

开源 VRPLite 软件包：为交通运输规划服务的车辆路径问题解决方案

周学松

亚利桑那州立大学可持续工程与建筑环境学院, Tempe, AZ 85281, 美国
邮箱:xzhou74@asu.edu

佟路(Carol) *

北京航空航天大学电子信息工程学院, 北京 100091, 中国
国家综合交通大数据应用技术工程实验室, 北京 100091, 中国
邮箱:ltong@buaa.edu.cn

MonirehalsadatMahmoudi

亚利桑那州立大学可持续工程与建筑环境学院, Tempe, AZ 85281, 美国
邮箱:mmahmoudi@asu.edu

诸葛丽娟

北京交通大学交通运输学院, 北京 100044, 中国
邮箱:16120948@bjtu.edu.cn

姚宇

北京交通大学交通运输学院, 北京 100044, 中国
邮箱:yaoyu1@bjtu.edu.cn

张永祥

西南交通大学交通运输与物流学院, 四川成都 610031, 中国
邮箱:bk20100249@my.swjtu.edu.cn

商攀

清华大学土木工程系, 北京, 100084, 中国
邮箱:Shangp16@mails.tsinghua.edu.cn

刘江涛

亚利桑那州立大学可持续工程与建筑环境学院, Tempe, AZ 85281, 美国
邮箱:jliu215@asu.edu

石铁

西南交通大学交通运输与物流学院, 四川成都 610031, 中国
西南交通大学综合与智能交通国家联合工程实验室, 四川成都 610031, 中国
邮箱: tshi2005@my.swjtu.edu.cn

潘芋燕

北京工业大学城市交通学院, 北京, 100124, 中国
邮箱: 201539087@emails.bjut.edu.cn

摘要

近年来，电气化、自动化和共享化被认为是未来交通运输领域的“三大变革”，传统的定时公共交通系统将因此而大大增加服务的灵活性和车辆调度能力的自动性。许多新兴的定时运输应用包括城市轨道交通的全自动运营（FAO）系统、高铁的联合线路规划及时间表、以及新兴的自动驾驶车辆调度。车辆路径问题（VRP）有望找到一套最佳的车辆路线及时刻表来满足客户的需求，并在定制最优交通出行服务运输系统（中具有至关重要的作用。由于其很难在大规模实例网络中找到最佳解决方案，因此，大量的研究工作致力于开发高效的算法，而本文从基于时间和状态的路径搜索的独特视角思考。本文的 VRPLite 是基于时间和状态相关的车辆路径搜索框架，为解决交通按需运输问题而开发的一种开源的、轻量级的 VRP 程序。本文用高维网络（hyper network）来描述 VRPPDTW 的空间-时间-状态（space-time-state）的建模过程。该解决方案引擎框架中可以嵌入列生成或拉格朗日松弛方法，以解决众多常规问题。为了验证在各种交通条件和乘客出行需求下的路径搜索算法的有效性，本文提出了许多算例。

关键词

带接送的车辆路径问题，空间-时间-状态网络模型，列生成，拉格朗日松弛

1. 引言

近年来,随着人口的激增,人类的出行活动持续增长。由于交通系统资源数量的限制,交通拥堵问题仍为现今亟需解决的问题。未来交通规划的首要目标是融合多种需求管理策略和交通控制措施,从而实现交通流动性、环保性和可持续发展性。在缓解交通拥堵问题上,各种研究都基于减少因单独驾驶出行而造成的交通堵塞。在本研究中,我们主要提出了基于时间-状态的路径搜索引擎算法,以服务于按需供应的乘用车/城市交通在下一代按需分配交通系统中的应用。

经典的车辆路径问题(VRP)主要分为以下两种:(1)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,如图 1(c)所示;(2)在时间窗的约束下,在给定的起点和终点间接送乘客或货物 vehicle routing problem with pickup and delivery。上述两种问题均包含在 VRPLite 内,因此,我们下文中仅考虑第二种情况。在这里我们总结了 Cordeau et al.[1], Parragh et al.[2]和 Psaraftis et al.[3]等人发表的关于带有接送的车辆路径问题的精彩文章。当经典车辆路径问题(VRP)加入固定的上车点和下车点,便成了带有接送的车辆路径问题(VRPPD)。VRPPD 的实际应用在城市交通管理中非常常见,例如,轨道交通线路规划[4],制定出行决策[6,7],列车运行管理[8,9],列车时刻表制定[10,11],以及基于区域间的货运[12]。而在点对点合乘服务中,乘客可以选择是否与他人合乘。在数学领域中,合乘问题可以描述为考虑时间窗约束的针对乘客接送的车辆路径问题[13]。在以前的文献研究中,已对模型构建和算法求解做出较大贡献。但是,大规模的车辆路径问题的求解算法仍有待解决,尤其是高速公路瓶颈路段以及有信号配时的城市道路等有通行能力限制或者交通延误限制的区域。

在运筹学领域中,目前很少有文献研究交通延误对运输网络造成的潜在影响。然而,大多数文献定义 VRPPDTW 问题为:一个具体固定的起终点,及其距离最短或行程时间最短的有向图集合。由于大规模路网中的约束具有复杂性,求解大规模 VRP 问题变得十分困难。广义的 VRP 问题在城市交通、物流和调度问题上都有广泛的应用[14-15]如表 2(b)所示。例如,其中最具代表性的是车辆分配问题和车辆路径问题,其他问题还包括乘务员安排问题、飞机编队及路线问题、商船行程计划问题以及铁路时刻表编制问题。

本文提出的算法是基于在运输网络建模领域有广泛应用的最短路算法[16]。定义物理网络 (N, A) ,其中 N 是节点集合, A 为弧集合。最短路问题在于找到起点 o 到终点 d 之间费用最小的路径 $(i, j) \in A$ 。 $x_{i,j}$ 为 0-1 变量,表示是否有流量流经此路段 (i, j) 。最短路问题的流平衡约束定义如下[17]:

$$\min z = \sum_{(i,j) \in A} c_{i,j} x_{i,j} \quad (1)$$

s.t.

$$\sum_{\{j:(i,j) \in A\}} x_{i,j} - \sum_{\{j:(j,i) \in A\}} x_{j,i} = \begin{cases} -1 & \forall i = o \\ 1 & \forall i = d \\ 0 & \text{其他} \end{cases} \quad (2)$$

$$x_{i,j} \in \{0, 1\} \quad \forall (i, j) \in A \quad (3)$$

通常,解决最短路问题的算法可分为两大类:标号设定法以及标号修正法[17]。二者的本质区别在于反复更新节点的费用。节点的费用在标号设定法中被定义为定值,其不随迭代次数的改变而改变,而标号修订法中每一步的节点费用是变化的。因此,标号设定法不能解决弧费用为负的最短路问题,而标号修正法不受此限制。通过对弧 (i, j) 加入基于时间的弧费用 $c_{i,j,t}$,以及时间依赖的弧行程时间 $TT_{i,j,t}$,使得静态最短路问题动态化。Ziliaskopoulos 和 Mahmassani [18]提出了高效的标号修正算法解决从所有节点到同一终点的时间依赖的最短路问题。Chabini [19]旨在降低基于时间的动态规划最短路问题中最坏情况下的问题复杂程度。近年来,三维或多维空间的最短路问题在交通优化问题中成为热点。Mahmoudi 和 Zhou [20]提出了基于时间的前向动态规划算法,重新定义 VRPPD 问题为基于时间和状态的带有弧费用的最短路问题,其中状

态为车辆运载乘客的数量。Liu et al. [21]在此基础上，解决了日常出行活动安排问题，运用累积的空间-时间-状态减少复杂约束条件的数量。Zhou et al. [22]提出了基于时间-空间-速度的高维网络动态规划算法，并应用于铁路运行图和速度曲线的联合优化。Lu et al. [23]提出了考虑动力资源的资源-空间-时间网络，以此来解决时间依赖的路径问题。作为高维网络的其他表现形式，Tong et al. [14]和 Ruan et al. [24]提出了一种分解策略来解决定制公交问题，分解部分由两个耦合子问题组成，分别为：基于空间-时间网络的最短路问题和传统的指派问题。通过拓展基于空间-时间的与时间相关的最短路问题，VRPLite 包旨在解决基于时间和状态的最短路问题，其中乘客的累积服务状态作为 VRPLite 包中的状态维度。

表 1 对上文提到的网络表示方法、弧费用和数据结构方面的最短路算法进行了详细比较，表 2 (a) 中更进一步对比了空间-时间-状态模型框架下的不同状态维度的表示形式，以此展示该模型方法对不同交通难题的适用性。

本文中的开源 VRPLite 包可在 <https://github.com/xzhou99/VRPLite> 中下载。其主要贡献有以下三个方面：

- (1) 提供开源代码。交通运输研究人员和软件开发人员可将其功能扩大至不同的交通运输管理领域。
- (2) 提供免费的用于教育的模型框架，帮助学生和研究人员理解复杂的基于空间-时间-状态网络的交通优化问题的建模过程，例如：铁路运行图编制以及合乘应用等。
- (3) 介绍了时间依赖和状态依赖的最短路问题求解方案和相关的软件操作。

表 1

不同最短路算法之间的比较(网络表示、弧费用、数据结构、解决方法)

维度	节点	弧	状态	成本	网络建立	方法
空间维 (Ahuja) [17]	物理节点	物理弧	-	正或负	扫描合格的列表	标号修订法 (Label correcting)
	物理节点	物理弧	-	正	累计寻找最小的时间标记	标记设定法 (Label setting)
时空维, 以时间为导向 (Ziliaskopoulos) [18]	物理节点	物理弧, 与时间有关的弧费用	-	正或负	没有显性的建立时空网络	标记修订法 (Label correcting)
时空维, 以时间为导向 (Chabini) [19]	物理节点	物理弧, 与时间有关的弧费用	-	正或负	没有显性的建立时空网络	动态规划
时间、空间和状态维, 以时间和状态为导向 (Mahmoudi 和 Zhou) [20]	时间、空间及状态节点	时间、空间及状态连接	座位数, 乘客人数	正或负	没有显性的建立时空及状态网络	动态规划, 但需要枚举车辆运载状态
时间、空间和状态维, 以时间和状态为导向(本文)	时间、空间及状态节点	时间、空间及状态连接	用 0,1,2 表示乘客的累积服务状态	正或负	没有显性的建立时空及状态网络	动态规划, 动态生成乘客累积服务状态

表 1

(a) 典型的状态表示法

状态	应用	相关文章
车辆携带乘客状态	VRPPDTW	Mahmoudi and Zhou [20]
乘客累积服务	VRPPDTW	Mahmoudi et al. [25]
资源, 例如能源消耗或车辆排放	绿色 VRP	Lu et al. [23]
出行活动执行状态	家庭出行活动模式问题	Liu et al. [21]
高速列车的速度	列车调度	Zhou et al. [22]
机组调度的累积工作时间, 动车组 (EMU) 维修调度的累积行驶距离	船员调度, 动车组维修调度	Chen et al. [26]

(b) 广义 VRP 问题总结对比

出版者	问题类型	问题性质	方法(算法)	目标函数
Lu (Carol) Tong (2017) [14]	公交定制服务 VRP	线性	拉格朗日分解	最小化乘未被服务的乘客数和路径费用
Huimin Niu 和 Zhou (2018) [15]	TVSP	非线性	拉格朗日启发式 算法	最小化不同时空位置的车辆运行延误

2. 基于空间-时间-状态的模型描述

2.1. 问题陈述

本文研究的 VRPPDTW 问题可以用下列语句正式定义。考虑一个具有有限节点集 N 和有限路径集 M 的物理运输网 (N, M) , 其中节点 $i, j \in N$, 有向路径 $(i, j) \in M$ 。根据最优化原则, 在一个时空网络 $G = (E, A)$ 中, 可以在计划时间范围 T 内由运输网络 (N, M) 构建, 并考虑载客状态 W , 有限的时空状态顶点集合 V 和有限空间-时间-状态弧集合 A 。在空间-时间-状态问题中, 每个顶点 $(i, t, w) \in V$ 同时表示时间、车辆位置和车辆载客状态; 每个弧 $(i, j, t, s, w, w') \in A$ 表示在时间 t 时处于节点 i 的具有 w 状态的乘客, 当变化为时间 s 时处于节点 j 并具有状态 w' 时所经历的空间-时间-状态路径变化。由于三维网络结构, 很容易模拟车辆的出行位置和车辆的载客状态随乘客的出行需求的变化而变化。VRPPDTW 问题主要表示为, 在特定的交通条件下, 给出一组乘客 P 和他们的出行需求, 以及上车/下车的位置 o_p/d_p , 时间窗 $[a_p, b_p]$ 和 $[a'_p, b'_p]$, 以及车辆的载客能力约束 Cap_v 和其他的路径约束, 为车辆组 V 中的每辆车 v 匹配最佳的乘客并分配车辆, 同时给出出行路线和时间表。表 3 和表 4 为本文中使用的符号及定义。

表 2 VRPPDTW 问题中的变量符号及定义

符号	定义
V	物理车辆集合
V^*	虚拟车辆集合
P	乘客集合
N	物理网络中的节点集合
M	物理网络中的路径集合
W	载客状态集合
v	物理车辆

v_p^*	专用于乘客 p 的虚拟车辆
p	物理乘客
w	载客状态
(i, j)	相邻节点 i 和 j 之间物理路径
$TT(i, j, t)$	在时间 t 开始从节点 i 到节点 j 的行程时间
Cap_v	车辆 v 的最大载客量
a_p	乘客 p 在起点时的最早出发时间
b_p	乘客 p 在起点时的最晚出发时间
a'_p	乘客 p 到达目的地时的最早到达时间
b'_p	乘客 p 到达目的地时的最晚到达时间
$[a_p, b_p]$	乘客 p 在起点时的出发时间窗
$[a'_p, b'_p]$	乘客 p 在终点时的到达时间窗
o'_v	车辆 v 在起点的虚拟节点
d'_v	车辆 v 在终点的虚拟节点
e_v	车辆 v 从起点车库出发的最早时间
l_v	车辆 v 到终点车库到达的最晚时间
o_p	乘客 p 起点的虚拟节点（乘客 p 的上车节点）
d_p	乘客 p 终点的虚拟节点（乘客 p 的下车节点）

表 3 所构建模型中的符号及定义

符号	定义
$(i, t, w), (j, s, w')$	节点的空间、时间、状态指标
(i, j, t, s, w, w')	时空状态弧，表示从节点 i 到节点 j ，从时间 t 到时间 s ，从状态 w 到状态 w' 的变化
V	物理车辆集合
V^*	虚拟车辆集合
B_v	车辆 v 的行驶网络中的时空状态弧集合
$c(v, i, j, t, s, w, w')$	车辆 v 通过弧 (i, j, t, s, w, w') 的成本
$TT(v, i, j, t, s, w, w')$	车辆 v 通过弧 (i, j, t, s, w, w') 的行程时间
$\Psi_{p,v}$	网络中车辆 v 接送乘客 p 的服务弧集合
$y(v, i, j, t, s, w, w')$	如果车辆 v 经过了弧 (i, j, t, s, w, w') ,那么此变量值为 1, 否则为 0

2.2. 基于空间-时间-状态网络表示的建模方法

现在我们用一个以往研究[20]中使用过的简单示例，来演示基于空间-时间-状态网络建模的关键点。一个物理交通网络包含六个节点，该网络中的每条径路都和时变的行程时间 $TT(i, j, t)$ 相关。两个乘客具有相同的上车节点（节点 2）和下车节点（节点 3），但其出发、到达的时间窗并不相同，并且只用一辆车来完成实现这两个出行请求任务。我们假设车辆从节点 4 出发，最终回到节点 1，并为每个乘客建立一个虚拟车辆出发点和虚拟车辆到达点以及对应的乘客上、下车节点。如图 1 所示，车辆应该在时间窗[4, 7]内接上乘客 1 从虚拟节点 o_1 上车，并在时间窗[11, 14]内送乘客 1 到虚拟节点 d_1 下车；还应该时间窗[8, 10]内接上乘客 2 从虚拟节点 o_2 上车，并在时间窗[13, 16]内送乘客 2 在虚拟节点 d_2 下车。车辆 1 从起点车库出发的最早发车时间为 $t=1$ ，其到达终点车库处的最晚到达时间为 $t=20$ 。

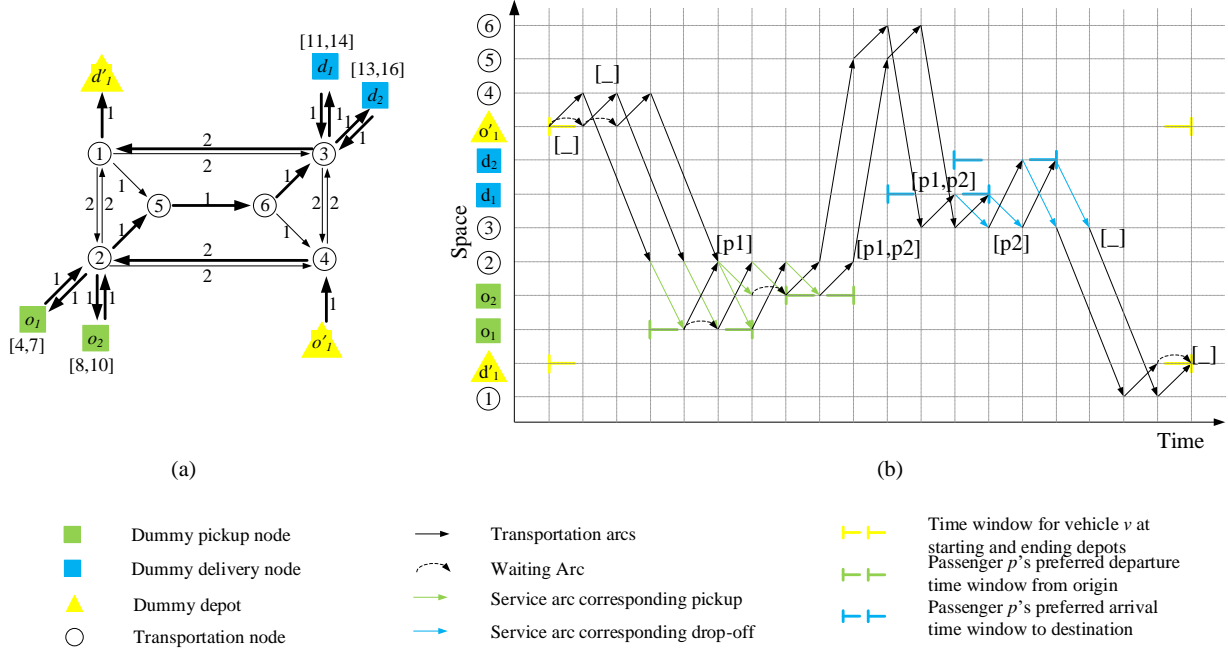
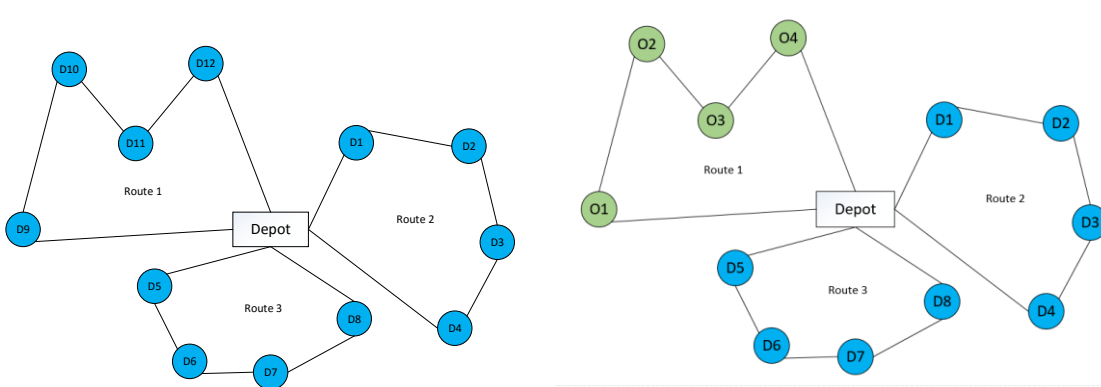


图 1 (a)带有相应虚拟节点的六点交通网络，其中[]代表乘客时间窗；(b) 时空网络中的最短路径，其中 [], [p1], [p1, p2]和[p1, p2]代表车辆的载客状态[20]



(c) 单程的送人或接人的车辆路径描述 (d) vehicle routing with pickup and delivery 的车辆路径描述

请注意，图中的粗体箭头表示当车辆 1 服务于两个乘客的出行需求时，其从起点车库到终点车库的节点序列变化情况($o'_1, 4, 2, o_1, 2, o_2, 2, 5, 6, 3, d_1, 3, d_2, 3, 1, d'_1$)。为了便于用空间-时间-状态模型表达，将时间范围离散化为具有相同时间单位的一组时间间隔，例如 1 分钟。图 1 (b) 表示在时空网络中，车辆 1 从起点到目的地的最短路径经过的弧。

每辆车从车辆载客人数为零的空车状态开始行驶，我们把这个空车状态称为初始状态 w_0 。所构建的时空状态网络中的每个顶点由三个不同元素组成：节点 j ，时间间隔 t 和载客状态 w 。

在时空状态网络中，我们可以看出从时间 t 、节点 i 、载客状态为 w 出发，到达时间 s 、节点 j 、载客状态 w' 的弧(i, j, t, s, w, w')。因此，在空间-时间-状态网络中，每个顶点(i, t, w)通过弧(i, j, t, s, w, w')连接到顶点(j, s, w')。

基于时空状态网络的 VRPPDTW 模型最早由 Mahmoudi and Zhou [20]提出。公式 (4) - 公式 (9) 给出了 VRPLite 中的模型。更详细的信息，请参阅原始研究论文，感兴趣的读者可以通过访问 https://github.com/xzhou99/VRPLite/tree/master/GAMS_SourceCode 找到 GAMS 源代码

目标函数：

$$\min Z = \sum_{v \in (V \cup V^*)} \sum_{(i,j,t,s,w,w') \in B_v} c(v,i,j,t,s,w,w') y(v,i,j,t,s,w,w') \quad (4)$$

车辆 v 起点的流量平衡约束：

$$\sum_{(i,j,t,s,w,w') \in B_v} y(v,i,j,t,s,w,w') = 1 \quad i = o'_v, t = e_v, w = w' = w_0, \forall v \in (V \cup V^*) \quad (5)$$

车辆 v 终点的流量平衡约束：

$$\sum_{(i,j,t,s,w,w') \in B_v} y(v,i,j,t,s,w,w') = 1 \quad j = d'_v, s = l_v, w = w' = w_0, \forall v \in (V \cup V^*) \quad (6)$$

中间点的流量平衡约束：

$$\sum_{(j,s,w'')} y(v,i,j,t,s,w,w'') - \sum_{(j',s',w')} y(v,j',i,s',t,w',w) = 0 \quad (i,t,w) \notin \{(o'_v, e_v, w_0), (d'_v, l_v, w_0)\}, \forall v \in (V \cup V^*) \quad (7)$$

乘客 p 的乘车需求约束：

$$\sum_{v \in (V \cup V^*)} \sum_{(i,j,t,s,w,w') \in \Psi_{p,v}} y(v,i,j,t,s,w,w') = 1 \quad \forall p \in P \quad (8)$$

定义 0-1 变量约束：

$$y(v,i,j,t,s,w,w') \in \{0, 1\} \quad \forall (i,j,t,s,w,w') \in B_v, \forall v \in (V \cup V^*) \quad (9)$$

2.2 协同服务下的

2.3. VRPLite 软件包中的不同优化模型

在实例中解决大规模的运输优化问题具有挑战性，这需要我们具有创新思维和技术进步。我们的 VRPLite 软件包解决了具有时间维度的路径规划和调度问题，并结合了一些理论模型和实际算法，例如 VRP 问题，指派问题，背包问题，时空依赖的问题，拉格朗日松弛算法和列生成算法。

(1) 车辆路径问题

VRP 问题是一类具有两层特殊结构的线性规划问题。具体来说，在货运方面，每个货源都供应固定数量的货物，货物必须被运送到目的地。每个目的地都存有从货源送来的固定数量的货物。类似地，运输需求问题或 VRPPDTW 乘客运输问题中，在复杂的交通网络中车辆都需要从起点出发将乘客运送至目的地。

(2) 指派问题

指派问题就是众多指派对象和指派任务之间的匹配，任何一个指派对象可以执行完成任何一个指派任务，指派对象与指派任务之间的不同分配，会产生不同的费用成本。指派问题便是在遵循特定规则的条件下，给所有指派对象分配指派任务的过程，从而完成任务的同时使总成本最小。在 VRPPDTW 问题中，为了满足乘客的出行需求，需要指派车辆给乘客。通过 Fisher [27] 引入的变量分离算法，复杂的时空状态变量可以分离为指派车辆给乘客的指派决策变量以及车辆的时空路径决策变量。变量分离法的优点在于避免列举车辆的载客状态。因此，基于 VRP 模型的指派问题在处理定制公交、公共交通等多载客量的运输问题方面具有优越性。

(3) 背包问题

背包问题指的是如何在背包里装入最有价值或最有用的物品，但所装物品数量受背包容量的限制。在 VRPPDTW 中，由于车辆载客能力或载客时间的限制，我们需要确定具体每辆车应该为哪些乘客提供运输服务。

(4) 基于时间-状态的最短路问题

在图论中，最短路径问题是在图中两个顶点（或节点）之间寻找路径的问题，使得路径组成权重之和最小。感兴趣的读者可以在文献[17,18]中学习更多的关于时空网络构建和基于时间-状态下的最短路径问题的知识。在 VRPPDTW 问题中，我们需要计算每辆车在时空状态下的最短路径。

(5) 动态规划算法

我们已经开发了几种有效的算法，在基于时间的弧成本网络中计算最短路径。在路径搜索引擎中，我们使用时间索引动态规划算法来解决最短路径问题。

(6) 拉格朗日松弛算法

拉格朗日松弛算法是一种利用一个比较简单的问题逐渐逼近带有复杂约束的优化问题的松弛算法。为了找到拉格朗日对偶问题的最优解，VRPLite 通过调用前文提出的基于时间的前向递推动态规划算法，通过不断改变拉格朗日乘子的值来计算每辆车在时空状态下的最短路径。

(7) 列生成算法

列生成算法是求解大规模的线性规划问题的有效方法。列生成的核心思想是将最初的问题分解为两个问题：主问题和子问题。主问题是在仅考虑一部分变量情况下的原问题，子问题（价格问题）是一个为找出加入主问题的新变量而建立的新问题。因此 VRPPDTW 问题可以分为列分割问题和基于时间的最短路径问题，这分别可以通过标准优化算法和前文提出的基于时间的动态规划算法来解决。

3. 软件架构，数据流及实施

3.1. 数据流程图

VRPLite 中设计的软件架构旨在将乘客与车辆之间的分配和基于时间的路段集成到开源 VRP 建模软件包中。如图 2 所示，三个黄色框图代表模型的主要组成部分，包括：（1）基于节点-路径的网络结构和车辆状态变化的动态规划；（2）拉格朗日松弛模块，利用乘客价格确定车辆在（1）动态规划模块中的接和送；（3）列生成大规模问题（即车辆路径）。最终输出结果包括乘客的费用/利润（即总等待时间，运输费用等），乘客和车辆之间的分配情况以及车辆的时空轨迹。

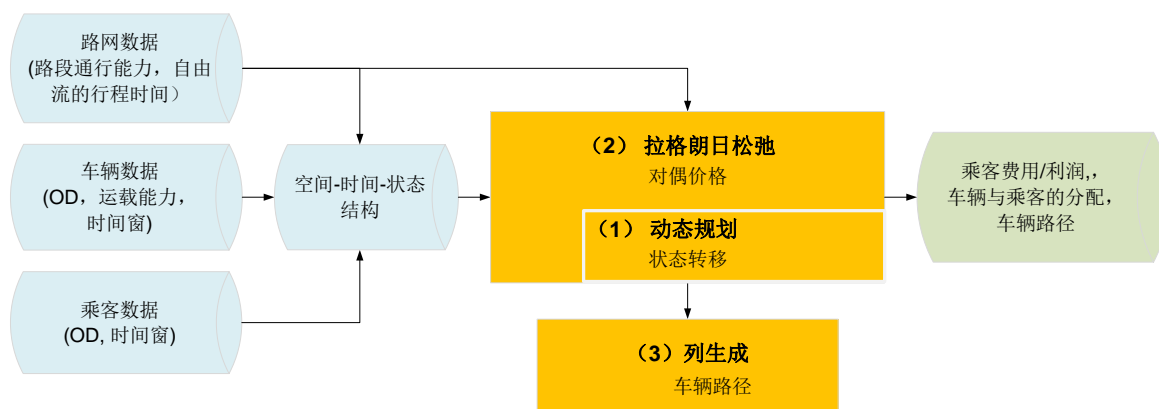


图 2 软件体系结构与模型的关键组成部分

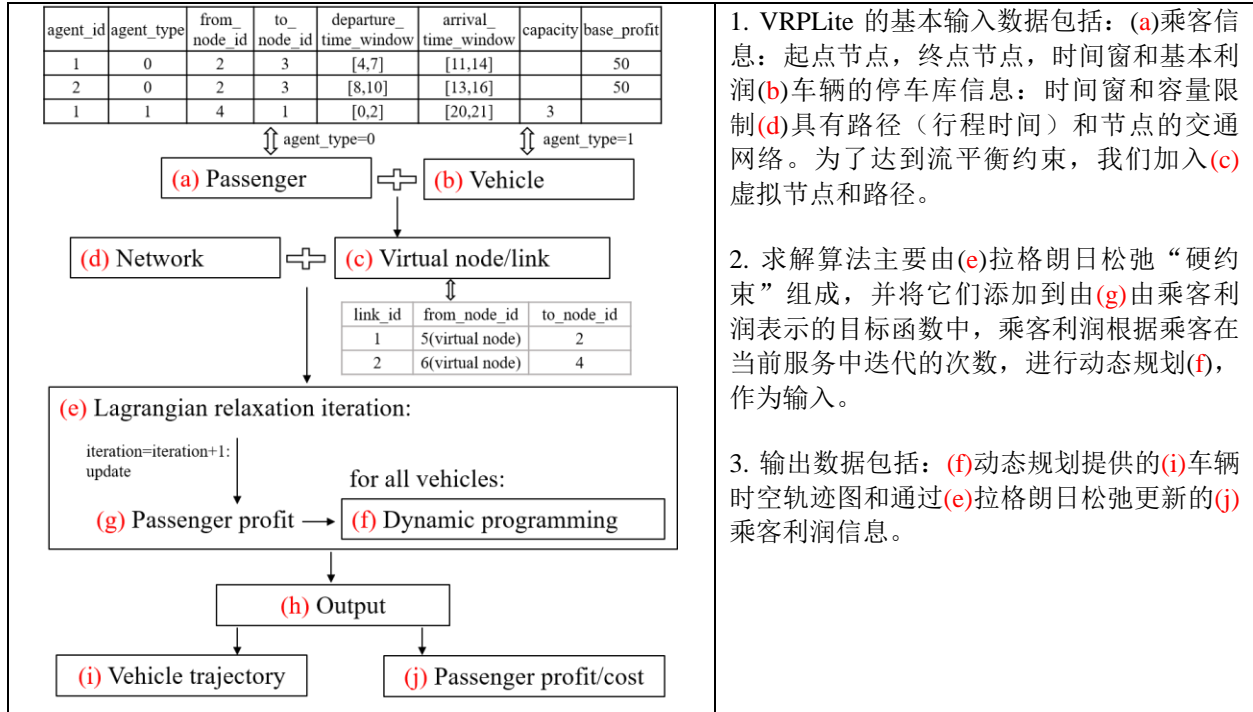


图 3 VRPLite 的数据流程图

3. 2. 通过动态编程实现集束搜索 (beam search) 过程的方案框架

图 3 为 VRPLite 的动态编程 (DP) 解决方案框架。总体而言, 动态规划方法适用于给定的多步或顺序的状态转换决策过程。在我们的示例中, 时间范围充当 DP 中的“阶段”, 并且在当前节点和当前时间, (累积服务) 状态由车辆的路径节点序列, 路径时间序列和服务乘客序列来定义。

图 4 描述了基于时间的集束搜索算法的伪代码, 与 Mahmoudi 和 Zhou [20] 的三阶段动态规划算法相比, 它是一个改进版本。本质上, 车辆从出发时刻开始从停车库出发, 并使用时间指标, 集束搜索的 k 指数和从当前节点往外搜索进行多个扫描循环。在每一次循环中, 所有可能的状态都通过目标函数进行评估, 并选择最佳的K继续进行。需要注意的是, 当前节点被存储在时间索引向量 $td_state[t][k]$ 中, 并且仅选择 k 个最佳解向前继续进行搜索。当车辆在其时间窗内到达目的地时输出最终解决方案。图 5 描述了基于时间集束搜索算法的过程。

Initialize vector $td_state[t0] = [o]$, $end_state[v] = []$, $KBestSize$, build virtual node for each passengers' and vehicle's origin-destination node

Do while t in $(T0, T)$

Sort $td_state[time\ t]$ according to the overall cost (transportation cost + profit for serving passengers+ passenger and vehicle waiting cost)

Do while index $k < \min(td_state[t0].size(), KBestSize)$

$current_node = td_state[t0][k].current_node$

Do while to_node in $current_node.outbound_node_vector.size()$

$to_node_time = t + outbound_link's\ travel\ time\ when\ entering\ link\ at\ time\ t$

Case 1: to_node is a dummy pick up or drop-off node and the arrival time to_node_time is within the service time window. If passenger p is not in $td_state[t0][k]$, then v can pick up passenger p ; if p has been in vehicle v 's carrying state $td_state[t0][k]$, then v would drop off passenger p .

此处 $[to][k]$ 是指什么, 如果是指已经的, 那么说明只要前面有 t 下的node结果乘客了, 则就之后的时间下的点内就不能接乘客了?

Update the vehicle carrying states in $td_state[to_node_time]$ with its $current_node = to_node$ and new cost with possible service profit.

Case 2: to_node is a physical node.
Update $td_state[to_node_time]$ with its $current_node = to_node$ and updated cost.

Case 3: to_node is vehicle's destination node d .
Update $end_state[v]$ with $td_state[to_node_time]$ and updated cost.

End // downstream node

End // index k

End // time t

Sort $end_state[v]$ according to its cost, choose the first one as our solution.

Parameter and variable definitions:

td_state : vehicle's time-dependent state

end_state : vehicle's ending state at destination

to_node_time : vehicle arrival time at the downstream node of the current node.

Index k : the index for beam search with the $KBestSize$ as beam width

图 4 每个 v 的时间索引集束搜索过程的伪代码，其离开起点时间为 T_0 ，到达终点 d 的时间为 T

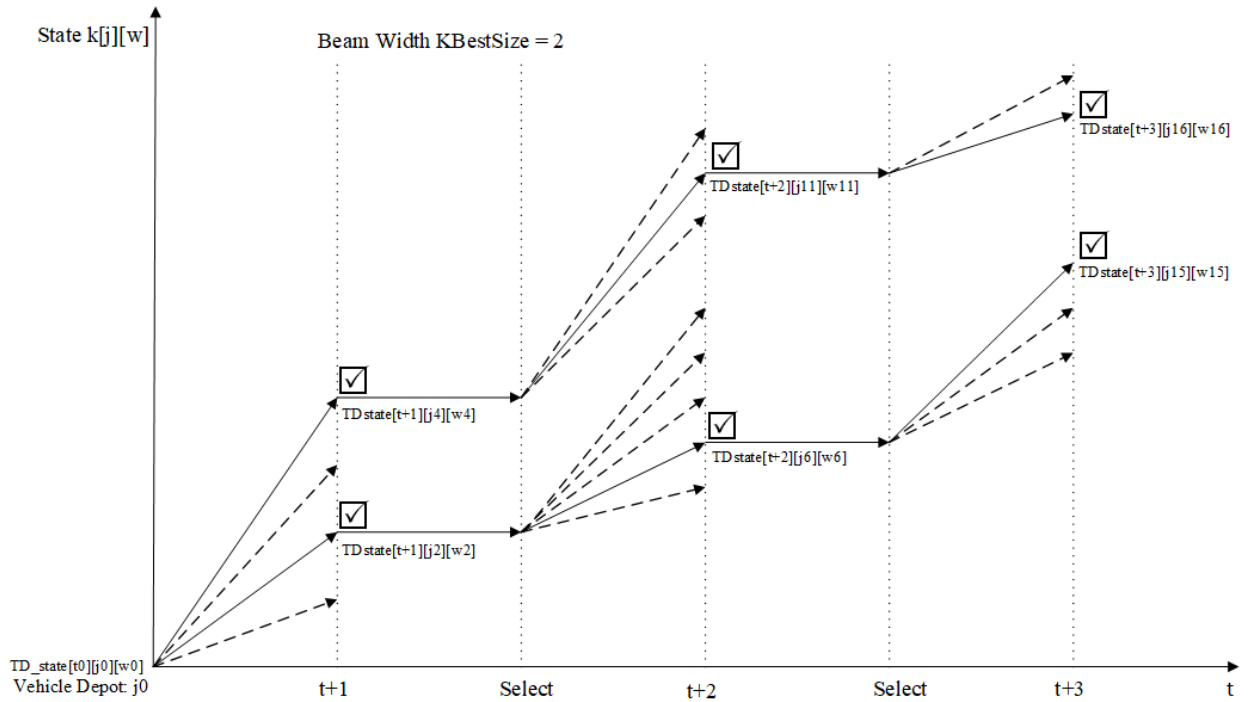


图 5 基于时间的集束搜索算法过程，以时间维度为横轴

为了满足模型中的约束 (8)，我们引入拉格朗日松弛方法来松弛掉这个约束，在新的目标函数中增加乘客利润（即对偶价格）。通过不断迭代，乘客利润根据次梯度方法更新，即检查在当前迭代期间乘客被服务了多少次从而相应地增加或降低价格。对于大规模应用，我们使用多辆车列生成的过程来更好地定义和搜索可行的解决方案。

3.3. 列生成框架

通过应用 Dantzig-Wolfe 分解, VRPPDTW 问题也可以将原始的基于弧的形式, 转化为集分割问题重新表达。我们可以引入一组新的基于路径的变量 $x(v, p)$, 如果乘客 p 由车辆 v 服务则等于 1, 否则等于 0。VRPPDTW 的集分割形式的模型可表示如下。

$$\text{Min } Z = \sum_{p \in P} \sum_{v \in V} c(v) x(v, p) \quad (10)$$

s.t.

$$\sum_{v \in V \cup V^*} x(v, p) = 1 \quad \forall p \in P \quad (11)$$

$$x(v, p) = \{0, 1\} \quad \forall p \in P, v \in V \quad (12)$$

目标函数 (10) 为最小化所选路径的成本。约束条件 (11) 保证每个乘客只被服务一次。式 (10) - (12) 的线性规划 (LP) 松弛称为限制性主问题 (RMP), 可以通过标准优化算法来解决。我们也可以从 RMP 的求解结果中获得每个乘客 $\pi(p)$ 的对偶价格。然后, 这些对偶价格被用于子问题的目标函数中, 子问题用来搜索可以使得成本降低的新路径。目标函数可以表示为式 (13)。

$$\text{Min } \sigma = \sum_{(i, j, t, s, w, w')} c(v, i, j, t, s, w, w') y(v, i, j, t, s, w, w') - \sum_p \pi(p) x(v, p) \quad (13)$$

价格子问题本质上是网络中——基于时间弧成本的最短路径问题, 目前已经开发了几种有效的算法来解决该问题。基于此, 我们可以在动态规划框架中, 运用基于时间和状态的路径搜索引擎的方法来计算一般列生成框架内的子问题。

如图 6 所示, 通过解决限制主问题, 输出乘客对偶价格的最优解并作为定价子问题的输入。经过多次迭代, 定价子问题被求解, 并产生车辆的新路径, 作为要添加到主问题中的新列, 最后, 得到最优车辆路线。有关列生成算法的更多细节, 特别是关于分支和价格框架及其最终收敛标准, 感兴趣的读者可以参考阅读 Lübbecke 和 Desrosiers [28]。

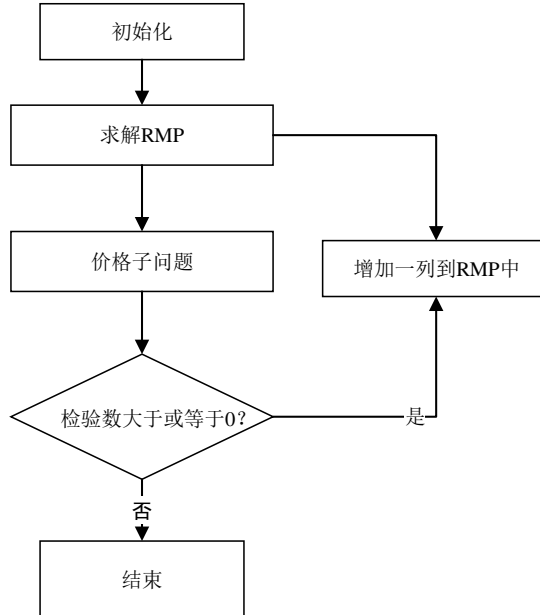


图 6 路径搜索的列生成框架

在列生成算法中, RMP 被松弛为线性规划问题, 因此我们获得的最优解可能是分数。然而, 表示车辆 v 是否为乘客 p 提供服务的变量 $x(v, p)$ 应该是二进制整数。因此, 我们必须设计分支策略来找到可行的整数解

决方案。当算法在变量 $x(v,p)$ 上分支时，它表明 $x(v,p)$ 是分数，并且通常分为两个分支 $x(v,p) = 0$ 和 $x(v,p) = 1$ 来求解。

4. 各种测试算例

4.1. 第一个例子：干道式网络

如图 7 所示，由 6 个物理节点和 5 条物理路径，6 个虚拟节点和 10 条虚拟路径组成的小型走廊式网络。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。此外，出发和到达时间窗直接在特定的虚拟节点的旁边标注。在本算例中，一辆车经过节点 1 到达节点 6，为两名乘客提供出行服务。第一位乘客 A 从节点 2 上车出发，并在节点 4 下车，而第二位乘客 B 从节点 3 上车，并在节点 5 下车。按照我们的费用成本更新规则，图 7 中所示的内部节点的数量及标注在表 5 中做详细解释。在图 7，图 9，图 11 和图 13 中的圆圈，矩形和三角形分别不同类型的节点编号;路径上面的数字代表相应的行程费用。

表 4
基本示例中的节点内部编号

节点名	内部编号	类型
1	1	物理节点
2	2	物理节点
3	3	物理节点
4	4	物理节点
5	5	物理节点
6	6	物理节点
o_1	7	虚拟节点, 乘客 A 的接乘点
d_1	8	虚拟节点, 乘客 A 的送达点
o_2	9	虚拟节点, 乘客 B 的接乘点
d_2	10	虚拟节点, 乘客 B 的送达点
o'_1	11	虚拟节点, 车辆 1 的起点车库
d'_1	12	虚拟节点, 车辆 1 的终点车库

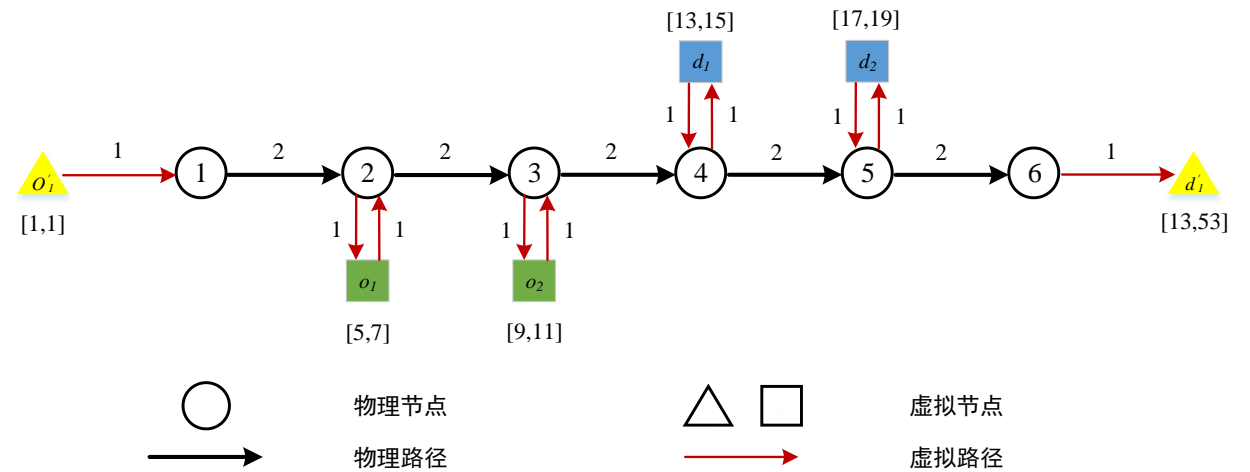


图 7 基本示例的网络布局和时间窗口设置图示

本示例结果的上界和下界都等于 20，因此证明最优解为 20。另外，单车的最优路线和时间序列如表 6 所示，图 8 所示为基本示例的车辆时空轨迹图。应当指出，在该示例中没有乘客等待时间或车辆等待时间，因为车辆可以在服务时间窗刚开始的时候到达服务点进行服务。

表 5 基本示例中路径的节点序列和时间序列变化情况

迭代次数	步长	节点序列	时间序列
0	1	11;1;2;3;4;5;6;12;	1;2;4;6;8;10;12;13;
1	1	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
2	0.5	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
3	0.333333	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
4	0.25	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
5	0.2	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
6	0.166667	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
7	0.142857	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
8	0.125	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
9	0.111111	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
上界:		11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;

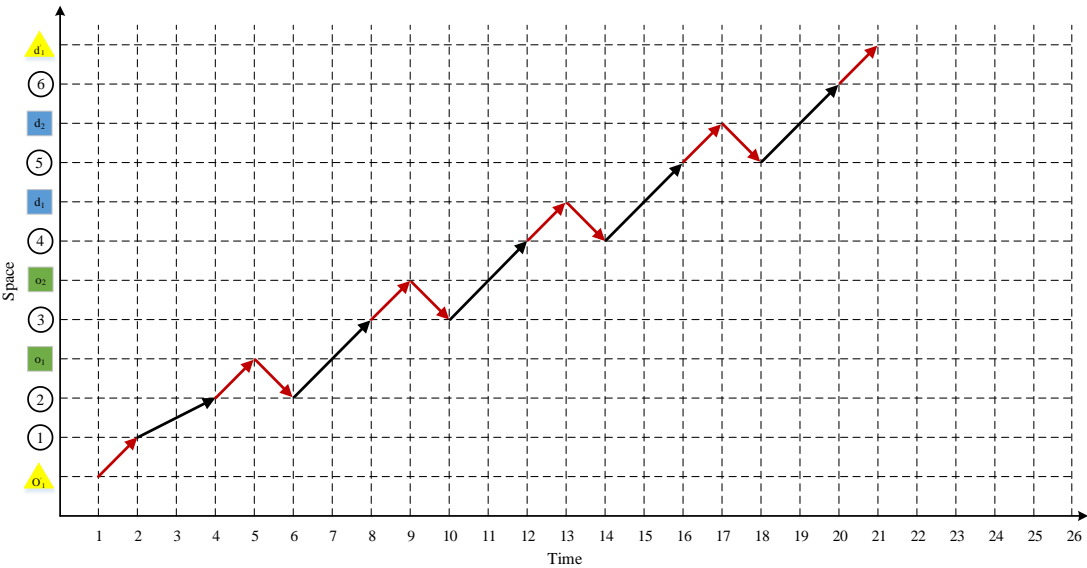


图 8 基本示例的车辆时空轨迹图

4.2. 第二个例子：延长路径行程时间

如图 9 所示，路径 (2, 3) 和 (4, 5) 的行程时间延长至 4，如表 7 所示为相应地更新乘客 A 和 B 的出发和到达时间窗，但并不引入车辆或乘客的等待时间，图 10 表示车辆的时空轨迹。

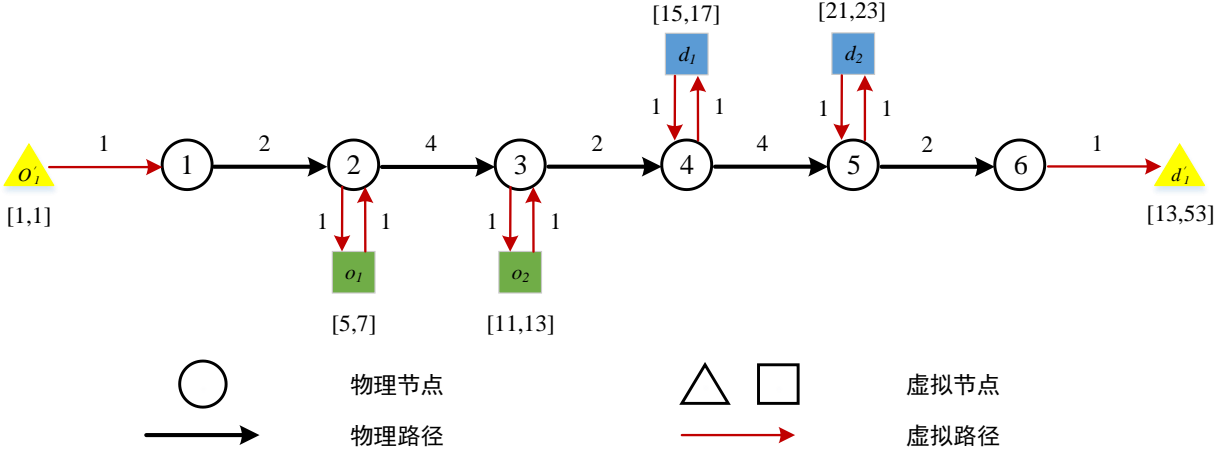


图 9 延长路径行程时间的网络布局和时间窗图示

该问题结果的上界和下界都等于 24，与第一个例子的最优解相比，增加了 4。显然，这是因为路径 (2, 3) 和 (4, 5) 的行程时间增加了 4，并且没有引入额外的车辆或乘客等待时间。

表 6 延长路径行程时间的算例中路径节点序列和时间序列的变化情况

迭代次数	步长	节点序列	时间序列
0	1	11;1;2;3;4;5;6;12;	1;2;4;6;8;10;12;13;
1	1	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
2	0.5	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
3	0.333333	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
4	0.25	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
5	0.2	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
6	0.166667	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
7	0.142857	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
8	0.125	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
9	0.111111	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;
上届:		11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	1;2;4;5;6;8;9;10;12;13;14;16;17;18;20;21;

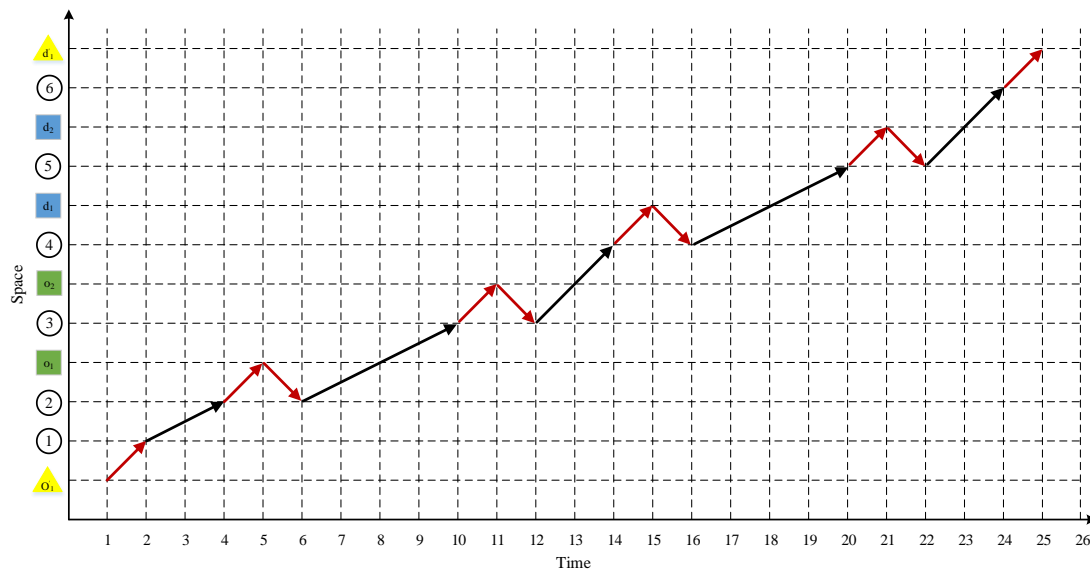


图 10 延长路径行程时间的车辆时空轨迹图

4.3. 第三个例子：增加车辆等待时间

为了测试车辆等待时间对最佳车辆路线和时刻表的影响，乘客 A 的出发时间窗从[5, 7]延误到[10, 12]。因此，车辆必须在节点 o_1 处等待，直到到达乘客 A 准备出发的时刻 10。此外，乘客 A 的到达时间窗和乘客 B 的时间窗也相应地做改动，使得车辆不会在其他地方等待。如图 11 所示为更新后的时间窗设置。

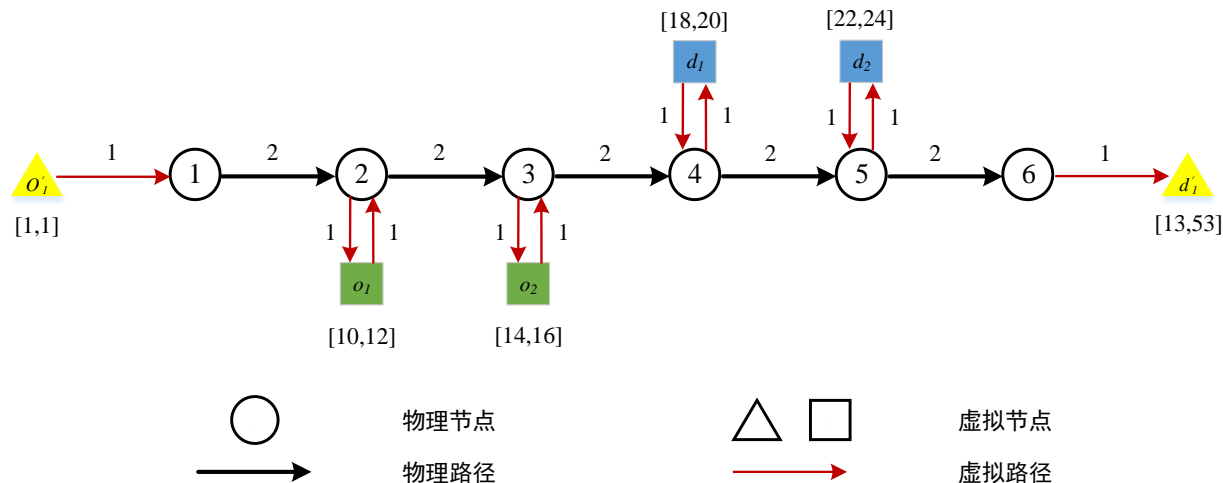


图 11 加入车辆等待时间的路径节点序列和路径时间序列的图示

该算例结果的上界和下界等于 22.5，这是最优解。如表 8 所示为车辆的最佳路线和时刻表，车辆的时空轨迹如图 12 所示。可以看出，车辆在时间 5 到达节点 o_1 并等待直到时间 10 开始发车，所以车辆等待时间为 5。另外，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 $25 - 0.5 * (10 - 5) = 22.5$ 。

表 7 加入车辆等待时间的算例中的路径节点序列和时间序列的变化情况

迭代次数	步长	节点序列	时间序列
------	----	------	------

0	1	11;1;2;3;4;5;6;12;	1;2;4;6;8;10;12;13;
1	1	11;1;2;7;7;2;3;9;3;4;8;4;5;10;5;6;1	1;2;4;5;10;11;13;14;15;17;18;19;21;22;23;25
2	0.5	2;	;26;
3	0.333333	11;1;2;7;7;2;3;9;3;4;8;4;5;10;5;6;1	1;2;4;5;10;11;13;14;15;17;18;19;21;22;23;25
4	0.25	2;	;26;
5	0.2	11;1;2;7;7;2;3;9;3;4;8;4;5;10;5;6;1	1;2;4;5;10;11;13;14;15;17;18;19;21;22;23;25
6	0.166667	2;	;26;
7	0.142857	11;1;2;7;7;2;3;9;3;4;8;4;5;10;5;6;1	1;2;4;5;10;11;13;14;15;17;18;19;21;22;23;25
8	0.125	2;	;26;
9	0.111111	11;1;2;7;7;2;3;9;3;4;8;4;5;10;5;6;1	1;2;4;5;10;11;13;14;15;17;18;19;21;22;23;25
上届:		2;	;26;

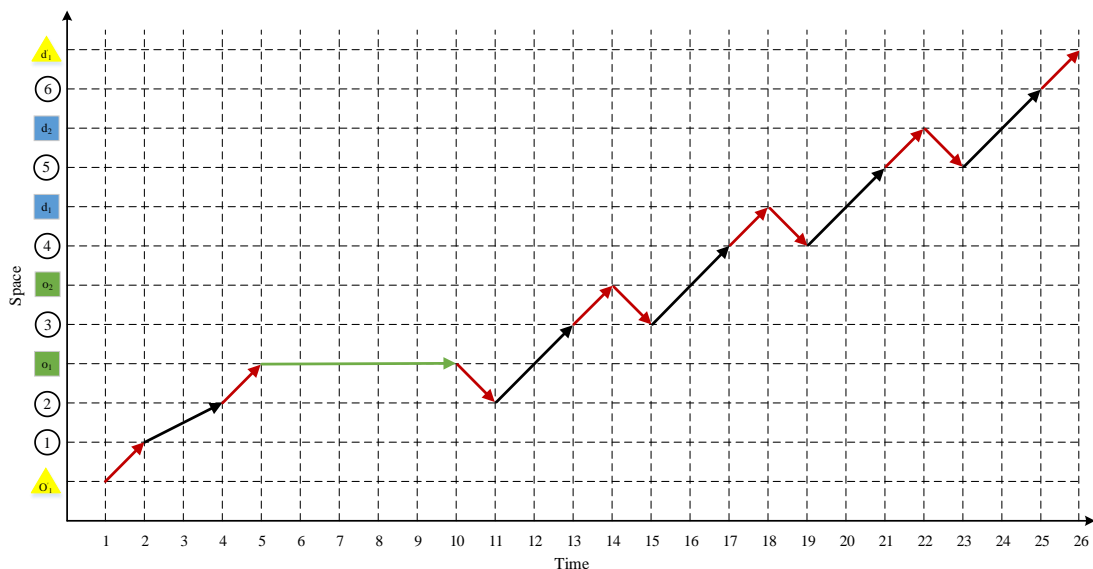


图 12 加入车辆等待时间的车辆时空轨迹图

4.4. 第四个例子：增加乘客等待时间

如果车辆从车库出发产生延误，那么乘客将不得不等待车辆到达。因此，与基本示例相比，车辆的出发时间窗被延误 5，并且这两个乘客的出发和到达时间窗被相应地调整。新的时间窗变化情况如图 13 所示。

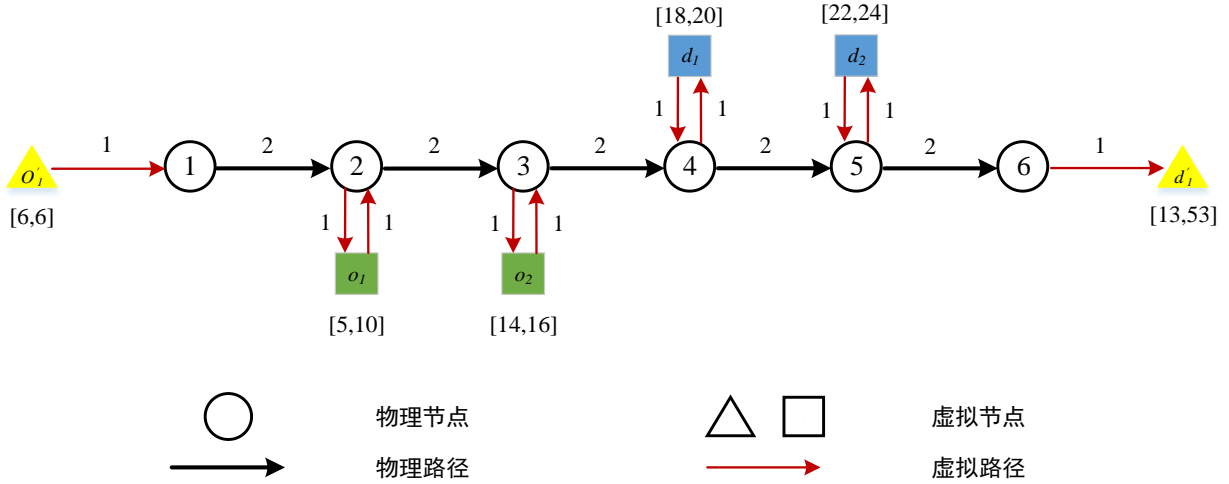


图 13 加入乘客等待时间的路径节点序列和路径时间序列的示例

这个例子的上界和下界等于 21.5，并且车辆的最佳路线和时间表如表 9 所示，车辆的时空轨迹如图 14 所示。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 $20 + 0.5 \times 5 = 21.5$ 。

表 8 加入乘客等待时间的算例中的路径节点序列和时间序列的变化情况

迭代次数	步长	节点序列	时间序列
0	1	11;1;2;3;4;5;6;12;	6;7;9;11;13;15;17;18;
1	1	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	6;7;9;10;11;13;14;15;17;18;19;21;22;23;25;26;
2	0.5	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	6;7;9;10;11;13;14;15;17;18;19;21;22;23;25;26;
3	0.333333	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	6;7;9;10;11;13;14;15;17;18;19;21;22;23;25;26;
4	0.25	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	6;7;9;10;11;13;14;15;17;18;19;21;22;23;25;26;
5	0.2	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	6;7;9;10;11;13;14;15;17;18;19;21;22;23;25;26;
6	0.166667	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	6;7;9;10;11;13;14;15;17;18;19;21;22;23;25;26;
7	0.142857	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	6;7;9;10;11;13;14;15;17;18;19;21;22;23;25;26;
8	0.125	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	6;7;9;10;11;13;14;15;17;18;19;21;22;23;25;26;
9	0.111111	11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	6;7;9;10;11;13;14;15;17;18;19;21;22;23;25;26;
上届:		11;1;2;7;2;3;9;3;4;8;4;5;10;5;6;12;	6;7;9;10;11;13;14;15;17;18;19;21;22;23;25;26;

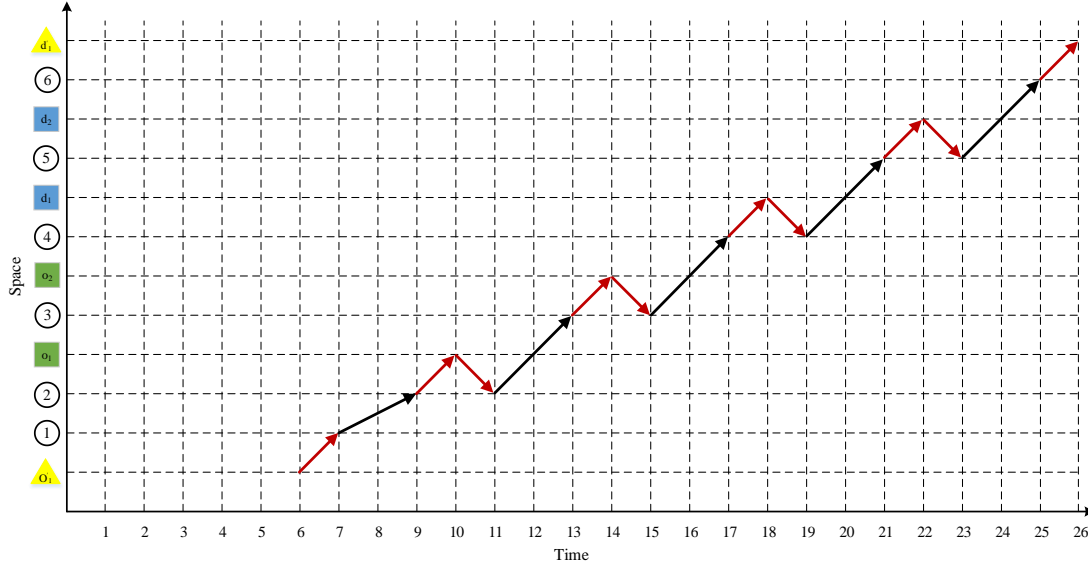


图 14 加入乘客等待时间的车辆时空轨迹图

4.5. 第五个例子：列生成过程中的分支过程

我们在图 1 中所示的六节点网络图中测试了我们的列生成算法，其中包括三名乘客和两辆车。表 10 所示点对为出发地和目的地，以及乘客的出发和到达时间窗。缩写“TW”和“TH”分别代表时间窗和时间范围。这个例子的代码和相关数据可以在 <https://github.com/YaoYuBJTU/VRPLite-python> 中下载。

表 9 乘客的起终点点对和对应的出发和到达时间窗

o_1	d_1	o_2	d_2	o_3	d_3	o'_1	d'_1	o'_2	d'_2
2	6	2	3	5	3	4	1	4	1
TW_{o_1}	TW_{d_1}	TW_{o_2}	TW_{d_2}	TW_{o_3}	TW_{d_3}	TH_{v_1}	Cap_{v_1}	TH_{v_2}	Cap_{v_2}
[5,7]	[8,11]	[5,8]	[11,15]	[7,9]	[12,14]	[1, 30]	2	[1, 30]	2

基于这种情况，我们在求解 RMP 问题时得到了一组分数最优解。该解为 $x(1,1) = x(1,2) = 0.5, x(2,1) = x(2,3) = 0.5, x(3,2) = x(3,3) = 0.5$ ，表示使用三种类型的路径（每两位乘客通过合乘经过每条路径的车辆到达目的地），但是决策变量的值为 0.5，这意味着只有 0.5 辆车通过该路径，不具有物理意义。因此，需要通过分支策略来获得可行解，即对于分数值 $x(1,1) = 0.5$ ，我们可以将 $x(1,1)$ 分为 1 和 0 两个子节点，确保乘客 p_1 由车辆 v_1 服务或不服。在分支后，再次使用列生成，并通过子问题生成一系列新路径。分支 $x(1,1) = 1$ 的可行解为 $x(1,1) = x(1,2) = 1, x(4,3) = 1$ ，而对于另一个分支 $x(1,1) = 0$ ，存在两个可行解， $x(2,1) = x(2,3) = 1, x(5,3) = 1$ 和 $x(3,2) = x(3,3) = 1, x(6,1) = 1$ 。最后，我们可得到最优可行解 $x(3,2) = x(3,3) = 1, x(6,1) = 1$ ，这意味着 p_2 和 p_3 由一辆车通过合乘模式， p_1 由另一辆车提供服务。

5. 结论与展望

本文研究在复杂的按需运输网络中通过运用车辆调度功能来提高交通系统的运输能力。更明确的说，VRPLite 软件包为交通系统中的预定问题打下了坚实的基础，旨在优化全网乘客流，并在新环境下为共享无人驾驶车和自动驾驶车辆 (AGV) 的路径规划提供解决方案。本开源程序框架可以帮助研究人员了解复杂的时空状态网络的建模方法，尤其是从时间和状态的角度寻求最短路。由于乘客或货物在城市轨道交通中共享无人驾驶车需要各种设施设备之间的高度协调地配合完成，因此本文提出的算法有助于指导相关部门，

尤其是成熟的物流运输领域[30]采取相应措施减少乘客或货物的交通运输费用和效率[29]。此外如果共享自动驾驶汽车是可充电的，将很大程度地减少其运输过程中的尾气排放[31]。

值得注意的是，与共享自动驾驶汽车不同的是，AGV 通常在具有特殊要求的路径拓扑中，基于视觉轨迹的网络上移动，并且两辆 AGVs 可能会在交叉路口产生冲突[32]。此外，AGVs 的路径通常提前设定以避免不同种类的路障[34]。由此可见，AGVs 的路径规划与铁路系统运行图非常相似[35-37]，其中每一种资源在同一时间只能被一辆车占用。所以，AGVs 的运行计划和路径规划是车辆路径问题的一种变形。文献[32,38]介绍了 AGVs 的求解方法及应用。

此外，VRPLite 软件包采用离散的空间-时间-状态建模方法，建模时考虑基于时间的路径行程时间。文献[39]提出的考虑基于时间的路径行程时间以及路径可行性车辆路径问题，在不同 AGVs 间，空间和时间并不冲突。

我们希望文中所提的理论模型，求解思路以及开源代码有助于自动驾驶车辆的管理与控制领域中的模型构建和优化。下一步研究将着眼于，基于计算机的普遍应用的分布式求解算法的开发，从而解决多行程分配问题、路径规划问题和车辆调度计划问题。

参考文献

1. Cordeau, J. F., Laporte, G., Potvin, J. Y., & Savelsbergh, M. W. P. (2007). *Chapter 7 Transportation on Demand. Handbooks in Operations Research and Management Science*. Elsevier B.V.
2. Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008). A survey on pickup and delivery problems. *Journal Für Betriebswirtschaft*, 58(1), 21-51.
3. Psaraftis, H. N., Wen, M., & Kontovas, C. A. (2016). Dynamic vehicle routing problems: three decades and counting. *Networks*, 67(1), 3-31.
4. Cheng, W. C., & Schonfeld, P. (2015). A Method for Optimizing the Phased Development of Rail Transit Lines. *Urban Rail Transit*, 1(4), 227-237.
5. Lu, K., Han, B., & Zhou, X. (2018). Smart Urban Transit Systems: From Integrated Framework to Interdisciplinary Perspective. *Urban Rail Transit*, 1-19.
6. Bao, X. (2018). Urban Rail Transit Present Situation and Future Development Trends in China: Overall Analysis Based on National Policies and Strategic Plans in 2016–2020. *Urban Rail Transit* 4(1), 1-12. <https://doi.org/10.1007/s40864-018-0078-4>
7. Kelly, J. & Marinov, M. (2017) Innovative Interior Designs for Urban Freight Distribution Using Light Rail Systems. *Urban Rail Transit* 3(4), 238-254.
8. Wang, Y., Zhang, M., Ma, J. & Zhou, X., (2016). Survey on driverless train operation for urban rail transit systems. *Urban Rail Transit*, 2(3-4), 106-113.
9. He, L., Liang, Q., & Fang, S. (2016). Challenges and innovative solutions in urban rail transit network operations and management: China's Guangzhou metro experience. *Urban Rail Transit*, 2(1), 33-45.
10. Niu, H., & Zhou, X. (2013). Optimizing urban rail timetable under time-dependent demand and oversaturated conditions. *Transportation Research Part C: Emerging Technologies*, 36, 212-230.
11. Shang, P., Li, R., Liu, Z., Yang, L., & Wang, Y. (2018). Equity-oriented skip-stopping schedule optimization in an oversaturated urban rail transit network. *Transportation Research Part C: Emerging Technologies*, 89, 321-343.
12. Dampier, A., & Marinov, M. (2015). A study of the feasibility and potential implementation of metro-based freight transportation in Newcastle upon Tyne. *Urban Rail Transit*, 1(3), 164-182.
13. Toth, P., & Vigo, D. (Eds.). (2002). *The vehicle routing problem*. Society for Industrial and Applied Mathematics.
14. Tong, L., Zhou, L., Liu, J. & Zhou, X., (2017). Customized bus service design for jointly optimizing passenger-to-vehicle assignment and vehicle routing. *Transportation Research Part C: Emerging Technologies*, 85, 451-475.

15. Niu, H., Zhou, X., & Tian, X. (2018). Coordinating assignment and routing decisions in transit vehicle schedules: a variable-splitting Lagrangian decomposition approach for solution symmetry breaking. *Transportation Research Part B Methodological*, 107, 70-101.
16. Pallottino, S., & Scutellà, M. G. (1998). *Shortest Path Algorithms In Transportation Models: Classical and Innovative Aspects. Equilibrium and Advanced Transportation Modelling*. Springer US.
17. Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: theory, algorithms, and applications*. Prentice Hall.
18. Ziliaskopoulos, A. K., & Mahmassani, H. S. (1993). A time dependent shortest path algorithm for real time intelligent vehicle/highway systems. *Transportation Research Record Journal of the Transportation Research Board* (1408), 94-100.
19. Chabini, I. (1998). Discrete dynamic shortest path problems in transportation applications: complexity and algorithms with optimal run time. *Transportation Research Record Journal of the Transportation Research Board*, 1645, 170-175.
20. Mahmoudi, M., & Zhou, X. (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. *Transportation Research Part B*, 89, 19-42.
21. Liu, J., Kang, J. E., Zhou, X., & Pendyala, R. (2017). Network-oriented household activity pattern problem for system optimization. *Transportation Research Part C*, DOI 10.1016/j.trc.2017.09.006
22. Zhou, L., Tong, L., Chen, J., Tang, J., & Zhou, X. (2017). Joint optimization of high-speed train timetables and speed profiles: a unified modeling approach using space-time-speed grid networks. *Transportation Research Part B Methodological*, 97, 157-181.
23. Lu, G., Zhou, X., Mahmoudi, M., Shi, T. and Peng, Q., 2018. Optimizing resource recharging location-routing plans: A resource-space-time network modeling framework for railway locomotive refueling applications. *Computers & Industrial Engineering*.
24. Ruan, J.M., Liu, B., Wei, H., Qu, Y., Zhu, N. and Zhou, X., (2016). How many and where to locate parking lots? A space-time accessibility-maximization modeling framework for special event traffic management. *Urban Rail Transit*, 2(2), 59-70.
25. Mahmoudi, M., Chen, J., Shi, T., Zhang, Y., & Zhou, X. (2018). A cumulative service state representation for the pickup and delivery problem with synchronized transfers. Submitted.
26. Chen, R., Zhou, L., Yue, Y., Tang, J., & Lu, C. (2018). The integrated optimization of robust train timetabling and electric multiple unit circulation and maintenance scheduling problem. *Advances in Mechanical Engineering*, 10(3), 1-16.
27. Fisher, M. L., & Jörnsten, K. O. (1997). *Vehicle Routing with Time Windows: Two Optimization Algorithms*. INFORMS.
28. Lübbecke, M. E., & Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, 53(6), 1007-1023.
29. Arslan, A., Agatz, N., Kroon, L., & Zuidwijk, R. (2016). Crowdsourced delivery: A dynamic pickup and delivery problem with ad-hoc drivers. *Technical report, ERIM Report Series Reference*. <http://ssrn.com/abstract2726731>.
30. Savelsbergh, M., & Van Woensel, T. (2016). 50th anniversary invited article—city logistics: Challenges and opportunities. *Transportation Science*, 50(2), 579-590.
31. Muñoz-Villamizar, A., Montoya-Torres, J. R., & Faulin, J. (2017). Impact of the use of electric vehicles in collaborative urban transport networks: A case study. *Transportation Research Part D: Transport and Environment*, 50, 40-54.
32. Qiu, L., Hsu, W. J., Huang, S. Y., & Wang, H. (2002). Scheduling and routing algorithms for AGVs: a survey. *International Journal of Production Research*, 40(3), 745-760.
33. Chen, X., & Li, Y. (2006). Smooth formation navigation of multiple mobile robots for avoiding moving obstacles. *International Journal of Control, Automation, and Systems*, 4(4), 466-479.

34. Ota, J. (2006). Multi-agent robot systems as distributed autonomous systems. *Advanced engineering informatics*, 20(1), 59-70.
35. Yin, J., Tang, T., Yang, L., Gao, Z., & Ran, B. (2016). Energy-efficient metro train rescheduling with uncertain time-variant passenger demands: An approximate dynamic programming approach. *Transportation Research Part B: Methodological*, 91, 178-210.
36. Rao, X., Montigel, M., & Weidmann, U. (2016). A new rail optimisation model by integration of traffic management and train automation. *Transportation Research Part C: Emerging Technologies*, 71, 382-405.
37. Yin, J., Yang, L., Tang, T., Gao, Z., & Ran, B. (2017). Dynamic passenger demand oriented metro train scheduling with energy-efficiency and waiting time minimization: Mixed-integer linear programming approaches. *Transportation Research Part B: Methodological*, 97, 182-213.
38. Fazlollahtabar, H., & Saidi-Mehrabad, M. (2015). Methodologies to optimize automated guided vehicle scheduling and routing problems: a review study. *Journal of Intelligent & Robotic Systems*, 77(3-4), 525-545.
39. Huang, Y., Zhao, L., Van Woensel, T., & Gross, J. P. (2017). Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological*, 95, 169-195.