j,t,s))= 1;

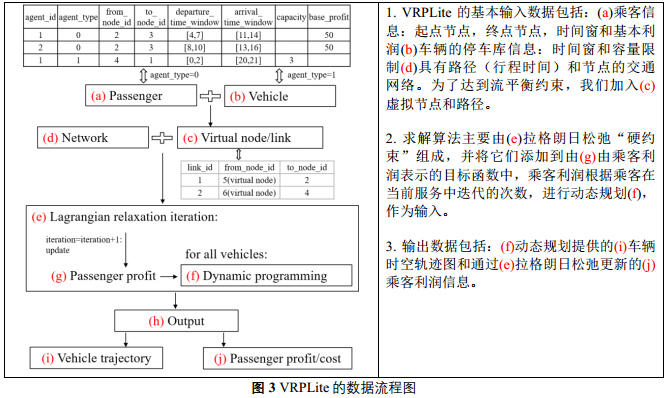
parameter arcs\_delivery(p,i,j,t,s,w,wp);

arcs\_delivery(p,i,j,t,s,w,wp)$((p\_destination(i,t)=-value\_p(p)) and (w\_trans(w,wp) eq p\_destination(i,t)) and arcs(i,j,t,s))= 1;

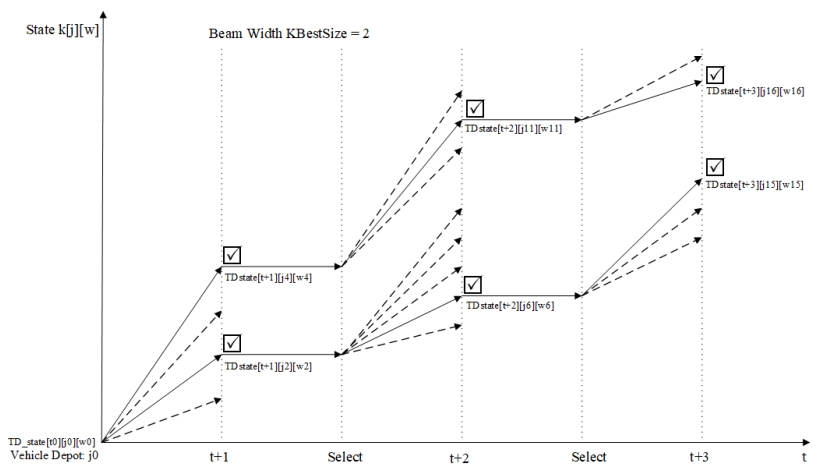
从上面三个确定可以看出，arcs\_w(i,j,t,s,w,w)$(trans\_arcs(i,j,t,s))和arcs\_w(i,j,t,s,w,w)$(waiting\_arcs(i,j,t,s))对于任意可行的(i,j,t,s)时空弧对应的(i,j,t,s,w,w)时空-状态转移弧都可行。

# 四、求解包框架

## 1、数据流程图



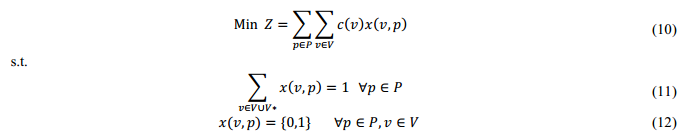
## 2、动态规划实现beam search（集束搜索）



## 3、列生成（更多细节看[28] Lübbecke, M. E., & Desrosiers, J. (2005). Selected topics in column generation. Operations Research, 53(6),1007-1023.）

决策变量从对于弧的选择转换为对于路径的选择，引入基于基于路径的变量x(v, p)，如果乘客p由车辆v服务则等于 1，否则等于 0。

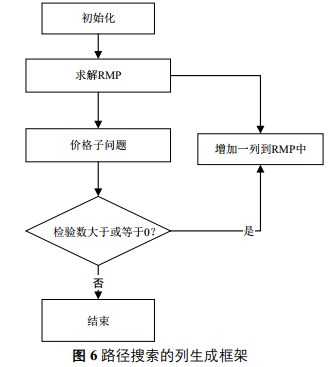
RMP问题：



RMP是线性规划问题，求解可以得到（11）中每个约束条件的乘子，即对偶价格。

然后带到子问题中，求解能使主问题降低的新路径，所以目标函数可以求出当前价格下的最小费用的路径（运用基于时间和状态的路径搜索引擎-隐枚举法来求解子问题）检验检验数后加入RMP中成为新列。

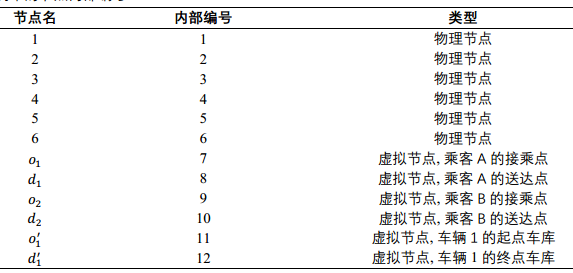
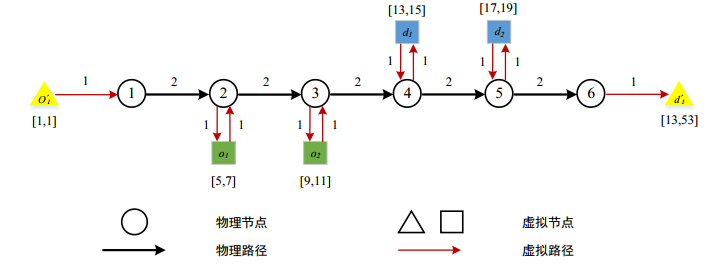
C:\Users\zhangqin\AppData\Local\Temp\1555601132(1).png

（我觉得这里是和否写反了）

RMP被松弛成线性规划问题求解时，可以对得到的非二进制整数分支成=0和=1，进行求解。（但是乘子的确定是基于非二进制整数，那带入子问题的应该是？？）

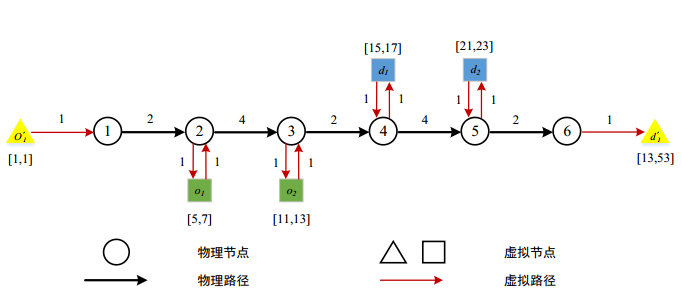
# 五、求解算例

## 1、干道路网

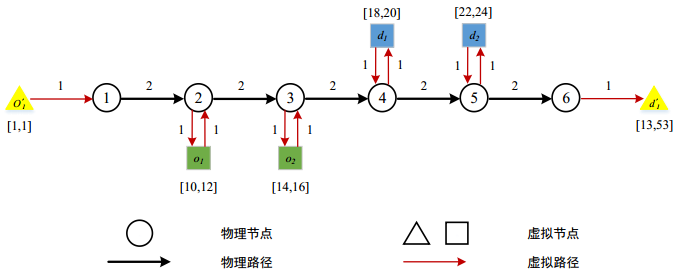


物理节点和虚拟节点的数量由给定的标号法来确定。首先，所有物理节点从 1 标记到 N，其中 N 是物理节点的总数。其次，每个乘客的接送节点按照从 N+1 到 N+2P 顺序编号，其中 P 表示乘客的总数。最后，我们按顺序从 N+2P+1 到 N+2P+2M 标记每辆车的起点车库和终点车库的数量。

## 2、1的基础上延长路径行程时间



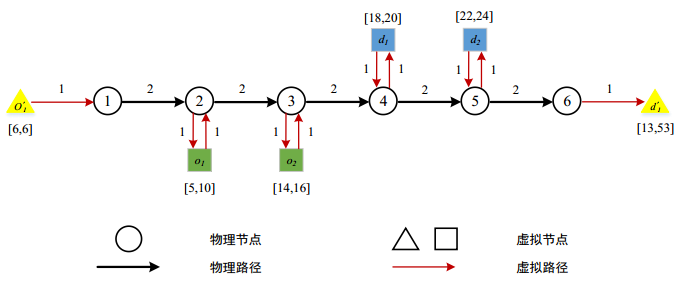
## 3、1的基础上增加车辆等待时间，必须在o1处等待-车等乘客



可以看出，车辆在时间 5 到达节点 o1 并等待直到时间 10 开始发车，所以车辆等待时间为 5。车辆的总行程时间为 25，但车辆等待时间的成本比率仅为 0.5，则总成本的最优值等于 25-

0.5\*（10-5） =22.5

## 4、1的基础上增加乘客等待时间（车辆出发时间延误到6）-乘客等车



显然，乘客 A 从时刻 5 开始等待直到车辆到达乘客出发点的时刻 10 才上车，所以乘客 A 的等待时间是 5。 另外，因为乘客等待时间的成本比率是 0.5， 并且车辆的总行程时间是 20，所以总成本是 20+0.3 \* 5= 21.5。

## 5、列生成过程的分支过程

使用Dynamic Programming.py的def add\_new\_node\_and\_link\_for\_agent()生成的虚拟网络如图：(按照乘客1的o-d，乘客2的o-d，。。。。车辆1的o-d，车辆2的o-d的顺序)



1个车辆，乘客起始费用10,车辆能力为3，车辆基本费用22。路途时间需更改（看程序）。

link\_type：

（1）transportation & drop-off为0；

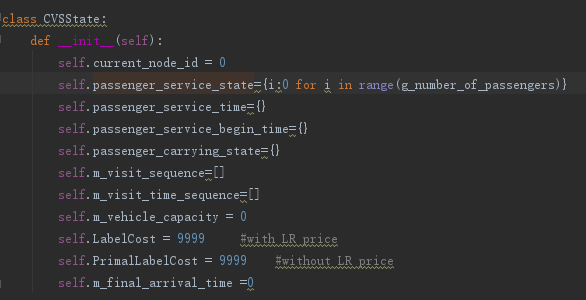
（2）pick-up 的分别为1和2，（2-7）和（7-2）=1，（5-9）和（9-5）=2；

（3）车辆进出库（11-4）=-101，（1-12）=101

node\_type:

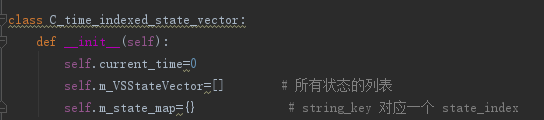
（1）物理点为0；（2）7,9为1；8,10为2；（3）11,12为3.

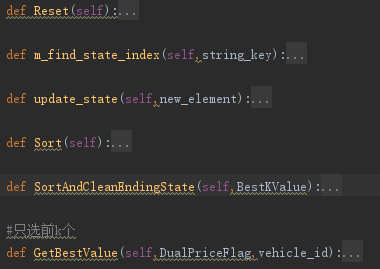
CVSState用来更新、记录车辆的位置、状态、费用等信息。



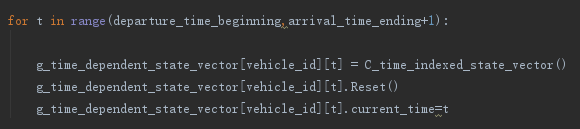


C\_time\_indexed\_state\_vector用来对每个时间t上不同状态费用进行排序。

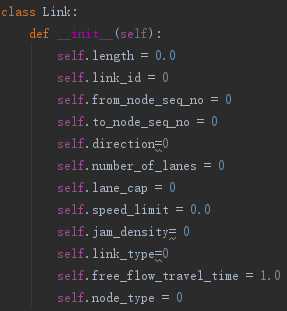
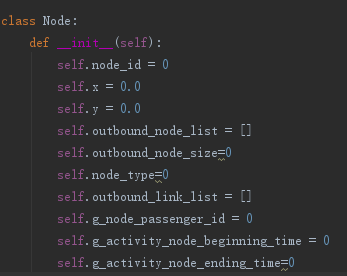


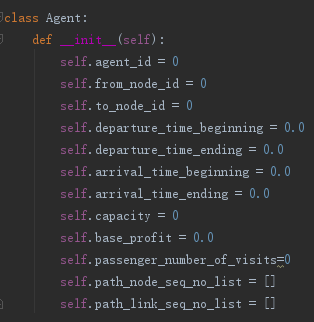


C:\Users\zhangqin\AppData\Local\Temp\1555824999.png

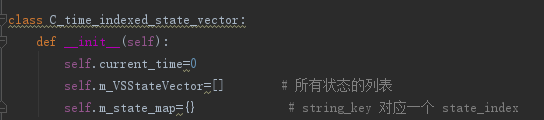


C:\Users\zhangqin\AppData\Local\Temp\1555828333(1).png





C:\Users\zhangqin\AppData\Local\Temp\1555824999.png的最小元素是C\_time\_indexed\_state\_vector这个形式 (对应于当前的v和t下的所有的当前状态)，每个元素C\_time\_indexed\_state\_vector记录：



即，当前时间，所有状态（状态里包括很多实例，每个实例都是CVSState这个状态），以及map（map也是当前v和当前t下已经有的状态，用string\_key的方式表示出来，和m\_VSStateVector对应）