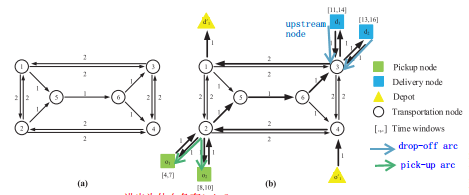
2016- Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations

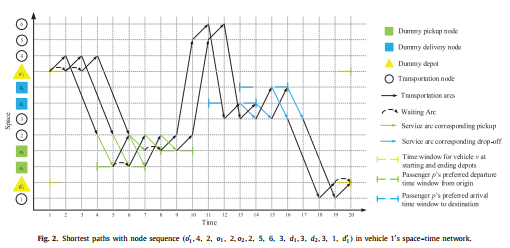
本文旨在解决带有时间窗的接取送达的VRP问题（VRPPDTW），从二维space-time network扩充到三维state-space-time network，求解算法基于拉格朗日松弛框架，对于松弛子问题采用动态规划（DP）算法。

# 1、一维网络：transportation network



将接客点、落客点，汽车起讫点，普通交通物理网络点三类区分，前两类引入虚拟点。

# 2、二维网络：space-time network



将时间以时间戳形式表示，由等待弧、一般弧和接取送达服务弧三类弧构成，司机的等待行为规定：只能在时间窗里接送乘客，否则需等待，以时间窗限定可行路径时间范围（几种可行路径方便表示）。为了保证能够生成可行解（因为有的客人可能不能够被服务，此时可能造成不可行解），引入虚拟车辆的概念，对不能用实体车辆服务的乘客用虚拟车辆来完成任务。

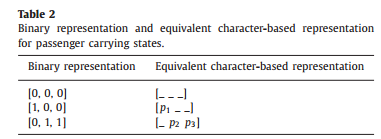
# 3、状态state的表示和数量的计算

补充：动态规划重要概念包括阶段stage、state、decision、state transition。

本文中：

1) 阶段以时间戳t为划分，以每一时间精度为一个阶段。

2) 状态由两个指标表示：点i和乘客携带状态w。乘客携带状态w可以由|P|维向量C:\Users\zhangqin\AppData\Local\Temp\1547111155(1).png1表示该车辆携带该乘客，否则0；同时引入乘客携带状态的解读：



这样的状态表示方便对车辆能力约束的直接控制，当车辆能力为2时，则该车辆的可行状态为最多两个乘客。因而对每个车辆的可能的状态个数就可以明确表示并计算出来为：C:\Users\zhangqin\AppData\Local\Temp\1547111438(1).png，该公式求得的事改车辆状态个数的上限，运用时间窗不重叠的两个乘客不可能被同一个车辆同时接，可以缩减状态个数。

3) 状态转移：从一个状态到另一个状态的转移有三个规则：

规则1.在去接人的过程中，那条弧从0变成1；

规则2.在送客的过程中，送完客回到路网的那条弧，从1变成0；

规则3.其他弧的状态不可能改变。

# 4、三维网络的构建：state-space-time network

三维网络点由单个指标构成：node index i, time interval index t, and passenger carrying state index w，弧表示为(i, j, t, s, w, w’)表示从点(i, t, w)到点（j, s, w’）.

5、基于三维网络（三种弧）构建多商品网络流规划模型（其实相当于4维，车辆构成第四维）

1) 建立模型前准备：

3种弧分别为service arcs, transportation arcs and waiting arcs.

其中，前两类的弧的费用（routing cost）计算方法一致，与旅行时间（travel time）成一定比例。但是虚拟车辆的比例要大于实体车辆（本文是两倍）。等待弧的费用是等待时间的比例，但是这个概率比实体的前两类的弧的费用比例要低（省油），且在车辆起讫点出为0，对于虚拟车辆，等待弧费用都为0；

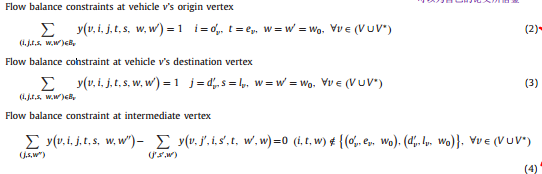
2) 决策变量：y(v, i, j, t, s, w, w’)

在约束条件中，送客约束和流平衡约束中最后为0回到终点相冗余，因此不考虑送客之后变成0这一约束条件，减少了冗余约束。

3) 目标函数：C:\Users\zhangqin\AppData\Local\Temp\1547115988(1).png减少总费用，虚拟列车情况下，以费用为目标函数，使虚拟列车的费用较高，从而减少使用虚拟车；同时又能简化模型，让困难约束易于表达，不用分门别类，保证得到可行解

4) 约束条件：

流平衡基本约束条件：



各个约束式子间相互独立，因而不属于hard约束



接乘客只能由一辆车辆完成（虚拟或实体），这个是hard约束，也是拉格朗日松弛约束的约束条件，因为车辆和车辆之间有联系，使得不能并行计算，增加求解时间。

C:\Users\zhangqin\AppData\Local\Temp\1547116146(1).png

时空网络保证了时间变量计算的正确性（arc直接保证了是否有对应时间距离），状态-时-空网络方便了对时间窗约束的处理（可以说是预处理了一些约束），能力约束可以以可行状态表示。

此时模型中，因为y也是变量，乘子也随时在变，因而不容易求解y，因此采用拉格朗日求解框架。

# 6、拉格朗日松弛

1）将相互关联的接乘客约束松弛至目标函数，编程拉格朗日对偶问题，并将其整理后变成time-dependent least-cost path problem：

C:\Users\zhangqin\AppData\Local\Temp\1547116321(1).png

广义费用ξ只有在service arc时才需额外加上对应乘客的乘子，剩余的arc均不变。

当每次给定一个乘子时，即变成单一商品网络流问题，然后可用动态规划问题解决子问题。

2）动态规划算法，相当于标号法（寻最短路）

对每一个车辆（实体+虚拟），先初始化累计费用，即L（i，t，w）=正无穷；对每一个点（点、时间、状态三个维度）的前一点为-1。之后对每一个时间t下，对link（i，j），计算每一个状态的下一个可能状态，状态转移时间、状态转移费用，并将费用记为累计费用，若改累计费用比上一个累计小，则更新对应点的四个指标，循环下去。

拉格朗日对偶问题的子问题（即每个V）用DP可求得最小费用，得到的解若是原问题的可行解，则该解为最优解，否则的话利用启发式算法求得可行上界解。

3）个人理解拉格朗日的重点：

a)如何求得对偶问题各个子问题的解（下界）---一般借鉴最短路、最小费用的相关算法；本文采用DP；

b）怎样将子问题的每个最优解（转为原问题时不可行时）变成原问题的可行解（即原问题的上界），本文采用单个车辆逐次满足的方式；

c)每步迭代如何更新拉格朗日乘子，（步长一般是随着迭代的增加递减，因为越往后越接近可行解，需要让其稳定，保证收敛），拉格朗日乘子（相当于惩罚数）当乘客P被多次接时，将其对应费用变大；当其为1时，保持费用不变，当其没被接时，使其费用变小，从而经过不断迭代使原问题尽可能可行。

# 7、缩小搜索区间：

3个规则可用于明显缩减搜索区间，（两个乘客的时间窗和车辆在两点间的最短旅行时间的大小关系）1个可用来估算缩减概率

# 8、本文可采纳的地方有以下内容：

1）本文问题实际相当于四维问题，车辆-state-space-time 网络的构建，对时间窗的预处理（time的限制），能力约束的预处理（state的限制），因而使得模型求解方便。

2）引入虚拟列车，保证解的可行性。

3）拉格朗日松弛求解框架，将四维问题转换为三维问题，然后可用最短路相应算法求解子问题。

# 9、本文的不明白之处：

1）service arc 为什么会有时间费用，是因为距离实际路网node有一定距离以及上下车时间？

答：将service arc剥离，建立虚拟接客送达的点，对于接取、送达乘客这一行为可以有以时间单位（时间戳最小精度），构造该类弧可以方便体现出状态转移。

赋予该弧的权重是人为规定的，至少为单位时间，保证都是以时间戳为最小单位的移动。同时在算法构造上防止因为时间未移动，但状态却改变造成的“打转”。

因此以后在使用该种（state）-space-time网络情况下，注意将每一路径（path）的始末单独拉出来可能会好一点。

2）为什么pick-up arc是downstream node 而drop-off是upstream node？？按照理解，接人是去的那里，从无变有，应该是downstream，但是送人也应该是去的那里，从有变无，也应该是downstream呀？

答：人为规定

3）为什么虚拟车辆的终点需要是其起点？

答：这样保证不出行，cost就是0，出行cost就变大，是人为故意使其使用成本增加。

算法初衷：尽量保证原问题和对偶问题的可行性，这种可行性有时候很难保证，所以在原问题时需要一步一步进行调整，使其由对偶问题得到的解变得可行，姚师姐文章在这方面有所成就，牛慧敏老师中指出了其复杂性（对称性等等）。

对偶问题的最优解时变量取值不一定唯一，姚师姐的convex问题，相对来说原始和对偶问题的解unique一点

给自己的思路：prime去尽量实现feasibility，而对偶问题中有组合拍卖方法可供借鉴道岔组合拍卖，时空网络对组合拍卖有意义，普通拍卖没有体现时空观。

接下来任务顺序：

姚师姐论文

牛惠民老师论文

周老师给的那本书中的assignment部分

组合优化