# 1、问题说明

本算法适用于多车辆带有乘客接、送时间窗的综合优化乘客需求、车辆供给、道路能力的问题。

## 1.1 传统VRP问题

传统的VRP问题可以概括为：一个车队为一定数量的乘客提供服务，我们需要在车辆的容量限制约束下，找到一系列车辆径路（从停车场出发，完成对乘客的服务后再回到停车场），达到诸如路程最短、成本最小、耗费时间最少等目的。

输入：虚拟点对点的服务网络，乘客需求（起讫点、服务时间窗）、车辆服务能力

输出：分配给乘客的对应车辆

目标：系统最优

## 1.2 交通流分配问题

交通流分配问题笼统来说是将多车辆供给分配到基础线路网络中。

输入：实际路网、道路能力（拥堵）、车辆的起讫点（每个车辆已经被分配客流）

输出：在link或者path上的连续车流量。

目标：系统最优或用户平衡

## 1.3 本文考虑的场景

综合考虑乘客需求、车辆供给和路网线路能力，将VRP问题和交通流分配问题相结合，具体如下：

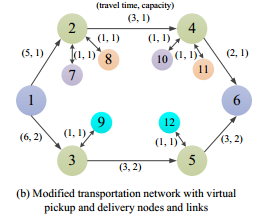
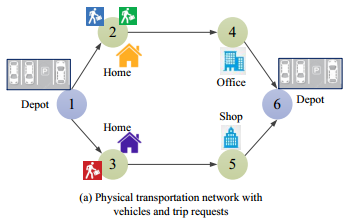
（1）考虑交通拥堵情况的实际交通路网；

（2）考虑带有接取、送达、服务时间天窗的乘客需求；

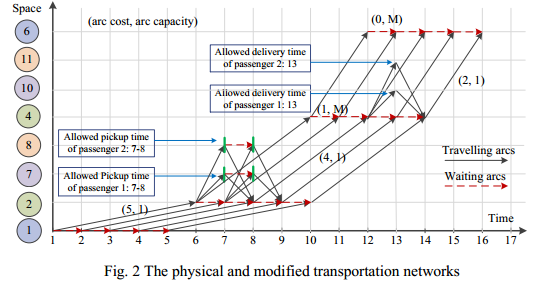
（3）考虑车辆能力的共享riding的自动驾驶车辆；

（4）最终实现系统最优。

## 1.4 实际物理网络→修订运输网络→时空网络→时间-空间-状态网络



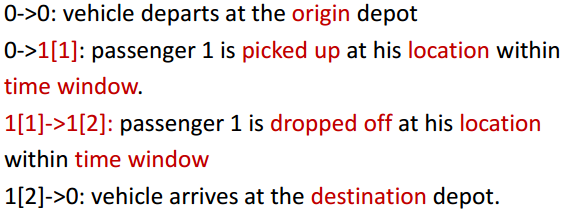
当车辆服务能力为1时，修订运输网络（b）中1→2→4→6时空网络图如下：



时空网络可以表示出有路网线路能力限制和无能力限制两种情况，但是无法表示出：（1）当乘客被一个车接取后，必须由同一个车送达；（2）车辆的能力无法表示。因此，引入累积乘客服务状态变量，并且扩展为时间-空间-状态高维网络。

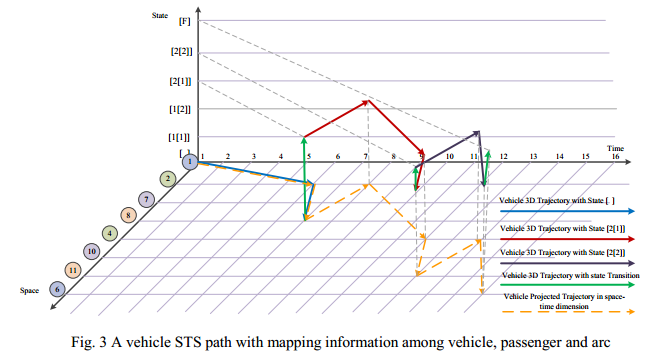
**状态转移逻辑：**

（1）以能力为1，1个乘客为例：

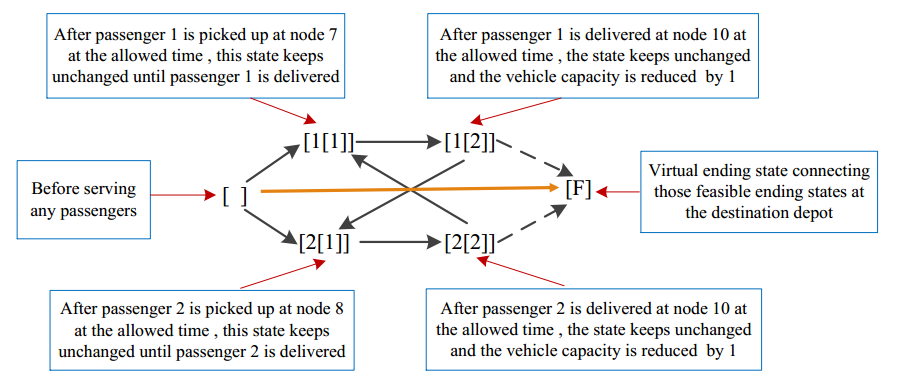


1[1]→0是不可行状态转移，因为必须要drop-off

对应的时-空-状态网络：（从t=1时出发，t=2时直接向后平移）

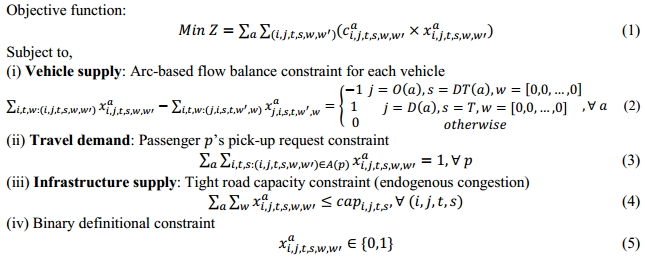


（2）以能力为1，2个乘客为例：



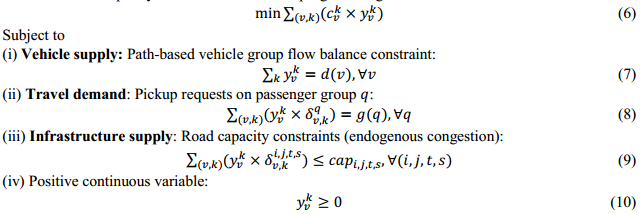
# 2、arc-based agent-based model（model 1）

每一个车辆和乘客均视为单独的个体:



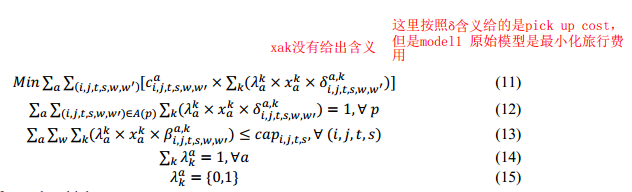
# 3、path-based flow-based mode（model 2）

引入车辆group和乘客group的概念，组内元素有相同的特性，对于一个车辆group内，每列车有相同的起讫点，起讫时间窗；对于一个乘客group内，每个乘客有相同的旅行需求（接取地点和接取时间/（和/或）送达地点和送达时间）。该模型思想为将连续流量分配到成对的OD pairs中，而不是单个的车辆，大大减少了变量数。所有可能的时-空-状态信息分配能够被提前枚举出来，从而使得后续工作只剩下将车辆分配到网络中从而满足顾客需求的同时不违反道路能力限制。

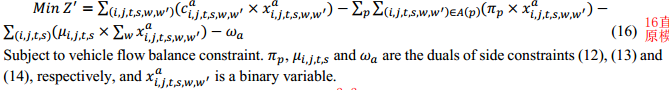


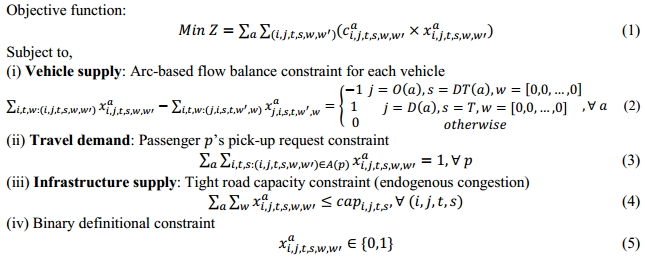
# 4、model 1 reformulation（DW分解）但是式11-15没有看懂

## 4.1 主问题



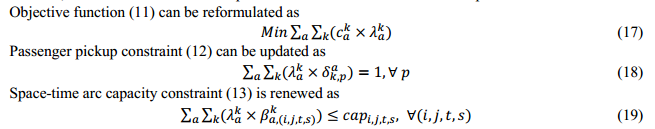
## 4.2 子问题为：





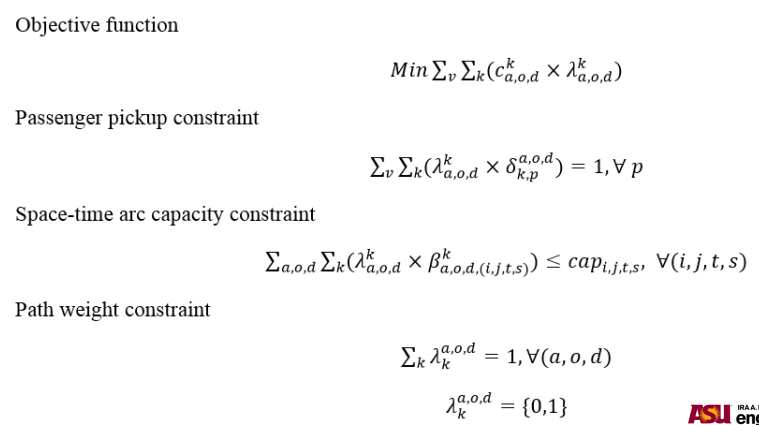
π、µ、*w*分别表示dual price of passenger pickup constraints，of capacity constraints and path weight constraints

## 4.3 最终通过path-based formulation构造主问题为：（所以11-15式是个过度？）



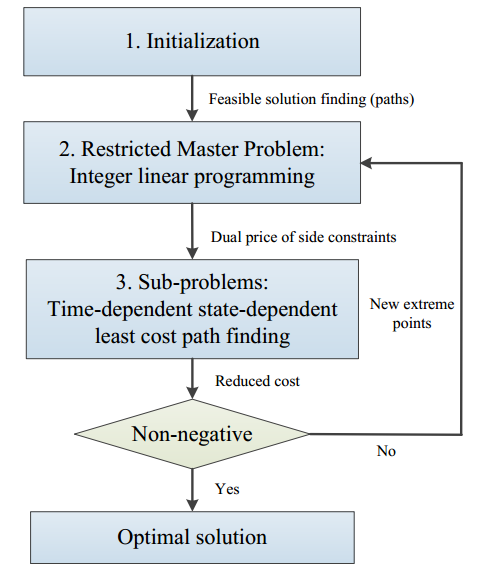
（式17-19中不会有完全一模一样的path，此处path相当于列，式18表示经过p乘客的locaiton的path只能存在一个，相同的path，其接取送达同一个乘客，同时少个path weight constraint。）

PPT中的书写：



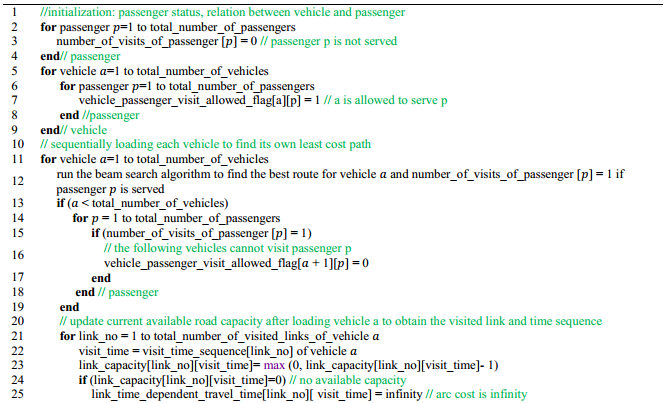
Path weight是对于同一个v，其起点相同，终点相同，则应保证只有一个路径，即lamda累和后，起讫弧xijtsww’应为1。

## 4.4 列生成算法



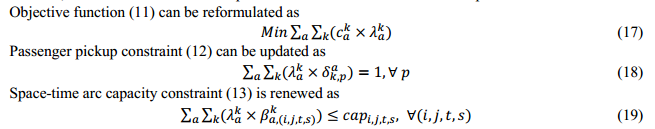
该reformulation of model1（Re-model1），主问题为（11-15），子问题为（16 subject to flow constraint），使用列生成算法流程：

第一步：初始化，得到一组可行解（即可行的paths）



C:\Users\zhangqin\AppData\Local\Temp\1554798354(1).png

第二步：求解RMP问题，得到一组解（），（solved by standard solver or a hybrid of solvers and branch-and bound）



同时可以求出两类side constraints的dual prices，即单纯形乘子

第三步：求解子问题（即生成一个可行的column）（怎么求解？beam search算法）

目标函数：（自己对于列生成目标函数的确定有点混，为什么是‘-’）

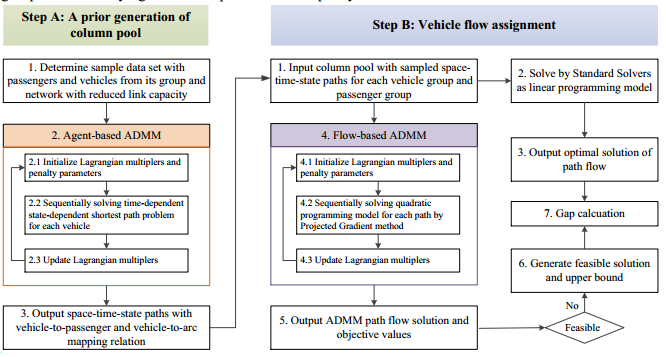


Subject to flow balance constraint

第四步：判断检验数，如果检验数≥0（我觉得这里是正时为最优解，因为表示新增加的列只会使得主问题的解增加？与tony的论文相反），则其为最优解，否则将生成的新列加入到RMP问题中，循环第二步。注意：当新列产生时，该path（即新列）的费用、incidenceδ和β即可以生成。

# 5、model 2 reformulation（Column-pool-based approximation）

将原问题分解为两个阶段：（图应该画错了，2.3应该循环至2.2）



## 5.1 第一阶段：生成column pool

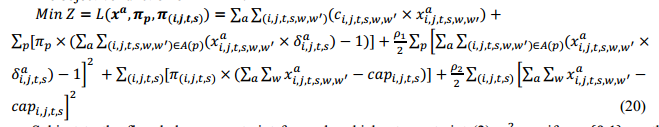
（即从每一个车辆group选择出sample（个数的确定还没给出理论解释，有待后续研究），通过ADMM求解arc-based agent-based formulation，生成许多的时-空-状态paths），这样对于未来顾客需求、车辆供给和交通网络供给有改变时，可以以这些为基础上进行改变。

具体操作：

（1）从每个车辆group和乘客group中选择一些作为sample，给定reduced路网（路网容量相对原网络来说需要相应减少）

（2）求解Agent-based ADMM（根据弧，生成path(即column)）

目标函数为：



（2.1）初始化拉格朗日乘子和惩罚参数

（2.2）按**顺序**对每个列车求解出最短path（每一个车计算时都基于已经求出的列车的最短路更新后的费用），求解方法有很多，比如动态规划使用Beam-search算法来求解time-dependent state-dependent shortest paths问题

C:\Users\zhangqin\AppData\Local\Temp\1554809744.png

（2.3）更新拉格朗日乘子，重复至2.2直到满足终止条件



（3）输出paths（column pool），其中包含信息：什么车能服务什么乘客，什么车占用了什么弧。

## 5.2 第二阶段：给定column求解，选择出哪些column保留

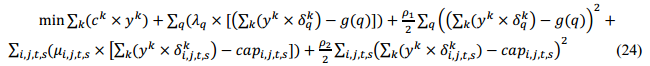
通过ADMM求解STS path-based flow-based 线性规划问题，将来自不同车辆group的车辆（即同一个）分配，从而去服务来自不同乘客group的乘客，同时满足道路能力约束，即往columns里去分配vehicles的数量从而来满足乘客需求和网络能力约束。

具体操作：

（1）输入第一步确定的每个vehicle组的columns和每个passenger组的columns。

（2）求解Flow-based ADMM（确定每个column分配的vehicle的量）

目标函数：



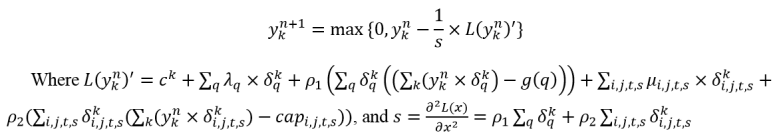
其中q是passenger group，其利用hessian矩阵求极值很困难。所以使用ADMM将原问题分解成子问题，在每步迭代内按顺序求解每个column。

（2.1）初始化乘子lamda和惩罚项roh，初始化每个column上的流量为0

（2.2）按顺序求解每个column的子问题

C:\Users\zhangqin\AppData\Local\Temp\1554810975(1).png

求解方法可以使用projected gradient method：



（2.3）更新拉格朗日乘子λ和µ，重复至2.2，直到满足终止条件

C:\Users\zhangqin\AppData\Local\Temp\1554811298(1).png

乘子也可以使用分支定界的方法来更新。

（3）输出第二步最终结果，看起是否可行解，如果不可行，将其调整为可行解（按照顺序将ADMM求得的column的流量放在路网上，当超出乘客需求或者能力需求时，将这些flow移除，当有些顾客没有办法被服务时，可以使用虚拟车（直接给费用）来寻找到可行解）。

（4）使用软件求解原始问题，生成最优解，可以同（3）结果对比，看出ADMM的解的质量。

补充：关于乘子即为对偶函数的解的理解

|  |  |  |  |
| --- | --- | --- | --- |
| 最终的基变量 | | 非基变量 | |
|  | (当松弛变量不能完全构成单位矩阵，此处XS指最开始确定的单位基) |
| I | （即为最终的解，因为非基变量取值为0） |  |  |
| O（单纯形法中检验数那一行） |  |  |  |

基可行解对应于可行域的顶点，而线性规划的最终最优解必然在某个顶点。所以最终的最优基对应的最优解时，非基变量的取值为0。迭代最终的单纯形表中B代表最终的基。

因为在取得最优解时，原问题的最优解时目标函数为：

因为原问题的最优解和对偶问题相同，所以，对偶问题的最优解时目标函数为：

所以可以知道Y\*=，y此处即为影子价格，也就是乘子的值。被称为单纯形乘子，用π表示

乘子表示对应资源每增加一个单位时，原问题目标函数的相应增量。

因为列生成中在当前主问题求解后，可以得到当前的π（乘子），检验数为（表示非基），在确定新生成的列（子问题求出的新列即为想要加入主问题的新的一列的约束矩阵系数）是否能加入主问题时，由已有的列构成的即为当前的基，新求得的子问题的解即为新要加入主问题约束矩阵的最后一列，此时相当于非基，则其，其中π是刚求解出的，已知了，是求解的子问题的列，是最最原始问题（即RMP的原问题）中对应的目标函数系数，所以就可以判断出，从而确定是否要加入。