

专 业 学 位 硕 士 学 位 论 文

基于车联网环境的共享自动驾驶汽车调度研究

**Shared Autonomous Vehicle Scheduling Research
Based on Connected Vehicle Environment**

作 者 姓 名：_____王睿_____

工 程 领 域：_____交通运输工程_____

学 号：_____31706223_____

指 导 教 师：_____王仲_____

完 成 日 期：_____2019 年 5 月 13 日_____

大连理工大学

Dalian University of Technology

大连理工大学学位论文独创性声明

作者郑重声明：所呈交的学位论文，是本人在导师的指导下进行研究工作所取得的成果。尽我所知，除文中已经注明引用内容和致谢的地方外，本论文不包含其他个人或集体已经发表的研究成果，也不包含其他已申请学位或其他用途使用过的成果。与我一同工作的同志对本文所做的贡献均已 在论文中做了明确的说明并表示了谢意。

若有不实之处，本人愿意承担相关法律责任。

学位论文题目：_____

作者签名：_____ 日期：_____年____月____日

摘 要

随着深度学习、5G 网络技术的发展,自动驾驶汽车与车联网逐渐成为人们关注的焦点。使用自动驾驶汽车提供城市出租车服务,能降低出租车服务的人力成本与运力的不稳定性,该模式被称为共享自动驾驶汽车服务。结合 5G 技术的车联网系统能快速整合城市交通信息,为共享自动驾驶汽车的调度提供决策依据。如何合理调度城市共享自动驾驶汽车为乘客提供更好的服务成为当前研究热点。

本文基于车联网环境研究共享自动驾驶汽车的调度问题,设计共享自动驾驶汽车调度架构,重点研究调度架构中的派单调度功能与区域调度功能。将派单调度功能中司乘匹配问题建模为权值二分图的最大权值匹配模型,区域调度中首先使用 K-Means 聚类算法划分交通小区,基于需求的时效性将区域调度问题建模为网络图的最小费用最大流模型,求解出区域间调度方案,并设计调度规则将区域间调度方案转换为实际可用的调度方案。本文最后以纽约市曼哈顿地区 2011 年 3 月 1 日至 7 日的黄色出租车数据为例,假设区域需求预测准确的情况下,对比不同调度模型的接单率和乘客平均等待时间。实验结果表明,当派单调度使用权值二分图的最大权值匹配模型、区域调度使用网络图的最小费用最大流模型且周期时长为 20 分钟时,调度系统运营表现最优。7440 辆共享自动驾驶汽车即可满足曼哈顿区域内全天 98% 的出行需求,7 天内所有小时接单率的下四分位数为 98%,乘客平均等待时间的上四分位数为 1.5 分钟,比传统调度模型更优。

关键词: 共享自动驾驶汽车; 车联网; 权值二分图; 最小费用最大流; 调度算法

Shared Autonomous Vehicle Scheduling Research Based on Connected Vehicle Environment

Abstract

With the development of deep learning and 5G network technology, connected vehicle system and autonomous vehicle have become the focus of attention. The use of autonomous vehicle to provide urban taxi service can reduce the labor cost and capacity instability of taxi service, which is known as SAV service. The connected vehicle system combined with 5G technology can quickly integrate urban traffic information, and provide decision-making basis for SAV scheduling. Properly scheduling of urban SAVs to provide better service for passengers has become a hot research topic.

This paper studies the scheduling problem of SAV based on the Internet of Vehicles environment, and designs a SAV scheduling platform architecture. This paper focuses on the problems of dispatching orders and the matching of zone-based supply and demand in the platform architecture, modeling the dispatch problem as the maximum weighted matching of weighted bipartite graph model, and in the zone-based scheduling, the K-Means clustering algorithm is first used to divide the traffic zone. Based on the timeliness of demand, the zone-based supply and demand matching problem is modeled as the minimum cost maximum flow model to solve zone-based scheduling scheme, and the scheduling rules are designed to convert the zone-based scheduling scheme into the actual available scheduling scheme. Taking the yellow taxi data from March 1st to 7th, 2011 in Manhattan, New York City as an example, assumed that the zone-based demand forecast is accurate, this paper compares the order-served rate and average waiting time of different scheduling models. The results show that using minimum cost maximum flow model to balance zone-based demand and supply, the cycle length is 20 minutes, and the weighted bipartite graph model is used to dispatch order for SAV, the operating scheduling system performs optimally. 7440 SAVs can meet 98% of the daily travel demand in the Manhattan, the lower quartile of the hourly order-served rate in 7 days is 98%, and the upper quartile of hourly average waiting time is 1.5minutes, which is better than the performance of traditional model.

Key Words: Shared autonomous vehicle; Internet of vehicle; Weighted bipartite graph; Minimum cost maximum flow; Scheduling algorithm

目 录

摘 要	I
Abstract	II
1 绪论	1
1.1 研究背景	1
1.2 调度问题概述	2
1.3 国内外研究现状	3
1.4 研究内容与意义	6
1.5 研究技术路线	7
1.6 本章小结	8
2 共享自动驾驶汽车调度架构与地理网络模型	9
2.1 调度架构	9
2.2 地理网络模型	10
2.2.1 起讫点匹配算法	10
2.2.2 路段通行时间标定算法	13
2.2.3 行程时间预测算法	17
2.2.4 交通小区划分算法	18
2.3 本章小结	19
3 共享自动驾驶汽车调度模型	20
3.1 派单调度功能	20
3.1.1 派单调度建模	20
3.1.2 派单调度求解算法	23
3.2 区域调度功能	26
3.2.1 区域调度建模	26
3.2.2 区域调度求解算法	33
3.3 本章小结	36
4 案例分析——以纽约市曼哈顿为例	37
4.1 出租车 GPS 数据	37
4.1.1 数据概述	37
4.1.2 数据预处理与分析	37
4.2 道路网络仿真	40
4.3 交通小区划分	42

4.4 共享自动驾驶汽车调度仿真	46
4.4.1 调度模型设计	46
4.4.2 仿真结果分析	47
4.5 本章小结	50
结 论	51
参 考 文 献	52
附录 A 区域通行时间数据	55
附录 B 各模型对比图	58
致 谢	61
大连理工大学学位论文版权使用授权书	62

1 绪论

1.1 研究背景

近年来随着深度学习与 5G 网络等技术的发展，“自动驾驶汽车”、“人工智能”、“5G 网络”、“车联网”成为人们关注的焦点。2018 年 7 月百度发布可以量产的 Apollo3.0 自动驾驶汽车，Apollo3.0 适应园区各种交通环境，标志自动驾驶技术逐渐成熟。2019 年 5G 网络设施在全国主要城市逐渐铺开，基于 5G 网络高稳定性、低延迟的特点，车联网系统各个设备之间可以快速共享信息资源，使得城市区域内自动驾驶汽车的商用成为可能。自动驾驶汽车的使用会使城市交通服务产生巨大的改变，最明显的改变之一是城市出租车服务。当前城市出租车服务以乘客下单方式的不同主要分为两类：一类是传统的出租车服务，另一类是网约车服务。

传统的出租车服务由司机、乘客、和调度平台组成。当乘客有用车需求时，可以通过电话预定出租车，或者在城市道路附近扬招可用车辆。出租车到达出行起点，乘客上车后，出租车状态从空车转为载客。当完成此次出行后，司机将出租车状态由载客转为空车。司机通过个人经验、部分司机间共享的信息、或者调度平台调度信息等决策寻客路线，寻客路线的选择通常以当前已知信息下个体收益最高为目标。

网约车服务组成部分与传统出租车基本相似，乘客在网络调度平台上输入出行信息如起讫点地理位置和下单时间，网约车平台负责派单调度与部分区域供需匹配调度。相较于传统的出租车服务，网约车的派单调度功能综合考虑多辆车接单的总体效率，在派单过程中更为高效。司机希望个体收益更高，会倾向于挑选行驶里程多、道路状况好等特征的订单。为避免司机的挑单行为，2014 年滴滴推出派单的积分奖励系统——滴米，当司机选择相对较差的订单时会增加滴米值，反之则会扣除滴米值。司机到出行起点的距离和滴米值决定派单的优先顺序。网约车服务中，寻客路线由司机决策，决策信息包含个人经验、部分司机间共享的信息、和网约车调度平台的相关积分奖励信息，寻客路线具有一定的可控性。

目前两种服务方式相互融合，网约车调度平台上架出租车服务，既保留传统出租车服务中扬招乘车的便利性、又提高了出租车接单的总体效率。当自动驾驶汽车投入使用时，当前城市出租车服务将受到剧烈冲击。自动驾驶汽车不需要司机，相较于传统出租车服务和网约车服务而言，使用自动驾驶汽车的人力成本更低，服务效率更高。例如，自动驾驶汽车不需要人来驾驶，因此基本不需要休息、不需要权衡个体当天的收益来决定是否继续工作、不从个体角度而从系统角度决策寻客路线，即从系统整体角度出发去

更容易接到乘客且收益更高的地区等。自动驾驶汽车作为城市出租车服务的交通工具，乘客的出行成本会因人力成本的减少而降低，运营商可以获得更高的利润并且降低现有出租车系统的运营成本。该服务被称为共享自动驾驶汽车服务，是城市出租车服务的一种新形式，通常由一个或者多个运营商提供的自动驾驶车队来满足城市的出租车出行需求。

随着自动驾驶技术的快速升级，共享自动驾驶汽车服务逐渐成为交通领域的热门研究方向。共享自动驾驶汽车调度问题作为共享自动驾驶汽车服务的核心技术问题之一，从提出就受到广泛学者的关注。与传统出租车和网约车的调度问题不同，基于车联网环境下信息快速交换特点的共享自动驾驶汽车调度能以系统最优为目标即不考虑个体车辆收益来提高系统整体调度效率。如何合理调度城市中共享自动驾驶汽车提高调度效率如减少乘客等待时间，提高乘客的服务满意度并增加运营收益等是目前急需解决的问题。

1.2 调度问题概述

调度问题的核心在于提高调度效率如利用有限的信息资源为乘客提供更好的出行体验并增加系统的运营收益等，传统出租车与网约车调度问题的研究对解决共享自动驾驶汽车调度问题有一定的借鉴意义。

司乘角度下，传统出租车与网约车在调度中面临以下问题：

传统出租车服务中信息收集手段不发达，司机无法实时获取大量乘客的出行信息如乘客的上下车地点与时间。乘客大多通过招扬搭乘出租车，这种乘车方法具有随机性，城市部分区域的乘客较难打到车，乘客打车体验差从而放弃使用出租车。同时司机无法准确得知哪些城市区域存在有乘车需求的乘客，只能基于个人经验等决策寻客路线。

随着互联网的发展，乘客通过移动端约车软件下单获得网约车服务。当乘客有出行需求下单时，手机平台收集乘客的出行订单并派单给可用车辆。乘客使用网约车能获得更好的出行体验，因为乘客可以在室内等车且等车时间通常较短。如果要使乘客等待时间合理，网约车平台需要具有高密度、覆盖广的运力和强大的派单系统，快速计算匹配车辆、最佳接客路线。合理的派单距离对司机很重要，司机的订单包含接客距离与行程距离，其中只有行程距离才计费，接单距离相对于行程距离比例过高，会导致司机在接单过程花费时间过多，降低系统整体派单效率。

调度平台或运营商角度下，传统出租车与网约车在调度中面临以下问题：

供需匹配问题：为提高系统调度效率，需要出行需求与可用车辆在时空上匹配，当出行需求与可用车辆在同一区域、同一时间，且密度均较高，则时空匹配成功的概率更

高。传统的出租车与网约车司机基于个人经验等选择寻客概率更高的地区主动寻求时空匹配。当网约车平台规模较小时，供需不易匹配，平台效率低，用户体验差。当网约车平台规模较大时，平台可以通过奖励等方式给予司机补偿，优化司机的寻客路线，提高平台效率，优化乘客体验。城市乘车出行需求一般具有早晚高峰特征，早晚高峰时出行需求远大于车辆供给且出行具有集聚特点，早高峰一般从市郊出发到市中心，晚高峰则相反，此时需要平台具有较高弹性的运力水平。

运力问题：传统出租车司机和网约车司机可以基于个人状态与预期收益决定是否继续工作，因此一天中的运力是不稳定的。尤其当天气恶劣时，司机考虑到安全等问题，下线停止接单，平台的运力大幅度下降，而乘客出行需求会增加，造成了供需不平衡的供需关系。当供需关系失衡时，平台从两个角度入手进行改善：一是提高运力即给予司机奖励或允许拼车行为等；二是提高订单价格，抑制出行需求。

成本问题：网约车平台需要考虑平台、乘客与司机三方的成本与收益。乘客对出行价格较为敏感，期望价格越低越好。司机作为运力供给方，希望价格越高越好。平台作为乘客与司机间的纽带，需要平衡双方利益，抽取部分佣金保证平台发展。

与传统出租车和网约车相比，共享自动驾驶汽车不需要司机驾驶，节省了人力成本，缓解了平台运营中的运力不足问题。用户在移动端软件上下单即可获得出行服务，服务质量更高。运营商无需考虑单个车辆的收益，从系统最优角度解决调度问题，使系统整体效率提高。

1.3 国内外研究现状

共享自动驾驶汽车（Shared Autonomous Vehicle, SAV），又称为自动驾驶出租车，是一种乘客能够使用移动电话或者应用程序呼叫共享自动驾驶汽车提供服务的出行模式，乘客只需要输入出行目的地等相关信息^[1]。共享自动驾驶汽车是城市居民从习惯拥有私人车辆到按出行需求付费的转变，也是“出行即服务”的未来交通出行理念的体现。由于共享自动驾驶汽车不需要考虑车辆个体的收益，实现系统最优调度成为可能。系统最优角度下的调度具有使系统整体运营收益增加、停车费用降低、乘客的等待时间减少等优点。

Ford^[2]提出一种共享自动驾驶出租车调度方法，该方法需要乘客步行去固定乘车站点，共享自动驾驶车辆不能自主去寻找下一位乘客，或者调度到更有可能接到乘客的区域。Kornhause 等^[1]在此基础上研究动态共享乘车调度方法对新泽西州内所有出行的影响，在此方法中，存在一个或者多个固定乘车站点，站点里的共享自动驾驶汽车提供离开站点时间信息，所有拥有共同出行目的地的乘客共享此次出行。Du 和 Hall^[3]使用车队

分配模型并分析车辆共享系统的运行。Barth 和 Todd^[4]、Uesugi 等^[5]、Ciari 等^[6]使用仿真模型通过改变车队规模和车辆再分配来分析系统成本和服务质量的敏感性。George 和 Xia^[7]提出排队模型来解决共享车辆的最佳车队规模问题。Chauvet 等^[8]和 Fan 等^[9]构建车辆再分配模型分别解决固定需求与动态需求。Kumar and Bierlaire^[10]与 Li^[11]分别使用离散整数规划模型与连续近似模型解决共享车辆站点选址问题。

Levin 等^[12]和 Rigole 等^[13]提出共享自动驾驶汽车服务框架,该框架由三个部分组成,分别为出行需求生成模块,道路网络仿真模块和共享自动驾驶车队调度模块。

需求生成模块的实现通常通过模型仿真生成出行需求,该出行需求包含出行的起讫点和出行时间等。Chen 等^[14]、Fagnant 等^[15]、Fagnant 和 Kockelman^[16]、Levin 等^[12]和 Martinez 和 Viegas^[17]、Viegas 和 Martinez^[18],还有 Bischoff 和 Maciejewski^[19]、Boesch 等^[20]分别使用奥斯汀、里斯本、柏林、苏黎世等地区的交通需求标定需求生成模块从而获得交通出行的时空需求分布状态。

不同研究中道路网络仿真模块设计不同。Ford^[2]、Zachariah 等^[21]的研究中道路网络仿真模块使用固定乘车站点而非城市道路网络。Fagnant 和 Kockelman^[22]、Spieser 等^[23]等分别使用曼哈顿网格网络和欧几里得平面作为抽象的道路网络。Zhang 等^[24]和 Fagnant 等^[15]、Fagnant 和 Kockelman^[16]、Martinez 和 Viega^[17]分别使用准动态网格、准动态实际道路和与时间相关的道路通行时间网络。

不同研究中共享自动驾驶汽车调度策略不同。Burns 等^[25]为乘客调度距离最近的可用车辆(包含空闲车辆和即将转为空闲状态的车辆)。Zhang 等^[24]在调度策略中只考虑空闲的自动驾驶车辆。Chen 等^[14]、Fagnant 和 Kockelman^[22]使用基于规则的调度策略,该调度策略将服务区域划分为多个子区域,并为未指定的乘客(随机排序)调度给其相应子区域内最近的空闲车辆。如果有乘客出行需求的子区域中没有空闲自动驾驶车辆,则搜索该子区域周围的子区域。Boesch 等^[20]采用相似的基于规则的调度策略。Fagnant 等^[15]在实际道路网络中使用类似的策略,并采用修改的 Dijkstra 算法确定空闲自动驾驶车辆和未分配车辆的乘客之间的最短路。Bischoff 和 Maciejewski^[19]对大规模的自动驾驶服务进行建模,使用较复杂的规则调度策略,该策略将系统状态分为两种相互排斥的两种状态。如果可用的自动驾驶汽车相对于未分配的乘客出行请求存在供过于求的情况,则为乘客调度最近的空闲车辆。如果自动驾驶汽车供应不足,将即将转为空闲状态的车辆预分配到最近的未分配乘客请求,当车辆转为空闲状态后立刻参与调度。

Vazifteh M M 等^[26]以纽约市 2011 年黄色出租车真实 GPS 数据为例(假设上车时间为乘客出行需求产生的时间,上下车时间间隔为出行行程时间)提出实时城市交通需求

中最小车队问题求解方法,该方法将城市区域内所有出租车出行转换为时间轴上的非循环有向网络图,一次出行为有向网络图中的一条边,当前出行结束至下一出行开始的时间间隔为车辆空驶时间,通过设置车辆最长空驶时间研究该有向网络图的最小路径覆盖问题,最小路径数量即为最小车队。该研究表明,在所有需求已知(已知一天中所有出行的起讫点地理位置和时间)且不考虑拼车的情况下,纽约市出租车的运营水平得到显著提高,与现有出租车运营系统相比,车队规模减少约 40%。按实时出行需求运营方式下,即小部分出行需求事先通过预约等方式已知,大部分出行需求需要实时提交至系统,该研究提出自由调度模型与批次调度模型两种实时调度方法。其中自由调度模型的理念是将距离下单乘客最近的可用车辆分配给乘客使用,批次调度模型理念是先收集一分钟的乘客乘车出行需求,再统一分配车辆给乘客。实验结果证明在单一运营商、乘客最大等待时间为 6 分钟、7440 辆出租车即原车队规模 70%左右的情况下使用批次算法即可满足约 90%的出行需求。

目前国内共享自动驾驶汽车的相关研究较少。李换平^[27]提出基于汽车云的城市自动驾驶车辆调度方法,即预约出行、先到先服务(FCFS)为规则建立单个分派中心的调度模型。谢榕等^[28]提出基于人工鱼群算法的出租车智能调度方法,使用优化算法优化出租车资源的全局调度。范子明^[29]提出基于大连出租车 GPS 的调度系统应用研究,分析出租车 GPS 轨迹数据并建立调度规则。郭瑞雪^[30]提出基于 BP 神经网络预测网约车的出行短时需求,结合车辆实时空间分布计算供需匹配度,以供需匹配度为依据调度空闲车辆。

部分学者通过研究出行需求的时空分布特征推荐寻客路线来解决供需匹配的调度问题。国内学者贾婷等^[31]通过挖掘历史出租车 GPS 数据,分析城市居民出行规律,结合区域内实时的出租车分布情况,提出一种宏观的调度策略,即车辆资源富足区域的车辆调往贫乏区域,具体为考虑载客热点区域出租车饱和率的空闲车辆调度方法、响应乘客需求的调度方法以及基于蚁群算法路径规划方法等。王郑委^[32]基于大数据平台使用聚类算法挖掘载客热点,为出租车调度提供了依据。

部分学者通过建立载客路径推荐算法与预测模型推荐寻客路线来解决供需匹配的调度问题。唐诗韵^[33]在对出租车路网客流分布的研究中,提出潜在呼叫热点和载客概率的概念,并建立载客概率的计算模型,为司机推荐最佳的寻客路线,使得沿推荐路线行驶时载客概率尽可能大且期望空驶时间尽可能短。刘丽^[34]提出基于出租车载客态单位时间收入及相邻空载态寻客时间的效益指标模型,分析出租车高效益客源的时空分布特征,结合出租车供需比、最短行驶时间、热点区域的交通状态等推荐寻客路线。陈轶非等^[35]使用马尔科夫决策模型解决出租车序列推荐载客地点问题,与典型的 Top-K 方法对比,证明该模型能提高司机收益。

于扬红雪^[36]使用天气状况、PM2.5、温度等数据构建多元线性回归模型、随机森林回归模型、梯度渐进回归树模型的线性变换组合模型预测区域出租车的需求量，得到良好的预测效果。朱本常^[37]基于传统的机器学习和循环神经网络两个方向设计一种组合预测模型，在网约车供需预测上有较好的效果。

综上所述，大部分现有研究将共享自动驾驶汽车调度问题分为三个模块，即出行需求生成预测模块、道路路网仿真模块、调度模型模块。出行需求生成模块一般以 GPS 等历史数据分析预测得到，道路网络仿真模块主要分为区域的抽象路网和实际路网，调度模型主要分为基于规则的调度和模型期望收益最高的调度。

以上研究存在以下问题：基于历史数据的出行热点挖掘，对于突发性的大量出行需求不敏感，甚至会使大量订单无法得到响应。基于历史数据推荐单个司机的最佳寻客路线，如果每个司机都选择自己收益最高的寻客路线，会使部分司机的期望收益降低，从而造成部分出租车空驶距离较多，系统整体效率变差等车辆资源浪费现象。将距离乘客最近的可用车辆派给乘客，可能会造成可用车辆资源使用不够充分。大多供需匹配调度的相关研究中仅考虑供需比等参数，由于乘客的出行需求具有时效性，实际调度中会产生无效调度等现象。

1.4 研究内容与意义

目前共享自动驾驶汽车实时调度相关研究主要分为两类，一类基于调度规则来调度可用车辆，另一类基于历史数据分析得到载客热点，以此为依据指导车辆调度。目前研究中很少考虑到以下三点：

1. 共享自动驾驶汽车完成订单转为空闲状态后停留在原地，缺少寻客路线决策过程，造成城市部分区域供需不平衡的现象，在乘客出行高峰期不平衡现象更明显；
2. 乘客的出行需求具有时效性，在区域调度中空闲车辆必须在乘客等待时间窗口内到达乘客上车点附近；
3. 调度需要从系统角度考虑，单个车辆的最优派单调度与供需匹配调度会造成系统资源使用不充分。

本文基于以上三点提出一种共享自动驾驶汽车调度架构，假设城市区域乘客出行需求数量预测准确的情况下，使用实际道路网络进行仿真，建立调度问题中的派单调度模型与区域调度模型，最后以纽约市曼哈顿地区为例分析不同调度模型的优劣。本文研究成果对未来共享自动驾驶汽车调度具有一定的指导意义，从系统最优角度进行车辆调度可以提高系统的调度效率，具体表现为增加系统收益，优化乘客乘车体验等。

1.5 研究技术路线

本文的技术路线如图 1.1 所示：

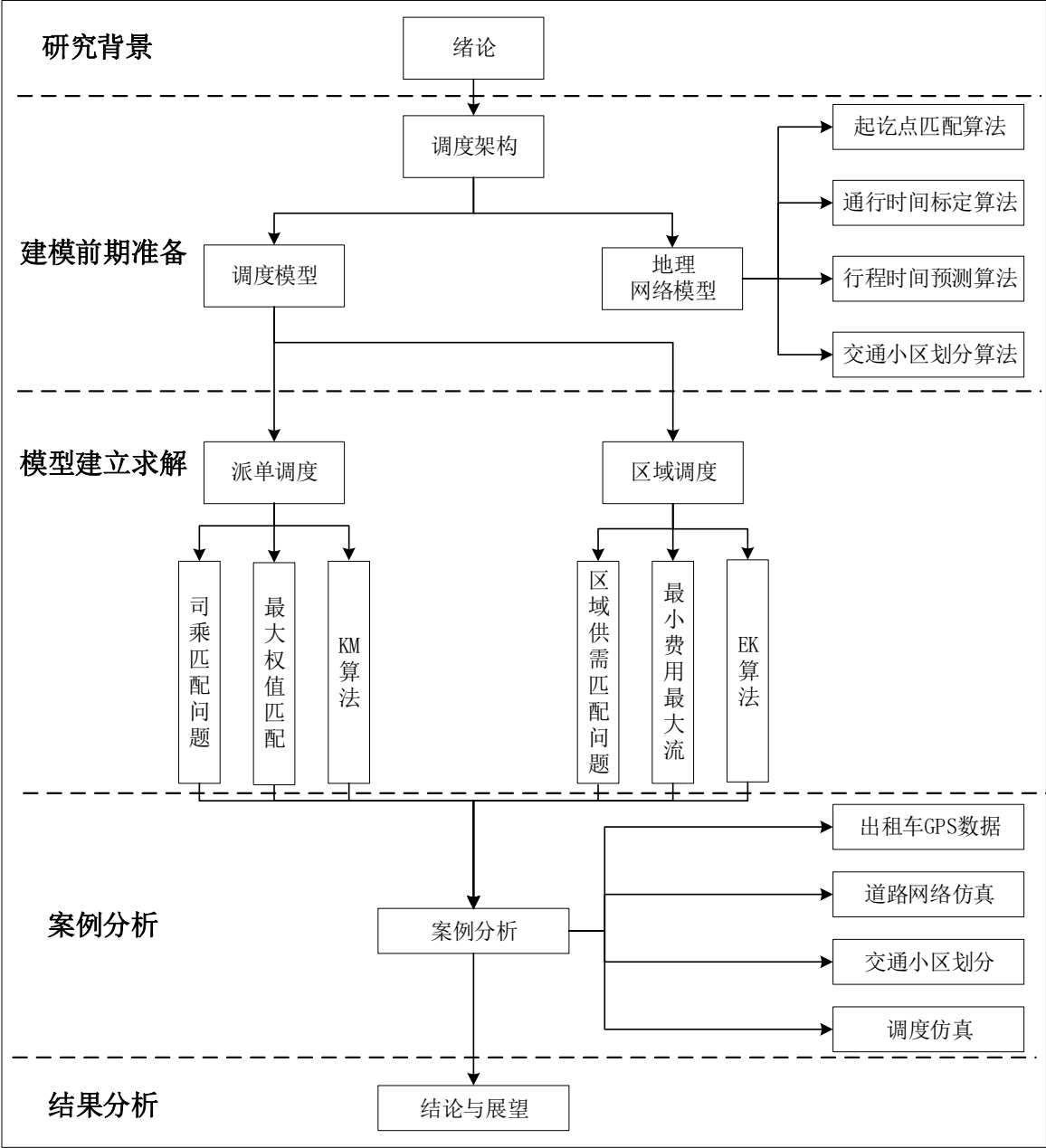


图 1.1 技术路线图
Fig.1.1 Technology roadmap

全文分为五个部分，第一部分为研究背景，总结国内外研究现状，并提出研究内容与意义；第二部分为建模前期准备，提出共享自动驾驶汽车的调度架构，介绍架构内容和地理网络模型，地理网络模型包含、路网相关算法、交通小区划分算法等；第三、四部分是本文的核心内容，第三部分详细介绍共享自动驾驶汽车调派单调度和区域调度的建模过程和求解算法，第四部分以纽约市曼哈顿黄色出租车为例对不同调度模型进行仿真，比较不同调度模型的结果；第五部分结论与展望。

1.6 本章小结

本章首先介绍本文的研究背景，在未来“出行即服务”的理念下，共享自动驾驶汽车对出租车服务产生巨大影响，共享自动驾驶汽车的调度问题急需解决。然后介绍以往调度研究中遇到的问题，论述国内外的相关研究，主流调度框架分为三个部分，分别是出行需求预测模块、道路网络仿真模块、调度模型模块。之后论述本文的研究内容与意义，提出以往研究中常被忽视的三个问题。最后介绍本文的技术路线。

2 共享自动驾驶汽车调度架构与地理网络模型

2.1 调度架构

本文使用的共享自动驾驶汽车调度系统架构如图 2.1 所示。

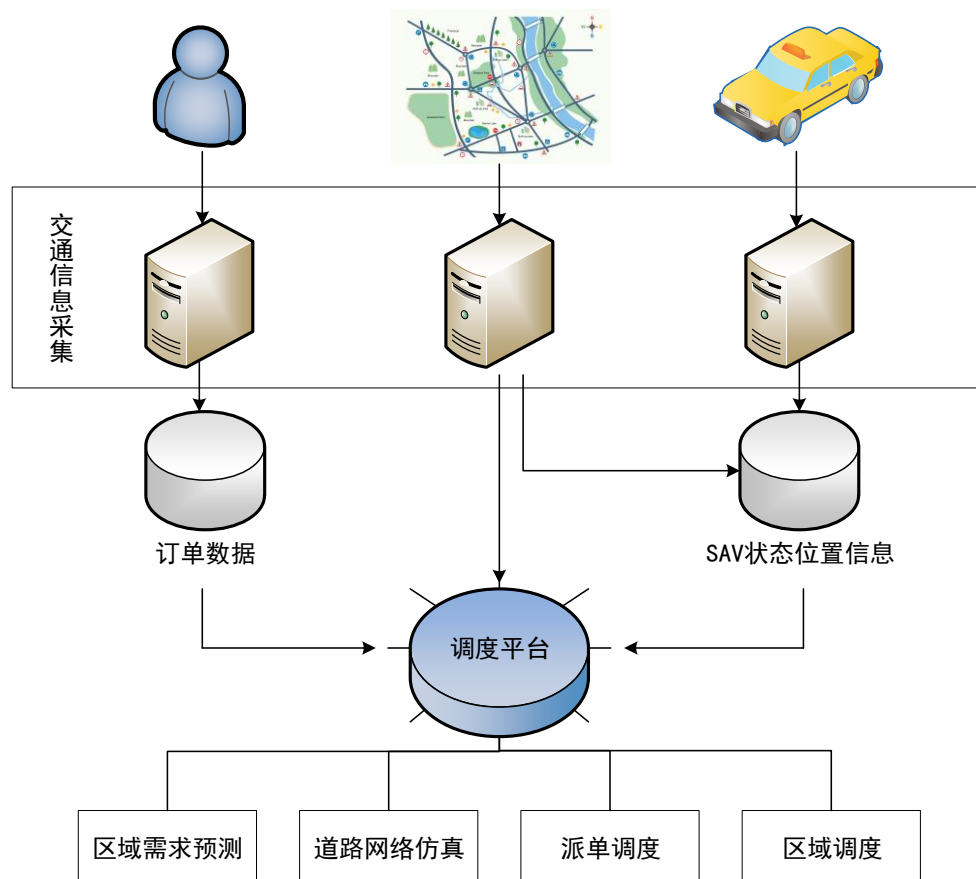


图 2.1 共享自动驾驶汽车调度系统架构图

Fig.2.1 SAV scheduling system architecture

由上图可知，在未来车联网环境下，共享自动驾驶汽车调度架构中数据主要来源于乘客、城市地理空间状态信息、共享自动驾驶汽车。其中乘客通过在移动端软件平台上下单产生订单数据，订单数据包含出行起讫点、下单时间等，城市地理空间状态信息包含城市地理空间信息、实时道路状态、天气、重大事件等，共享自动驾驶汽车提供车辆 GPS 位置信息和载客状态等。调度平台收集以上三种数据，具有区域出行需求预测、道路网络仿真、派单调度、区域调度等功能。区域出行需求预测功能一般将城市划分为若干区域，通过收集区域内交通状态、用地性质、历史订单数量等特征预测未来短时订单

数量。道路仿真功能是结合车辆的实时 GPS 信息和地理道路网络信息计算城市道路状态，提供点到点的路径规划与行程时间预测。派单调度功能是结合乘客位置信息与共享自动驾驶汽车地理位置和状态信息，计算合理的调度方案将车辆分配给乘客使用。区域调度功能是基于区域出行需求预测和共享自动驾驶汽车位置与状态信息来调度当前空闲车辆平衡各个区域的供需，使系统的收益更高。

本文主要研究共享自动驾驶汽车调度架构中的派单调度功能与区域调度功能，关于区域出行需求预测与道路网络仿真以往浮动车相关研究已经有较为成熟的理论方法。本文简化区域出行需求预测与道路网络仿真过程，假设区域出行需求预测和道路通行时间预测准确，即准确知道各个区域的短时需求，能预测城市中任何点到点的通行时间。在此基础上研究不同调度模型下共享自动驾驶汽车的服务质量。

2.2 地理网络模型

将城市道路网络简化成有向图网络，道路网络中交叉口表示为节点（节点集合为 $I = \{I_1, I_2, \dots, I_m\}$ ），道路表示为有向边（边集合为 $S = \{S_1, S_2, \dots, S_n\}$ ）。已知乘客出行信息集合 $T = \{T_1, T_2, \dots, T_k\}$ ，其中出行 $T_i = (o_i, d_i, tt_i)$ ，即出行信息包含起讫点坐标和出行行程时间。在原始数据集中，出行的起讫点 o_i 、 d_i 由 GPS（纬度，经度）坐标构成。

2.2.1 起讫点匹配算法

本文将出行 T_i 起讫点 o_i 、 d_i 的 GPS 坐标匹配到最近的城市道路有向图节点上，记录出行起讫点 o_i 、 d_i 匹配交叉口，并由有向图节点表示。起讫点匹配如下图 2.2 所示。

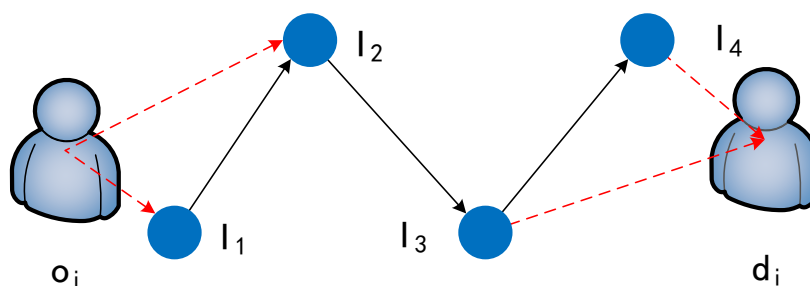


图 2.2 起讫点匹配图

Fig.2.2 Match OD to intersections

在实际的起讫点匹配中，起点和讫点可能在城市街道路段上。本文简化匹配过程，出行起点与讫点匹配算法相同，将起点与讫点均匹配到距离最近的道路网络交叉口上，如上图 2.2 所示，起点 o_i 匹配节点 I_1 ，讫点 d_i 匹配节点 I_4 。

起点 o_i 匹配的节点 I_1 满足以下条件：

$$I_1 \in I, \forall I_j \in I, dis(o_i, I_1) \leq dis(o_i, I_j) \quad (2.1)$$

同理，讫点 d_i 匹配的节点 I_4 满足以下条件：

$$I_4 \in I, \forall I_j \in I, dis(d_i, I_4) \leq dis(d_i, I_j) \quad (2.2)$$

两点间的距离为欧氏距离，定义任意两点 $A(x_1, y_1)$ 与 $B(x_2, y_2)$ 的距离计算公式如下：

$$dis(A, B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.3)$$

由于城市一天的订单数较多，每次使用所有节点遍历算法计算起讫点最近邻节点的较慢，本文采用 KD 树算法加速起讫点匹配的过程。Friedman^[38]提出 KD 树算法，该算法是多维二叉树的扩展——基于优先队列的空间划分树，因为具有快速检索的优点常被用于近邻分析。

参考罗群^[39]的做法，本文使用 KD 树最近邻搜索 BBF (Best Bin First) 算法搜索起讫点的最近节点。假设图中有 N 个节点，使用穷举法搜索起讫点的最近节点需要计算 N 次欧氏距离即平均时间复杂度为 $O(N)$ ，而 BBF 算法平均时间复杂度为 $O(\log N)$ 。城市一天的订单数量较大，使用 BBF 算法能有效减少起讫点最近节点匹配时间。

使用 BBF 搜索算法需要先生成 KD 树，KD 树生成算法伪代码如下：

输入：数据集 $Data-set$ 和其所在的空间 $Range$ ；

输出： $Kd-tree$ ；

1. 如果数据集 $Data-set$ 是空的，则返回空的 $Kd-tree$ ；
2. 调用节点生成程序：
 - 1) 确定 $split$ 域：对于所有描述子数据（特征矢量），统计每个维度上的方差。其中方差最大值对应的维度就是 $split$ 域的值。数据方差大表明数据集沿该维度方向上分散，在这个方向上进行数据分割可以获得更好的效果；
 - 2) 确定 $Node-data$ 域：数据集 $Data-set$ 按其第 $split$ 维度的值排序，位于正中间的数据点被选为 $Node-data$ ， $Data-set' = Data-set \setminus Node-data$ （除去 $Node-data$ 这一点）；
3. 完成以下计算：

$$dataleft = \{d \in Data-set' \mid dis[d, split] \leq Node-data[split]\} \quad (2.4)$$

$$Left_Range = \{Range \ \&\& \ dataleft\} \quad (2.5)$$

使用式 2.4 与 2.5 建立树的左子节点；

$$dataright = \{d \in Data - set' \ \&\& \ d[split] > Node - data[split]\} \quad (2.6)$$

$$Right_Range = \{Range \ \&\& \ dataright\} \quad (2.7)$$

使用式 2.6 与 2.7 建立树的右子节点；

4. 完成以下迭代计算：

$$left = Kd_tree_by(datalleft, Left_Range) \quad (2.8)$$

设置 $left$ 的父节点为 Kd ；

$$right = Kd_tree_by(datarightt, Right_Range) \quad (2.9)$$

设置 $right$ 的父节点为 Kd 。

将数据集合以 KD 树数据结构存储后，需要设计搜索算法快速查找近邻点。本文使用 BBF 最优节点优先算法来搜索数据集中最近邻点，BBF 算法伪代码如下：

输入：已经构造好的 Kd 树与目标点 $Target$ ；

输出： $Target$ 的最近邻节点；

1. 在 Kd 树中找出包含目标点 $Target$ 的叶结点：

从根结点出发，递归地向下搜索 Kd 树。若目标点 $Target$ 当前维度的坐标小于切分点的坐标，则移动到左子结点，否则移动到右子结点，直到子结点为叶结点为止；

2. 以此叶结点为“当前最近点”；

3. 递归的向上回溯，在每个结点进行以下操作：

1) 如果该结点保存的点比当前最近点距离目标点更近，则更新当前最近点，也就是说以该点为当前最近点；

2) 当前最近点一定存在于该结点的一个子结点对应的区域，检查子结点的父结点的另一子结点对应的区域是否有更近的点。具体做法是，检查另一子结点对应的区域是否以目标点为球心，以目标点与当前最近点间的距离为半径的圆或超球体相交：如果相交，可能在另一个子结点对应的区域内存在距目标点更近的点，移动到另一个子结点，接着，继续递归地进行最近邻搜索；如果不相交，向上回溯；

4. 当回退到根结点时，搜索结束，最后的“当前最近点”即为 $Target$ 的最近邻点。

KD 树更适用于数据点数量远大于空间维数时的 k 近邻搜索，当空间维数接近训练实例数时，它的效率会迅速下降到线性扫描的速度。

从上述标准的 KD 树查询过程可以看出其搜索过程中的“回溯”是由“查询路径”决定的，并没有考虑查询路径上一些数据点本身的一些性质。一个简单的改进思路就是将“查询路径”上的结点进行排序，如按各自分割超平面（也称 bin）与查询点的距离排序，也就是说，回溯检查总是从优先级最高（Best Bin）的树结点开始。

2.2.2 路段通行时间标定算法

在计算道路网络图中任意起讫点的最短行程时间前，需要得到道路网络图中路段的通行时间。由于数据集中缺少出租车运行轨迹和速度信息，无法直接使用出租车的 GPS 轨迹来计算路段行程时间，因此本文参考并优化 Santi P^[40]的路段通行时间标定算法，该算法是使用数据集中起讫点及上下车的行程时间估计的启发式算法。

给定一组实际的出行集合 $T = \{T_1, \dots, T_k\}$ ，其中每个出行 $T_i = (o_i, d_i, tt_i)$ 由起点位置 o_i 、讫点位置 d_i 和行程时间 tt_i 确定。在原始数据集中，出行的起讫点被定义为 GPS 坐标，使用起讫点匹配算法将行程起讫点 o_i 、 d_i 更新为道路网络图节点集合 I 中获取的匹配节点（即真实道路网络中的交叉口），即 $\forall T_i \in T, o_i, d_i \in I$ ，路段集合 $S = \{S_1, S_2, \dots, S_n\}$ 定义为道路网络图中连接两个相邻交叉口的所有路段集。

路段通行时间标定算法要解决的问题是估计每条边即路段 $S_i \in S$ 的通行时间 t_{S_i} ，使所有出行 T_i 的行程时间 tt_i 与预测的行程时间 et_i 的相对误差最小化。当确定 S 中每条边误差最小的通行时间时，可以使用最短路算法计算任意两个节点之间的行程时间 $I_i, I_j \in I$ 。除了出行集合 T ，本文已知路段集合 S 中任意 S_i 的路段长度记为 $L(S_i)$ ， p 个路段的最大通行速度集合 $PV = \{pv_1, pv_2, \dots, pv_p\}$ （道路网络数据集中部分道路没有道路等级和最大通行速度）。

本文将出行集合 T 按工作日出行和休息日出行划分为两个部分，每个部分按时间划分为时间切片子集 T_1, \dots, T_{24} ，其中子集 T_i 包含第 i 个小时内所有出行。本文为简化表示，以 T 表示任意时间片子集 T_i ，后文的行程时间预测过程可以在每个时间切片的出行子集上独立计算。在时间片出行集合 T 中，会形成等效出行 T^1, \dots, T^h ，当两出行 T_u, T_v 满足 $(o_u, o_v) \& (d_u, d_v)$ 时具有相同效果。在最短路算法中给定相同的起讫点 I_i, I_j ，从起点到讫点的最佳路线对于任意两次出行均相同。 $T_u, T_v \in T^{i,j}$ ，则 $T^{i,j}$ 是具有起点 I_i 和讫点 I_j 的出行。因此，所有具有相同起讫点的 $T^{i,j}$ 出行都可以看作是同起讫点最短路上行程时间的

多个样本。 $T^{i,j}$ 中的所有出行均由单一出行 $T_{i,j}$ 替换,行程时间 $\bar{tt}_{i,j}$ 等于 $T^{i,j}$ 中所有出行的平均行程时间。对所有等效出行执行此步骤后,我们得到合并的集合 T_{agg} 。

路段通行时间标定启发式算法如下。首先,过滤起讫点相同的出行和行程时间过短或过长的出行。然后使用一个简单的启发式方法补全路段的最大通行速度,使用补全后的最大通行速度 v (不同道路等级不同)计算初始路线的行程时间,过滤速度过快或过慢的行程。由于行程时间预测过程中噪音数据会使估计误差增加,出行时间切片集合 T 预处理的过程中去除了噪音数据。

完成数据预处理后进入迭代过程。迭代过程由两个嵌套循环组成。在外部循环中,基于更新的路段通行时间计算新的出行路线。得到新的出行路线后,计算新的出行行程时间估计值,并计算所有出行的平均相对误差。此外,计算每个路段的偏移值,偏移值是包含该路段所有出行的行程时间估计值与真实值的偏差。然后,开始内部循环,根据计算的偏移值重新确定路段通行时间估计值:初始化增加或减少步长 k ,用于暂时改变路段通行时间估计值(仅当得出的平均速度 v 为 $0.5 \text{ m/s} < v < 30 \text{ m/s}$)。如果新计算得到的平均相对误差相对于当前值减少,则接受初步估计。否则,内部的另一次迭代以较小的 k 开始。重复此过程,直到更新路段通行时间估计值,或 k 达到指定的最小值。当内部循环执行后没有更新的路段通行时间估计时,外部迭代过程终止。

迭代过程结束后,该算法对数据集的路段集合 S_{trip} 中至少一个出行的一个最优路径中包含的每条路段产生了通行时间估计。然后,根据一个简单的启发式方法计算剩余路段的通行时间:具有至少一个相同交叉口的每条路段的通行时间根据相邻路段(即路段 s 和 s' 有相同交叉口)估计的平均速度估计。重复此过程,直到估计完所有路段的通行时间。

路段通行时间标定算法伪代码:

输入: 出行集合 T ; 交叉口集合 I ; 路段集合 S ; 路段长度集合 L ;

输出: 标定的路段通行时间集合 T_s ;

1. 等价出行合并:

$T^{i,j}$ 中所有的出行 T_u ,例如 $(o_u = I_i) \& (d_u = I_j)$;对每个 $T^{i,j}$,用单一出行 $T_{i,j}$ 代替 $T^{i,j}$,计算:

$$tt_{i,j} = \bar{tt}_{i,j} = \frac{\sum_{T_u \in T^{i,j}} tt_u}{|T^{i,j}|} \quad (2.10)$$

让 T_{agg} 成为合并的 $T_{i,j}$ 的出行集合；

2. 第一次过滤：

对于每个 $T_{i,j} \in T_{agg}$ ，如果 $i = j$ ，将 $T_{i,j}$ 从 T_{agg} 中去掉；对于每个 $T_{i,j} \in T_{agg}$ ，如果 $(\bar{tt}_{i,j} < 2min) \text{ or } (\bar{tt}_{i,j} > 1h)$ ，将 $T_{i,j}$ 从 T_{agg} 中去掉；

3. 初始速度计算：

令 εS 为存在最大通行速度的路段集合， $\varepsilon S = S_{PV}$ ， NS 为不存在最大通行速度的路段集合， $NS = S - S_{PV}$ ；令 $N(S)$ 为路段集合 S 中与 εS 共享至少一个交叉口的路段集合；对于每个 $S_i \in NS$ ，计算 $n_{S_i} = |N(S_i) \cap \varepsilon S|$ ；根据 n_{S_i} 的值，由大到小对 NS 进行排序；对于每个 $S_i \in NS$ 的排序序列，计算：

$$pv_{S_i} = \frac{\sum_{S_j \in N(S_i) \cap \varepsilon S} pv_{S_j}}{|N(S_i) \cap \varepsilon S|} \quad (2.11)$$

$$\varepsilon S = \varepsilon S \cup \{S_i\}; NS = NS - \{S_i\} \quad (2.12)$$

重复上述步骤，直到 $NS = \emptyset$ ；

得到全部路段的最大通行速度 PV 记为 $V = \{v_1, v_2, \dots, v_n\}$ ， $V(S_i)$ 为路段 S_i 的初始速度；

4. 初始路线计算：

对于每个 $S_i \in S$ ，设定对应的初始速度 $v_{S_i} = V(S_i)$ ，设置初始路段通行时间 $t_{S_i} = \frac{L(S_i)}{v_{S_i}}$ ；

对于每个 $T_{i,j} \in T_{agg}$ ，运用 *Dijkstra* 算法计算 I_i 到 I_j 的最短路线；对于每个 $T_{i,j} \in T_{agg}$ ，使得 $S^{i,j} = \{S_1^{i,j}, \dots, S_k^{i,j}\}$ 成为 $T_{i,j}$ 的最短路线的路段集合；

5. 第二次过滤：

对于每个 $T_{i,j} \in T_{agg}$ ，计算平均速度 $as_{i,j} = \frac{\sum L(S_h^{i,j})}{\bar{tt}_{i,j}}$ ；对于每个 $T_{i,j} \in T_{agg}$ ，如果

$(as_{i,j} < 0.5m/s) \text{ or } (as_{i,j} > 30m/s)$ ，将 $T_{i,j}$ 从 T_{agg} 中去掉；

6. 迭代步骤：

1) 设定 $again = true$ ；

2) *While again do*：

$again = false$ ；

对于每个 $T_{i,j} \in T_{agg}$, 运用 *Dijkstra* 算法计算 I_i 到 I_j 的最短路线; 对于每个 $T_{i,j} \in T_{agg}$, 使得 $S^{i,j} = \{S_1^{i,j}, \dots, S_k^{i,j}\}$ 成为 $T_{i,j}$ 的最短路线的路段集合; 对于每个 $T_{i,j} \in T_{agg}$, 计算 $et_{i,j} = \sum_{S \in S^{i,j}} t_s$, 其中 $et_{i,j}$ 表示从交叉口 I_i 到交叉口 I_j 的行程时间估计; 使得 $S_{trip} = \bigcup_{T_{i,j} \in T_{agg}} S^{i,j}$;

$$RelErr = \sum_{T_{i,j} \in T_{agg}} \frac{|et_{i,j} - \overline{tt}_{i,j}|}{\overline{tt}_{i,j}} \quad (2.13)$$

对于每个 $S \in S_{trip}$, 使得 $T_S = \{T_{i,j} \in T_{agg} \mid S \in S^{i,j}\}$; 对于每个 $S \in S_{trip}$, 计算:

$$O_S = \sum_{T_{i,j} \in T_S} (et_{i,j} - \overline{tt}_{i,j}) \cdot |T_{i,j}| \quad (2.14)$$

$k=1.2$;

3) *While true do* :

对于每个 $S \in S_{trip}$:

如果 $O_S < 0$, 那么 $t_s = k \cdot t_s$; 否则 $t_s = \frac{t_s}{k}$;

对于每个 $T_{i,j} \in T_{agg}$, 计算 $et'_{i,j} = \sum_{S \in S^{i,j}} t_s$, 计算:

$$NewRelErr = \sum_{T_{i,j} \in T_{agg}} \frac{|et'_{i,j} - \overline{tt}_{i,j}|}{\overline{tt}_{i,j}} \quad (2.15)$$

如果 $NewRelErr < RelErr$:

对于每个 $T_{i,j} \in T_{agg}$, $et_{i,j} = et'_{i,j}$; $RelErr = NewRelErr$; $again = true$; 返回 2);

否则: $k = 1 + (k-1) \cdot 0.75$;

如果 $k < 1.0001$, 那么从 3) 步骤里退出; 否则回到 3);

7. 剩余路段估计通行时间计算:

$\varepsilon S = S_{trip}$; $NS = S - S_{trip}$; 令 $N(S)$ 为共享至少一个交叉口的路段集合 S ; 对于每个 $S_i \in NS$, 计算 $n_{S_i} = |N(S_i) \cap \varepsilon S|$; 根据 n_{S_i} 的值, 由大到小对 NS 进行排序; 对于每个 $S_i \in NS$ 在排序序列, 计算:

$$v_{S_i} = \frac{\sum_{S_j \in N(S_i) \cap \varepsilon S} v_{S_j}}{|N(S_i) \cap \varepsilon S|} \quad (2.16)$$

$$t_{S_i} = \frac{L(S_i)}{v_{S_i}} \quad (2.17)$$

$$\varepsilon S = \varepsilon S \cup \{S_i\}; NS = NS - \{S_i\} \quad (2.18)$$

重复上述步骤，直到 $NS = \emptyset$ ；

2.2.3 行程时间预测算法

道路网络图中任意两个交叉口 I_i, I_j 之间的行程时间是基于估计的行程时间使用 *Dijkstra* 算法进行计算，首先计算 I_i, I_j 之间的最佳路线，然后通过求最佳路线中路段的通行时间和来表示交叉口间行程时间记为 $ET(i, j)$ 。

计算位于交叉口 I_j 的共享自动驾驶汽车去接出行 $T_i \in T, T_i = (o_i, d_i, tt_i)$ 的乘客上车时，该行程时间计算算法如下：

1. 计算接客最短路线：

调度车辆与出行 T_i 的接客交叉口组合为 (I_j, I_{o_i}) ，基于路段标定的通行时间集合 T_s 使用 *Dijkstra* 最短路算法计算最短路线 $I_j \rightarrow I_{o_i}$ ；使得 $S^{I_j, I_{o_i}}$ 成为最优化路线 (I_j, I_{o_i}) 的路段集合；

2. 计算接客的行程时间：

$$ET(I_j, I_{o_i}) = \sum_{S_h \in S^{I_j, I_{o_i}}} \frac{L(S_h)}{v_{S_h}} = \sum_{S_h \in S^{I_j, I_{o_i}}} t_{S_h} \quad (2.19)$$

$$RelET(I_j, o_i) = ET(I_j, I_{o_i}) + dis(I_{o_i}, o_i) / \sum_{S_h \in S^{I_j, I_{o_i}}} L(S_h) \cdot ET(I_j, I_{o_i}) \quad (2.20)$$

返回 $RelET(I_j, o_i)$ ，接客行程时间为共享自动驾驶汽车所在交叉口 I_j 与出行起点匹配交叉口 I_{o_i} 之间的最短行程时间和起点匹配交叉口 I_{o_i} 至起点 o_i 的通行时间之和。

当共享自动驾驶汽车接到乘客后，对于任何一个出行 $T_i \in T, T_i = (o_i, d_i, tt_i)$ ，假设行程时间 tt_i 为起点至讫点的行程时间，当共享自动驾驶汽车完成本次出行订单后，经过讫点 d_i 至讫点匹配交叉口 I_{d_i} 的通行时间后以空载状态停在讫点匹配交叉口 I_{d_i} 。

当共享自动驾驶汽车进行区域调度时, 假设起讫点交叉口为 (I_i, I_j) , 则对应行程时间为 $ET(I_i, I_j)$, 即共享自动驾驶汽车通过时间 $ET(I_i, I_j)$ 后到达讫点交叉口 I_j 。

2.2.4 交通小区划分算法

本文使用 $K-Means$ 聚类算法^[41]划分交通小区。

$K-Means$ 聚类算法的目标是将数据集划分为 K 个簇, 并给出每个数据簇对应簇中心点。 $K-Means$ 聚类算法的步骤如下:

1. 对数据级进行预处理, 如归一化、处理离群点等;
2. 从数据集中随机抽取 K 个数据点作为簇中心, 记为 $\mu_1^{(0)}, \mu_2^{(0)}, \dots, \mu_k^{(0)}$;
3. 设置 $K-Means$ 聚类算法的目标函数为: $J(c, \mu) = \min_{\mu} \min_c \sum_{i=1}^M \|x_i - \mu_{c_i}\|^2$;
4. 令 $t = 0, 1, 2, \dots$ 为迭代的步数, 重复以下过程至 J 收敛;
 - 1) 对于数据集中的每一个样本 x_i , 将其分配到距离该样本最近的簇:

$$c_i^{(t)} \leftarrow \underset{k}{\operatorname{argmin}} \|x_i - \mu_k^{(t)}\|^2 \quad (2.21)$$

- 2) 对于每一个类的簇 k , 重新计算该类簇的中心:

$$\mu_k^{(t+1)} \leftarrow \underset{\mu}{\operatorname{argmin}} \sum_{i: c_i^{(t)} = k} \|x_i - \mu\|^2 \quad (2.22)$$

$K-Means$ 聚类算法在迭代计算时, 如果当前函数 J 没有达到最小值, 首先固定每个簇的中心 $\{\mu_k\}$, 调整每个样本点 x_i 所属的类别让 J 函数减小, 然后固定每个样本点 $\{x_i\}$, 调整簇中心 $\{\mu_k\}$ 使得 J 函数减小。这两个过程交替循环, 使 J 函数单调递减: 当 J 函数减小到最小值时, $\{\mu_k\}$ 与同时收敛 $\{c_i\}$ 。

本文首先将道路网络中交叉口节点的 WGS_1984 经纬度地理坐标系转为以米为单位的 WGS_1984_World_Mercator 坐标系, 以 X 、 Y 轴的最小值作为坐标系的原点。更新 x', y' 值的公式为:

$$\forall x'_i \in X', x'_i = x_i - \min(X) \quad (2.23)$$

$$\forall y'_i \in Y', y'_i = y_i - \min(Y) \quad (2.24)$$

交叉口除了地理坐标特征外, 增加一天中每个小时的平均出行数量特征记为 $T = \{T_1, T_2, \dots, T_{24}\}$, 对 T 进行加权处理得到 T' , 加权处理公式如下:

$$\forall t_{i,j}' \in T', t_{i,j}' = w * \left(\frac{t_{i,j} - \min(T)}{\max(T) - \min(T)} \right) \quad (2.25)$$

其中, $t_{i,j}$ 为 T 中第 i 个交叉口第 j 个小时的平均出行数量, $\max(T)$ 为 T 中所有元素的最大值, $\min(T)$ 为 T 中所有元素的最小值。 w 为该特征的权重系数, 避免出行需求时间分布特征与地理坐标分布特征相差量级过大。当 w 较大时, 产生部分簇在地理空间上较分散, 当 w 较小时, 簇中部分数据点的出行需求时间分布特征差异性较大。本文为了保证同一簇中各个交叉口在地理空间上的连续性, w 选择较小值。

2.3 本章小结

本章首先介绍共享自动驾驶汽车调度架构, 提出调度架构中数据来源有乘客、共享自动驾驶汽车以及城市活动信息等, 该架构具有出行需求预测、道路网络仿真、派单调度、区域调度等功能, 并重点介绍地理网络模型。地理网络模型包含城市道路网络模型和交通小区划分。本章提出使用 KD 树数据结构和 BBF 搜索算法快速匹配出行的起讫点交叉口, 考虑到数据缺少轨迹信息, 本章优化了一种启发式的路段通行时间标定算法, 并建立接客、送客、区域调度的行程时间预测算法, 最后提出使用 K-Means 聚类算法划分交通小区。

3 共享自动驾驶汽车调度模型

共享自动驾驶汽车调度架构包含派单调度功能与区域调度功能，派单调度功能是将乘客的出行需求信息（起讫点地理位置、出发时间等）即订单合理推荐给某个共享自动驾驶汽车；区域调度功能是基于交通需求的短时预测和需求时效性调度空闲车辆满足区域用车需求，减少部分区域用车紧张而部分区域车辆富余的情况。派单调度功能可以减少车辆空驶时间与乘客平均等待时间，区域平衡功能从系统最优的角度出发，避免了传统司机出于个体收益最大（用户最优）寻客造成的车辆整体资源的浪费。

3.1 派单调度功能

3.1.1 派单调度建模

在派单过程中，运营商希望司乘匹配率尽可能高，同时减少乘客等车时间提高服务质量减少运营成本。本文假设乘客最长等车时间为 6 分钟，当等车时间超出 6 分钟后，乘客放弃本次乘车出行，平台流失本次订单。具体派单过程分为两步：第一步为乘客找到合适的共享自动驾驶汽车，第二步将订单分配给平台认为最优的车辆。

第一步为乘客找到合适的车辆。本文预设订单派单范围，当用户叫车时，调度平台依据乘客的上车位置计算距离最近的交叉口，以该交叉口为终点，搜索出所有 6 分钟内能到达终点的交叉口，这些交叉口集合作为搜索区域，搜索每个交叉口中空闲车辆和即将到达的车辆作为车辆备选集。在城市车辆派单调度中，部分场景下调度未来即将到达的车辆比调度空闲车辆更合理，如下图 3.1 所示。

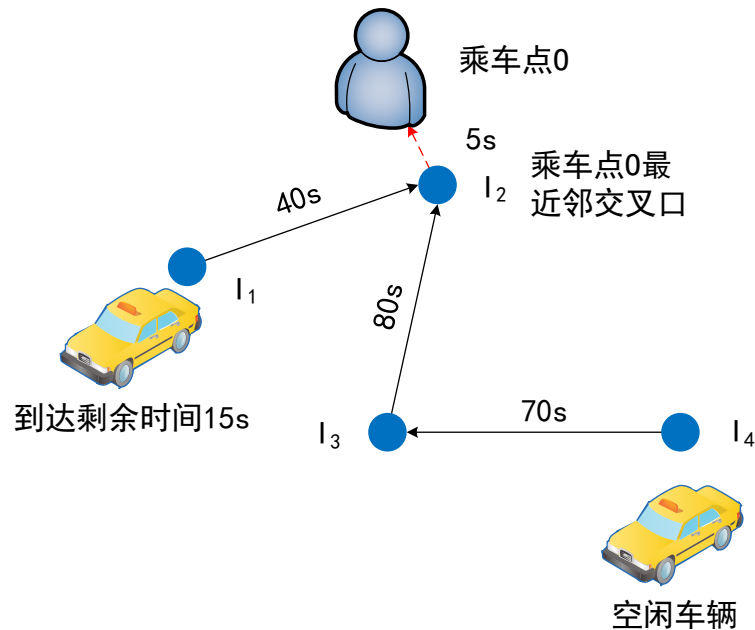


图 3.1 调度即将到达车辆图

Fig.3.1 Dispatch upcoming vehicle to passenger

上图所示 I_1 点的即将到达车辆到达乘车点 O ，乘客等待时间为 60 秒，而 I_4 点的空闲车辆到达乘车点 O ，乘客等待时间为 155 秒。因此在考虑调度车辆备选集时应同时考虑当前空闲车辆和未来即将到达车辆。尤其在城市用车高峰时期，空闲车辆资源紧张下，即将到达车辆在车辆备选集中更加重要。

车辆备选集中空闲车辆到达乘客上车点的行程时间由式 2.20 计算得到，未来即将到达的车辆调度给乘客的接客行程时间等于当前时刻到此车辆转为空车状态的时间与讫点交叉口到乘客上车点的通行时间之和。

计算车辆备选集中每辆车到达乘客最近邻交叉口的行程时间，将行程时间小于 6 分钟的车辆位置信息（停放交叉口或未来到达交叉口）、状态信息（是否是空闲车辆）、估计的乘客等待时间等信息按估计的乘客等待时间排序作为可调度车辆集。

第二步将订单分配给调度平台认为最优的车辆。对于乘客而言，希望平台调度等待时间最短的车辆，车辆越快到达上车点，乘客满意度越高。对于单个车辆而言，乘客等待时间越短，则空驶时间越短，可以节省单个车辆的接客成本，提高运行效率。但是对于调度平台而言，简单地将乘客等待时间最短的可用车辆调度给乘客不一定是最合理的，在现实场景中会遇到如下图所示问题。

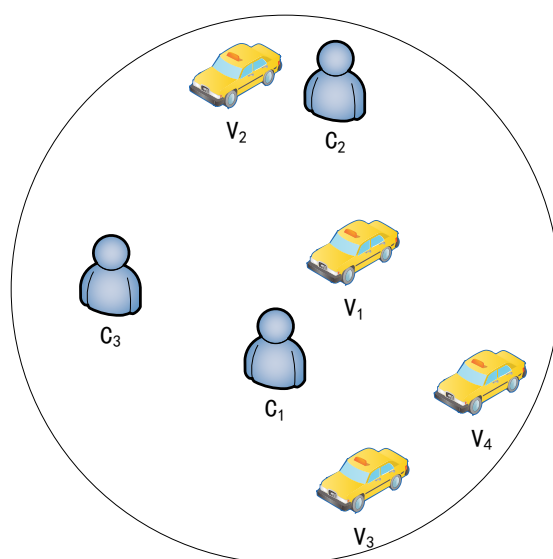


图 3.2 平台订单分配场景图

Fig.3.2 Order allocation scene

表 3.1 平台订单分配时间成本表

Tab.3.1 Time cost of order allocation

乘客&车辆	V ₁	V ₂	V ₃	V ₄
C ₁	2	10	6	8
C ₂	8	1	22	17
C ₃	10	14	16	20

图 3.2 中三位乘客 C_1, C_2, C_3 在较短的时间间隔内依次下单, V_1, V_2, V_3, V_4 为区域内可用空闲车辆, 表 3.1 为各个车辆服务各位乘客的调度时间成本, 即乘客等待时间。由图表可知, 如果按乘客等待时间最短调度可用车辆, 则 C_1 匹配 V_1 , C_2 匹配 V_2 , C_3 由于 V_1 、 V_2 已经匹配, 只能匹配 V_3 , 调度总时间成本为 19。如果 C_1 匹配 V_3 , C_2 匹配 V_2 , C_3 匹配 V_1 , 则调度总时间成本为 17。因此调度乘客等待时间最短的可用车辆成本从平台角度来说未必最优。调度平台希望找到运营效率最高的司乘匹配关系, 运营效率的量化指标为平均乘客等待时间、接单率。平均乘客等待时间越短, 车辆资源利用效率越高。接单率越高, 调度平台收益越高。

综上所述，司乘匹配问题是在一段较短时间内，可派单区域中多个乘客呼叫多个可用车辆，每一位乘客只能匹配一辆可用车辆，或者每一辆可用车辆只能匹配一位乘客，实现派单效率最大化即接单率尽可能高、平均乘客等待时间尽可能短。

解决司乘匹配效率最大化问题有两种方法：一种是枚举法，遍历所有可能匹配关系，计算出匹配个数最高、平均等待时间最短的匹配关系，该算法复杂度较高。另一种是将司乘匹配问题转为权值二分图求解最大权值匹配问题如下图 3.3 所示。

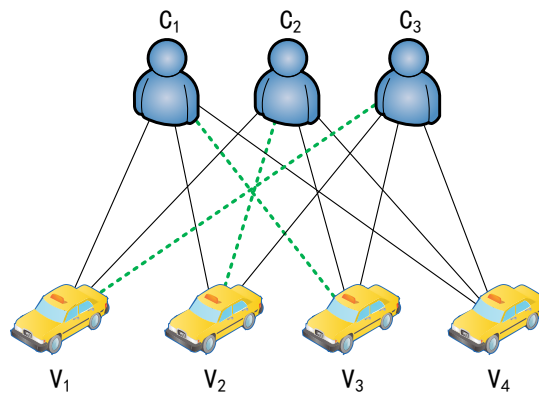


图 3.3 权值二分图模型

Fig.3.3 Weighted bipartite graph matching model

由图可知，司乘匹配问题是在乘客最大等待时间内完成司机与乘客的最佳匹配，每一位乘客都可以有一辆车与之匹配或者每一辆可用车辆都可以分配到一名乘客且系统的总体调度效率最高即系统满足乘客需求数量尽可能多，乘客整体的等待时间尽可能短。

3.1.2 派单调度求解算法

本文将司乘匹配问题转为权值二分图求最大权值匹配问题，二分图的节点分为两个集合，集合 $R = \{R_1, R_2, \dots, R_m\}$ 为所有乘客构成的节点集，集合 $S = \{S_1, S_2, \dots, S_n\}$ 为所有可用车辆集合构成的节点集，边的权值计算公式如下：

$$weight = Max_Waiting_Time - Dispatch_Time \quad (3.1)$$

其中， $Max_Waiting_Time$ 为乘客最大等待时间，值为 6 分钟， $Dispatch_Time$ 为乘客等待时间。

在解释权值二分图之前，先介绍以下概念：

二分图：图论中，图由点和边组成，边的权值有实际业务场景中的意义，如时间或者距离等。把一个图的顶点划分为两个不相交的点集合，图中的任何一条边分别连接这两个集合中的顶点。如果图中存在这样的划分，则称此图为二分图，如下图 3.4 所示。

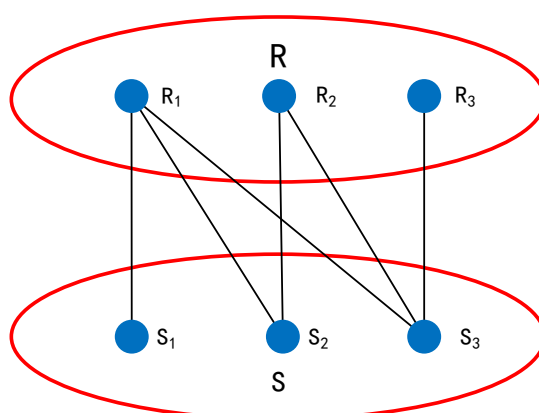


图 3.4 二分图概念示意图

Fig.3.4 Bipartite graph concept diagram

如图 3.4 所示，图 G 由边 $\{(R_1, S_1), (R_1, S_2), (R_1, S_3), (R_2, S_2), (R_2, S_3), (R_3, S_3)\}$ 和节点 $\{R_1, R_2, R_3, S_1, S_2, S_3\}$ 构成，将图 G 的点划分成集合 $R: \{R_1, R_2, R_3\}$ 与集合 $S: \{S_1, S_2, S_3\}$ ，图中任何一边分别连接这两个集合中的顶点，因此图 G 是二分图。

匹配、最大匹配与完美匹配：在图论中，一个匹配是一种边的集合，匹配中任何两条边没有公共顶点，构成匹配的边为匹配边，匹配边的顶点称为匹配点。一个图的所有匹配中，匹配边数量最大的匹配称为图的最大匹配，一个图可能有多种最大匹配。如果一个图中的某个匹配中，所有顶点都是匹配点，则该匹配是一个完美匹配，完美匹配一定是最大匹配，最大匹配不一定是完美匹配。

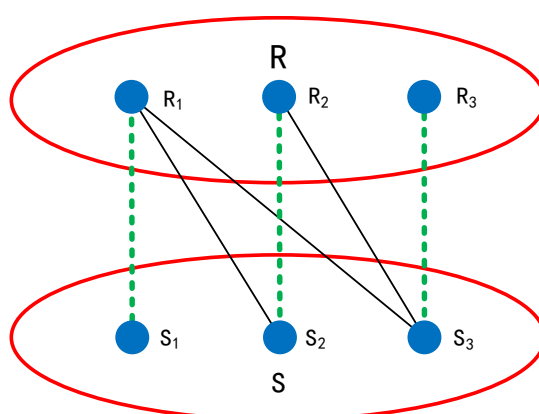


图 3.5 二分图最大匹配图

Fig.3.5 Max matching of bipartite graph

图 3.5 所示, 绿色选中边 $\{(R_1, S_1), (R_2, S_2), (R_3, S_3)\}$ 构成图 G 的一种匹配, 该匹配是最大匹配也是完美匹配。图 G 只有一种最大匹配。

可行顶标: 对于原图中任意一个节点, 给定一个函数 $L(node)$ 求出节点的顶标值。用数组 $LR(R)$ 记录集合 R 中的节点顶标值, $LS(S)$ 记录集合 S 中的节点顶标值。且对于原图中任意一条边 $edge(R_i, S_j)$ 都满足:

$$LR(R_i) + LS(S_j) \geq weight(R_i, S_j) \quad (3.2)$$

相等子图: 相等子图是原图的一个生成子图, 生成子图包含原图的所有节点, 但不包含所有的边, 包含的边被称为可行边必须满足:

$$LR(R_i) + LS(S_j) = weight(R_i, S_j) \quad (3.3)$$

交替路: 从一个未匹配的边出发, 一次经过非匹配边、匹配边、非匹配边……行程的交替路被称为交替路。

增广路: 从一个未匹配点出发, 走交替路, 如果途径另一个非出发点的未匹配点, 则成这条交替路为增广路。通过不断找增广路增加匹配中的匹配边与匹配点, 当无法找到新的增广路时, 达到最大匹配。

本文使用 $KM(Kuhn \text{ and } Munkres)$ 算法求解权值二分图最大权值匹配问题, 该算法是匈牙利算法的贪心扩展算法, 首先介绍匈牙利算法。

二分图 $G=(R, S, E)$, 其中 $R=\{R_1, R_2, \dots, R_m\}$, $S=\{S_1, S_2, \dots, S_n\}$, M 为二分图 G 的任意一个匹配, 匈牙利算法的详细步骤如下^[42-46]:

1. 所有顶点为未扫描顶点, 标记 R 中没有与匹配 M 关联的顶点;
2. 如果上一步没有新的标记在 R 中产生, 则停止, 否则进行下一步;
3. 当 R 中存在被标记且未被扫描的顶点, 选择一个被标记但未被扫描的 R 的顶点, 如果不存在被标记但未被扫描的顶点, 进行下一步;
4. 如果在上一步中没有新的标记被标到 S 的顶点上, 则停止。否则进行下一步;
5. 当 S 存在顶点被标记且未被扫描时, 选择 S 的被标记但未被扫描的顶面, 如果不存在返回到步骤 2。

$KM(Kuhn \text{ and } Munkres)$ 算法定理: 如果原图的一个相等子图中包含完美匹配, 那么这个匹配就是权值二分图的最大权值匹配。由于算法中会一直保持顶标的可行性, 所以任意一个匹配的权值之和小于所有节点的顶标之和, 所以相等子图中的完美匹配一定是最优匹配。

$KM(Kuhn \text{ and } Munkres)$ 算法流程:

1. 初始化可行顶标的值（设定 LR ， LS 初始值）
2. 使用匈牙利算法寻找相等子图的完美匹配；
3. 若未找到增广路则修改可行顶标的值；
4. 重复步骤 2 与步骤 3 直到找到相等子图的完美匹配为止。

3.2 区域调度功能

3.2.1 区域调度建模

城市区域的乘客出行需求随时间变化，必然存在区域供需不匹配的情况。部分区域可用共享自动驾驶汽车较多而当前或者较短的未来该区域出行需求较少，则供大于求，造成可用车辆资源的浪费。部分区域可用车辆少而且目前或者较短的未来该区域出行需求较多，部分乘客在最长等待时间内无法得到乘车出行服务，造成订单损失，降低了服务质量，用户满意度下降，运营商的收益减少。为改善城市乘客出行需求与可用车辆时空不匹配的问题，本文将城市区域抽象化建立了一种网络图模型。

首先将城市划分为不同的小区，每个小区的乘客出行数量在时间轴上具有的一定的规律，目前区域出行需求短时预测精度已经有了较大提升。本文假设城市短时出行需求预测准确下建立区域调度模型。

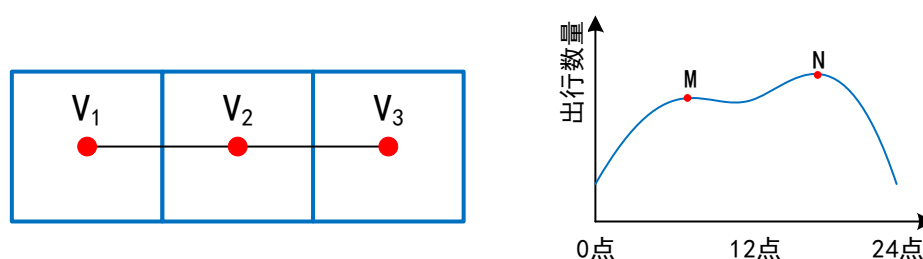


图 3.6 区域调度概念图

Fig.3.6 Concept of zone-based scheduling

如上图所示，三个区域 V_1, V_2, V_3 ， V_1 区域出行需求数量如右图所示，存在需求高峰期如 M, N 分别为早晚高峰点。 V_1 连接 V_2 ， V_2 连接 V_3 。本文提出一种联动式调度如下图 3.7 所示：

表 3.2 供需数据表

Tab.3.2 Supply and demand data

小区编号	V_1	V_2	V_3
出行需求数量	2	2	5
区域空闲车辆数量	4	2	1
区域即将达到车辆数量	1	1	1
区域供给车辆数量	5	3	2
区域富余数量	3	1	-3

表 3.3 区间调度时间表

Tab.3.3 Dispatch time between zones

O&D	V_1	V_2	V_3
V_1	0	1	2
V_2	1	0	1
V_3	2	1	0

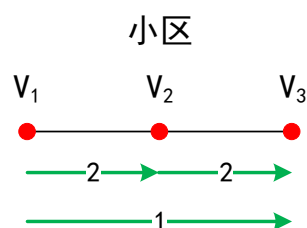


图 3.7 联动式调度策略图

Fig.3.7 Linkage scheduling strategy

小区出行需求数量为预测的当前周期小区乘客出行总数量，区域空闲车辆数量为此刻小区内空闲的车辆数量，区域即将到达车辆数量为当前周期到达该区域的车辆数量，区域供给数量是小区内空闲车辆与即将到达车辆之和，区域富余数量为区域供给数量与出行需求数量之差。小区 V_1, V_2, V_3 的出行需求数量为 2、2、5，区域空闲车辆数分别为 4、2、1，区域即将到达车辆数量分别为 1、1、1，区域供给车辆分别为 5、3、2， V_1 有 3 辆车富余， V_2 有 1 辆车富余， V_3 需要额外调度 3 辆车进来，如果从 V_1 区域直接调度到 V_3 区域，区域间的调度时间过长，对于当前和未来周期供需时空匹配来说并非最优。因此，采用联动式调度方法，能尽快调度可用车辆资源。该图显示联动式调度策略为 V_1 调度 2 辆车到 V_2 ，调度 1 辆车到 V_3 ，同时 V_2 调度两辆车到 V_3 。另一种简单策略从 V_1 直接调度 3

辆空车到 V_3 。调度总距离两种策略均为 6，但联动式调度 V_3 小区等待平均时长为 1.3 小于直接策略的 2。

联动式调度的优点是基本没有改变调度总成本，却使缺乏可用车辆区域的调度车辆平均等待时间缩短，即缺乏区域能更快的获取可用车辆资源。缺点是存在以下问题：

1. 会增加部分区域的乘客平均等待时间：

牺牲部分区域的最优策略使总体区域更优。例如上文举例的 V_2 区域，在时长为 1 的时间中，该区域的可用车辆较少，这一时段内 V_2 区域呼叫乘车服务的乘客的平均等待时间会有所延长， V_3 区域更快的获取可用车辆资源使接单率上升，乘客平均等待时间缩短；

2. 存在区域运力瓶颈：

在城市用车高峰期时，部分区域几乎没有空闲车辆，导致城市部分区域流出流量较小的情况，对联动式调度的效率有较大影响；

3. 区域需求具有时效性：

即虽然联动式调度速度快，但调度到目标区域后，区域的需求已经产生变化，尤其是联动式调度中部分跨区调度车辆。

为解决联动式调度存在的问题，本文考虑需求的时效性与区域流出流量限制（此刻区域的空闲车辆）下建立有向网络图 $D=(V,E,E_C,V_C,W)$ ，在区域需求短时预测的基础上调度城市各个区域的可用车辆，使接单率尽可能高的同时减少乘客的平均等待时间。其中区域由节点 V 表示，节点之间的路径由 E 表示， E_C 表示边的容量即起点区域空闲车辆数量， V_C 表示节点的容量即区域空闲车辆数量， W 表示边的权重即小区之间的平均通行时间。

区域的此刻可用空闲车辆有限，因此节点有容量限制，即区域调度去其它区域的车辆不能超过节点容量，网络图满足以下约束：

容量约束：

$$\forall V_i, V_j \in V, V_i \neq V_j \quad 0 \leq flow(V_i, V_j) \leq edge_capacity(V_i, V_j) \quad (3.4)$$

平衡约束：

$$\forall V_i \in V \quad \sum_{V_j}^{V_j \in Outnodes(V_i)} flow(V_i, V_j) \leq vertex_capacity(V_i) \quad (3.5)$$

其中， $flow(V_i, V_j)$ 表示 V_i 到 V_j 的流量， $edge_capacity(V_i, V_j)$ 表示 V_i 到 V_j 的边的容量， $Outnodes(V_i)$ 表示 V_i 流出的邻近节点集合， $vertex_capacity(V_i)$ 表示节点 V_i 最大流出的容量。将节点容量限制转为边的容量限制，即节点 V_i 拆分成两个节点 V_i_In 和 V_i_Out ，设

置有向边 (V_i_In, V_i_Out) 的边容量为节点容量，该区域所包含的可用空闲车辆数量，构造以下图网络：

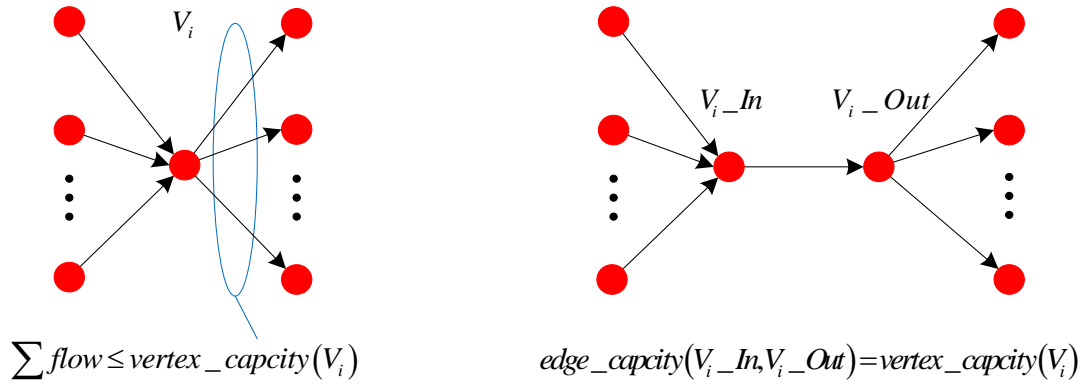


图 3.8 节点容量转换图

Fig.3.8 Conversion of node capacity

如上图所示，重新建立有向网络图 $D=(V,E,C,W)$ ，其中 V 为所有小区各个周期流入点与流出点的集合， E 为考虑时间长度的节点连线， C 为 E 的容量限制即区域间最大调度车辆数量， W 为 E 的权重即区域间的通行时间长度。

以联动式调度数据为例，设置调度周期时长为 1，建立如下区域调度基础网络图。

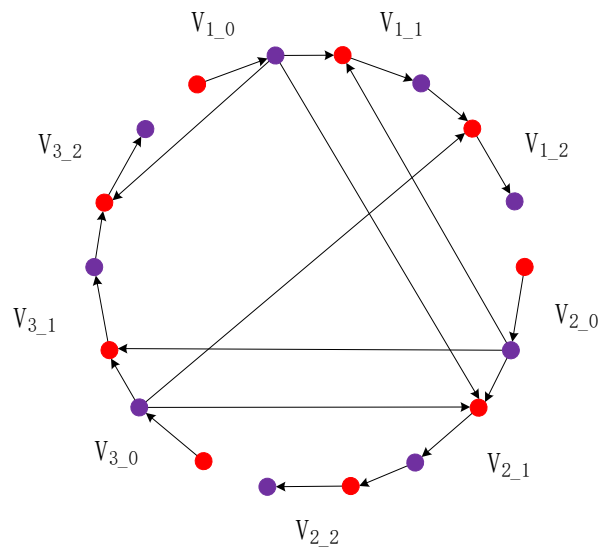


图 3.9 区域调度基础网络图

Fig.3.9 Zone-based scheduling basic network graph

联动式调度例子中，区域的最长行程时间为 2，由于区域间的调度需要时间，因此设置周期时长为 1 将每个区域划分为 0, 1, 2 三个周期，0 代表当前周期，1 代表第二周期，2 代表第三周期，如下图所示。

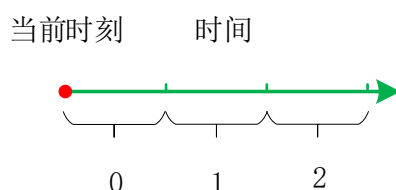


图 3.10 周期示意图

Fig.3.10 Cycle diagram

每个节点每个周期按节点容量转换方式构建流入节点与流出节点，流入流出节点间的边容量为该区域该周期的空闲车辆数量，权值为 0。同一节点不同周期期间，上一周期该节点的流出节点到下一周期该节点的流入节点的边的容量为无穷大，权值为 0。各个节点间的通行时间确定当前周期该节点流出节点到其它节点流入节点的周期（通行时间与周期时长之商向下取整），构建调度的边，容量为调度起始节点的空车数量，权重为区域间的通行时间。每个节点不同周期需求的空车数量不同，需求空车数量为该周期的出行需求与该周期中的到达车辆数量之差，公式如下：

$$IDLE_Vehcile_Demand = \begin{cases} \text{if } (Real_Demand - Future_Arrival_Vehcile) > 0: \\ Real_Demand - Future_Arrival_Vehcile \\ \text{else: } 0 \end{cases} \quad (3.6)$$

在仅知所有区域各周期的出行需求下，为使当前周期空车调度策略最优，修改区域调度基础网络图，转换区域节点的供给与需求建立超级源点 S 与超级汇点 E 。超级源点 S 连接所有区域节点当前周期的流入节点，连接当前周期节点边的容量为该区域当前周期空车数量，所有边的权值为 0。所有节点所有周期的流出节点连接超级汇点 E ，容量为该区的空车需求数量，权值为 0。以联动式调度数据为例，如下图所示。

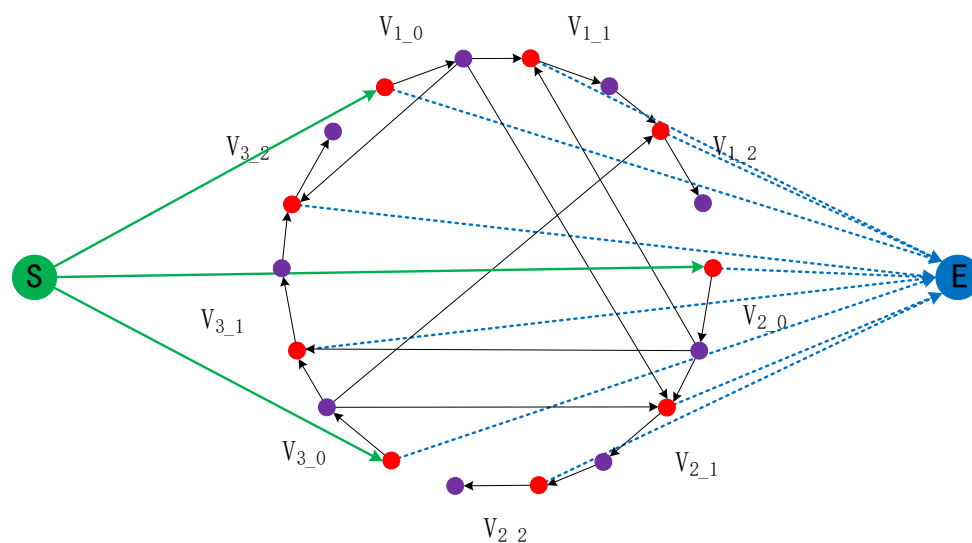


图 3.11 区域调度网络图

Fig.3.11 Zone-based scheduling network graph

上图所示区域调度网络图将区域调度问题转化为求解网络图中的最小费用最大流问题。超级源点 S 连接的每条边容量为区域的空车数量，超级汇点 E 连接的每条边容量为需要的空车数量。该问题描述为如何调度当前周期空闲车辆使得调度及时且成本最小，满足城市整体的出行需求。通过区域调度网络图可知，尽可能满足城市的用车需求即求超级源点 S 到超级汇点 E 的最大流，一个网络图中可能有多种最大流状态，选择成本最小的最大流为城市区域车辆资源调度问题的解。区域调度问题转为网络图的最小费用最大流问题。

求最小费用最大流时计算最大流并选择所有最大流中费用最小的流，在区域调度网络图中会造成调度惰性问题。调度惰性问题是因为在计算当前周期区域调度方案时，选择成本最小收益最高的区域调度方案，但是成本最小收益最高的方案对于系统的调度来说未必最优。例如，当前周期任何一个区域的空闲车辆，在满足本区域本周期需求后，基于调度成本最小的原则会优先选择留在本区域等待下一周期，直至满足本区域各个周期的所有需求后才会选择调度去其它区域。为解决调度惰性问题，本文改进了区域调度网络图如下所示：

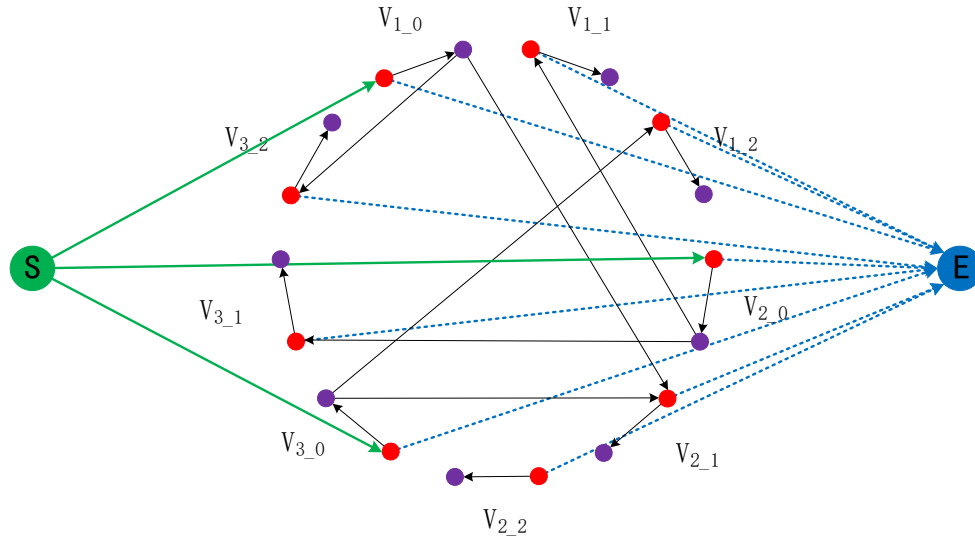


图 3.12 改进的区域调度网络图

Fig.3.12 Improved zone-based scheduling network graph

如上改进的区域调度网络图所示，设置每个区域当前周期空闲车辆满足需求的优先级为：首先满足本区域当前周期的需求，然后满足调度到其它区域相应周期的需求，最后剩余空闲车辆停留在原区域。

通过最小费用最大流求解算法得到区域间的调度方案，区域调度方案包含调出区域编号、调入区域编号、调度车辆数量。区域调度方案转为实际可用的调度方案需要解决从调出区域的哪个交叉口调出多少辆车，去往调入区域的哪个交叉口。由于出行需求具有随机性、区域分布不均匀性、短时持续性，本文设计调度规则流程如下：

1. 统计实时数据：

分别统计调出的区域 V_i 与调入区域 V_j 的所有交叉口记为集合 $N_{V_i} = \{N_1, N_2, \dots, N_m\}$ 、 $N_{V_j} = \{N_1, N_2, \dots, N_n\}$ 。其中调出区域有 m 个交叉口，调入区域有 n 个交叉口，分别统计每个交叉口上个周期的订单数量、当前空车数量；

2. 调出区域交叉口选择方法：

计算调出区域所有交叉口可用车辆数量与上个周期订单数量（如果为 0 则取 1）的商向上保留一位小数乘 10，所得的数作为该交叉口放入调出选择集中的数量，从调出选择集中随机抽取一个交叉口。如果交叉口空车数量为 0，则在集合中删除所有此交叉口元素，重新抽取；否则该交叉口空车数量减 1（可用车辆数量减 1），得到实际可用的调出交叉口与调出车辆数；

3. 调入区域交叉口选择方法：

计算调入区域上一周期所有交叉口的订单数量与此周期可用车辆数量（如果为 0 则取 1）之商向上保留一位小数乘 10，所得的数作为该交叉口放入调入选择集中的数量，从调入选择集中随机抽取一个交叉口，作为实际可用调度方案的调入交叉口，该交叉口可用车辆数量加 1；

4. 循环：

重复步骤 2 与步骤 3，直至微观调度方案数量为对应的宏观调度方案调度车辆数量。

3.2.2 区域调度求解算法

上节将区域调度的供需匹配问题转化为求最小费用最大流问题。

在解释最小费用最大流以前，先介绍以下概念^[47]：

网络与流：给定有向图 $D=(V,A)$ ，在 V 中指定其中一点称为源点（记为 v_s ），另一点为汇点（记为 v_t ），其余的点称为中间点。对于网络图中 $\forall(v_s, v_t) \in A$ ，均有 $c(v_i, v_j) \geq 0$ （简写为 $c_{i,j}$ ）称为弧的容量，限制该弧的最大流。这样的网络记作：

$$D=(V,A,C) \quad (3.7)$$

网络上的流是定义在弧集合 A 上的函数 $f=\{f(v_i, v_j)\}$ ，称 $f(v_i, v_j)$ 是弧 (v_i, v_j) 上的流量（简写为 $f_{i,j}$ ）。

可行流：可行流 f 需要满足以下条件：

1. 容量限制条件：

$$\forall(v_i, v_j) \in A \quad 0 \leq f_{i,j} \leq c_{i,j} \quad (3.8)$$

2. 平衡条件：

对于网络图中间点：

$$\forall(v_i, v_j) \in A \quad \sum_{(v_i, v_j) \in A} f_{i,j} - \sum_{(v_j, v_i) \in A} f_{j,i} = 0 \quad (3.9)$$

对于源点 v_s ：

$$\sum_{(v_s, v_j) \in A} f_{s,j} - \sum_{(v_j, v_s) \in A} f_{j,s} = v(f) \quad (3.10)$$

对于汇点 v_t ：

$$\sum_{(v_t, v_j) \in A} f_{t,j} - \sum_{(v_j, v_t) \in A} f_{j,t} = -v(f) \quad (3.11)$$

上式 3.10 与 3.11 等公式中 $v(f)$ 为可行流的流量，即源点的净流出量。可行流 $f=0$ 称为零流，流量 $v(f)=0$ 。

最大流：最大流是求一个可行流 $\{f_{i,j}\}$ 使其流量 $v(f)=0$ 达到最大，并满足约束条件如下所示：

$$0 \leq f_{i,j} \leq c_{i,j} \quad \forall (v_i, v_j) \in A \quad (3.12)$$

$$\sum f_{i,j} - \sum f_{j,i} = \begin{cases} v(f) & (i=s) \\ 0 & (i \neq s, t) \\ -v(f) & (i=t) \end{cases} \quad (3.13)$$

增广链：网络图中 $f_{i,j} = c_{i,j}$ 的弧称为饱和弧， $f_{i,j} < c_{i,j}$ 的弧为非饱和弧， $f_{i,j} = 0$ 的弧为零流弧， $f_{i,j} > 0$ 的弧称为非零流弧。

若 μ 是网络中连接源点 v_s 与汇点 v_t 的一条链，定义链的方向从 v_s 到 v_t ，链上的弧分为两种，其中一种弧的方向与链的方向一致，称为前向弧，全体记为 μ^+ ，另一种弧方向相反称为后向弧，全体记为 μ^- 。

已知可行流 f ，如果 μ 是从 v_s 到 v_t 的一条链，且 μ 满足以下条件，则 μ 为可行流 f 的增广链。

$$\begin{aligned} (v_i, v_j) \in \mu^+, 0 \leq f_{i,j} < c_{i,j} \\ (v_i, v_j) \in \mu^-, 0 < f_{i,j} \leq c_{i,j} \end{aligned} \quad (3.14)$$

最小费用最大流问题即给定网络 $D=(V, A, C)$ ，网络中任意一个弧 $(v_i, v_j) \in A$ 包含容量参数 $c_{i,j}$ 和单位流的费用 $b(v_i, v_j) \geq 0$ （记为 $b_{i,j}$ ）。则最小费用最大流问题为求网络中一个最大流 f ，使最大流的总运输费用取极小值。其中流的运输费用公式如下：

$$b(f) = \sum_{(v_i, v_j) \in A} b_{i,j} * f_{i,j} \quad (3.15)$$

最小费用最大流问题是运筹学中的经典问题，目前已有较为成熟的算法。

寻找最小费用最大流的方法是从某个可行流出发，找到一条增广链 μ 。沿着 μ 以 $\theta=1$ 调整 f ，得到新的可行流 f' ($v(f')=v(f)+1$)，则增广链 μ 的费用公式如下：

$$\begin{aligned}
 b(f') - b(f) &= \left[\sum_{\mu^+} b_{i,j} (f'_{i,j} - f_{i,j}) - \sum_{\mu^-} b_{i,j} (f'_{i,j} - f_{i,j}) \right] \\
 &= \sum_{\mu^+} b_{i,j} - \sum_{\mu^-} b_{i,j}
 \end{aligned} \tag{3.16}$$

如果 f 是流量为 $v(f)$ 的所有可行流中费用最小的流，而 μ 是 f 所有增广链中费用最小的增广链，沿 μ 调整流 f ，得到新的可行流 f' ，则 f' 是流量为 $v(f')$ 的所有可行流中的费用最小的流。重复上一过程，直到 f' 是最大流时， f' 为该网络图的最小费用最大流。

由于 $b_{i,j} \geq 0$ ，所有当 $f = 0$ 时一定是流量为 0 的最小费用流。从 $f = 0$ 开始，设已知 f 是流量 $v(f)$ 的最小费用流，问题转化为如何寻找关于 f 的最小费用增广链。为此，构造一个赋权有向图 $W(f)$ ，该有向图的顶点是原网络图 D 的顶点，将 D 中的每一条弧 (v_i, v_j) 转变为两个相反方向的弧，分别为 (v_i, v_j) 和 (v_j, v_i) 。定义 $W(f)$ 中弧的权 $w_{i,j}$ 为：

$$w_{i,j} = \begin{cases} b_{i,j} & \text{if } f_{i,j} < c_{i,j} \\ +\infty & \text{if } f_{i,j} = c_{i,j} \end{cases} \tag{3.17}$$

$$w_{j,i} = \begin{cases} -b_{i,j} & \text{if } f_{i,j} > 0 \\ +\infty & \text{if } f_{i,j} = 0 \end{cases} \tag{3.18}$$

其中，长度为 $+\infty$ 的弧在 $W(f)$ 中可以省略。

于是在网络图 D 中寻找流 f 的最小费用增广链转化为在赋权有向图 $W(f)$ 中，寻找从 v_s 到 v_t 的最短路。求解最小费用最大流有 EK 算法^[48-49]，Edmonds 和 Karp^[48]提出改进 Ford-Fulkerson 标记法的最短增广链算法，该算法流程如下：

1. 开始取 $f^{(0)} = 0$ ，在第 $k-1$ 步得到最小费用流 $f^{(k-1)}$ ，则构造新的赋权有向图 $W(f^{(k-1)})$ ，在 $W(f^{(k-1)})$ 中，找从 v_s 到 v_t 的最短路。若不存在最短路（即最短路的权为 $+\infty$ ），则 $f^{(k-1)}$ 就是最小费用最大流；
2. 如果存在最短路，则在原网络 D 中得到相应的增广链 μ ，在增广链上 μ 上对 $f^{(k-1)}$ 进行调整。调整量为：

$$\theta = \min \left[\min_{\mu^+} (c_{i,j} - f_{i,j}^{(k-1)}), \min_{\mu^-} (f_{i,j}^{(k-1)}) \right] \tag{3.19}$$

令：

$$f_{i,j}^{(k)} = \begin{cases} f_{i,j}^{(k-1)} + \theta & (v_i, v_j) \in \mu^+ \\ f_{i,j}^{(k-1)} - \theta & (v_i, v_j) \in \mu^- \\ f_{i,j}^{(k-1)} & (v_i, v_j) \notin \mu \end{cases} \quad (3.20)$$

得到新的可行流 $f^{(k)}$ ，再对 $f^{(k)}$ 重复上述步骤。

3.3 本章小结

本章讲述共享自动驾驶汽车调度模型中的两个主要功能，派单调度功能与区域调度功能。其中派单调度功能是调度可用共享自动驾驶汽车为乘客提供出行服务，采用权值二分图的最大权值匹配算法从系统的角度去优化派单过程。区域调度功能是基于出行需求短时预测，考虑出行需求时效性与系统最优角度等调度区域间空闲车辆，减少乘客等待时间增加共享自动驾驶汽车收入，采用最小费用最大流算法求解区域调度方案，得到区域调度方案后设计调度规则得到实际可操作的调度方案。

4 案例分析——以纽约市曼哈顿为例

4.1 出租车 GPS 数据

4.1.1 数据概述

本文假设未来共享自动驾驶汽车车队由单一运营商运营，替代传统的出租车车队服务城市的出租车出行需求。出租车出行需求数据来源于纽约市 2011 年 3 月黄色出租车出行记录数据，该数据集包含超过 1600 万次的出行记录。数据集中包含如下字段：起点经度、起点纬度、起点上车时间、讫点经度、讫点纬度、讫点下车时间、行程时间、行程距离、乘车人数、出行所需支付金额等。

表 4.1 纽约市黄色出租车出行记录主要字段表

Tab.4.1 Main field of yellow taxi travel record in New York City

编号	字段名称	字段含义
1	Pickup_longitude	起点经度
2	Pickup_latitude	起点纬度
3	Pickup_datetime	起点上车时间
4	Dropoff_longitude	讫点经度
5	Dropoff_latitude	讫点纬度
6	Dropoff_datetime	讫点下车时间
7	Trip_time_in_secs	行程时间
8	Trip_distance	行程距离
9	Passenger_count	乘车人数
10	Total_amount	出行所需支付金额

该数据集不包含出行的相关个人信息，其中起讫点坐标为经纬度格式，上下车时间为日期时间格式，行程时间以秒为单位。本文假设起点上车时间为乘客下单时间，起讫点为乘客出行订单中的起讫点，不存在车辆到达起点需要等待乘客的现象。

4.1.2 数据预处理与分析

美国纽约市 2011 年 3 月无节假日，3 月 1 日为星期二，统计全天乘客乘车出行数量随日期变化如下图 4.1 所示：

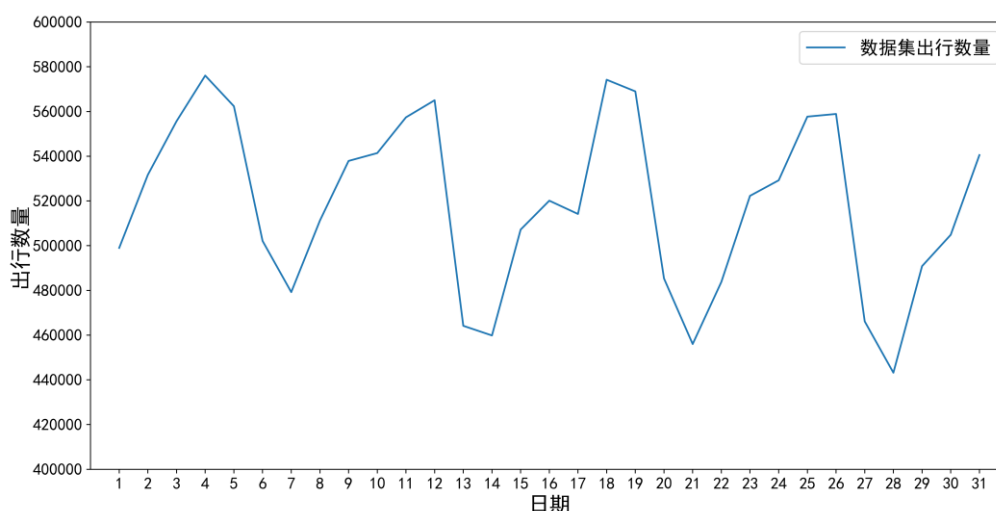


图 4.1 乘客乘车出行数量日期分布图

Fig.4.1 Travel demand quantity distribution with date

由图可知，纽约市全天乘客乘车出行数量具有周期性，周期时长为 1 周。周期内出行数量变化具有以下特征：周一至周六出行数量逐步递增，周日大幅度降低，每周周一出行最少，周五周六出行数量最多。

本文以纽约市曼哈顿区域内的出租车出行记录为调度仿真中的出行需求，构建曼哈顿区域向外扩展 100 米的仿真区域，保留起讫点均在仿真区域内部的出行记录，删除其中部分异常数据。得到仿真区域内出行数量与占全部出行数量的百分比如下图 4.2 所示：

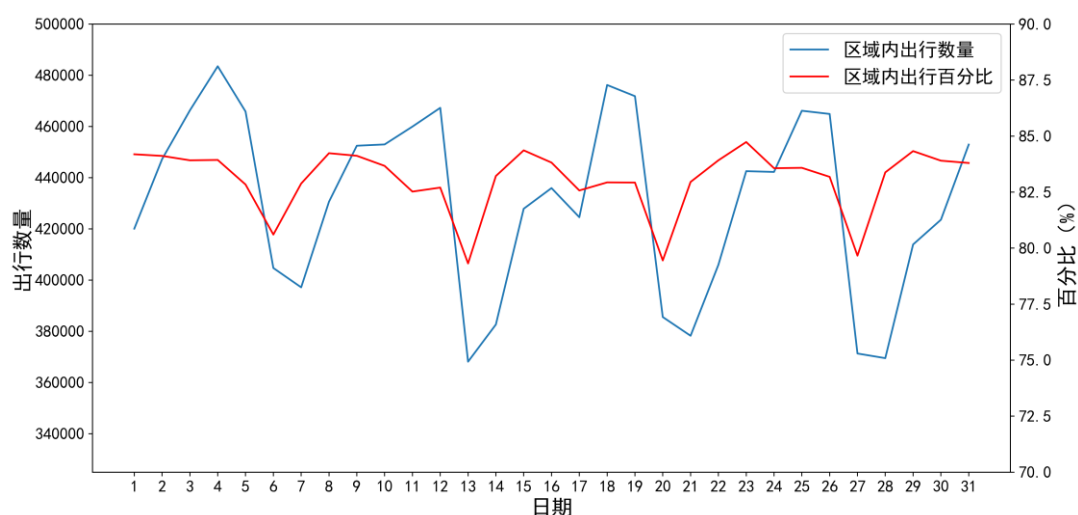


图 4.2 区域内出行数量与占全部出行数量百分比日期分布图

Fig.4.2 Travel demand quantity and the percentage of all demands distribution with date

由图可知，纽约市曼哈顿区域内出租车出行数量约占纽约市全部出行数量的 82%，周一至周六曼哈顿区域内乘车出行所占百分比较高，周日较低。以 2011 年 3 月 1 日至 7 日共 1 周的出行数据为例，统计每小时乘客乘车出行记录的数量（起点上车时间即仿真中的下单时间），得到曼哈顿区域内乘客出行数量随时间的分布如下图所示：

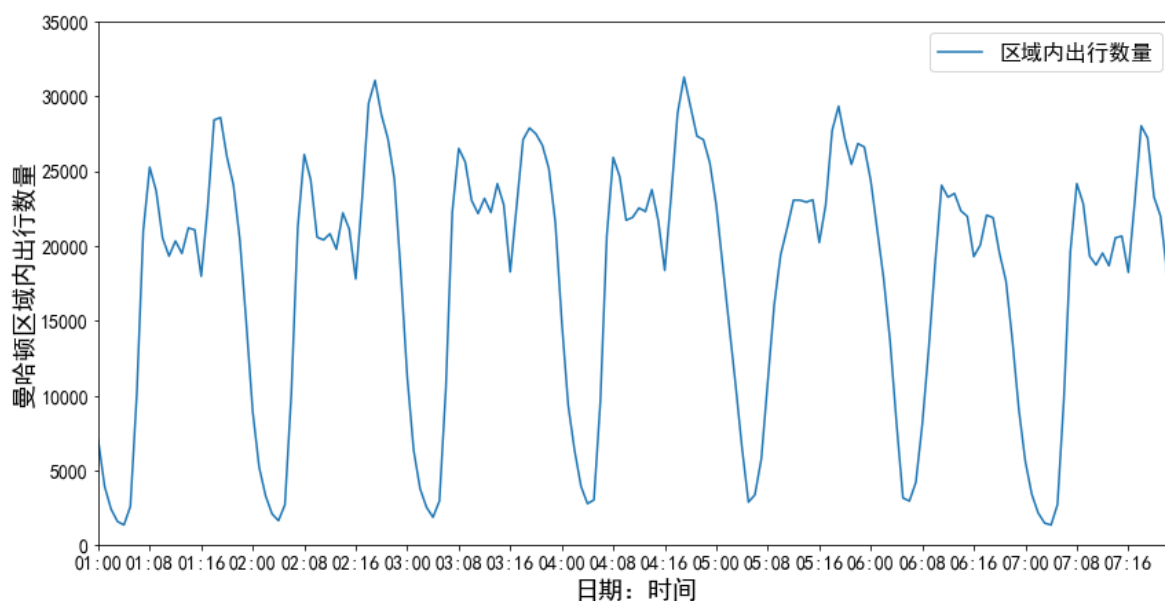


图 4.3 区域内出行数量时间分布图

Fig.4.3 Regional travel demand quantity distributed with time

由图可知，曼哈顿区域出行数量具有早晚高峰特征，早高峰时间为 7:00 至 10:00，晚高峰时间为 17:00 至 22:00，22:00 后需求迅速减少。一天中高峰小时的出行数量最高约为 30000，最低约为 22000，夜晚每小时出行数量较少，基本低于 10000。周一至周六晚高峰出行数量高于早高峰出行数量，周日则相反。

以 2011 年 3 月 1 日全天的出行数据为例，得到纽约市曼哈顿区域的出租车出行起讫点空间分布如下图所示：

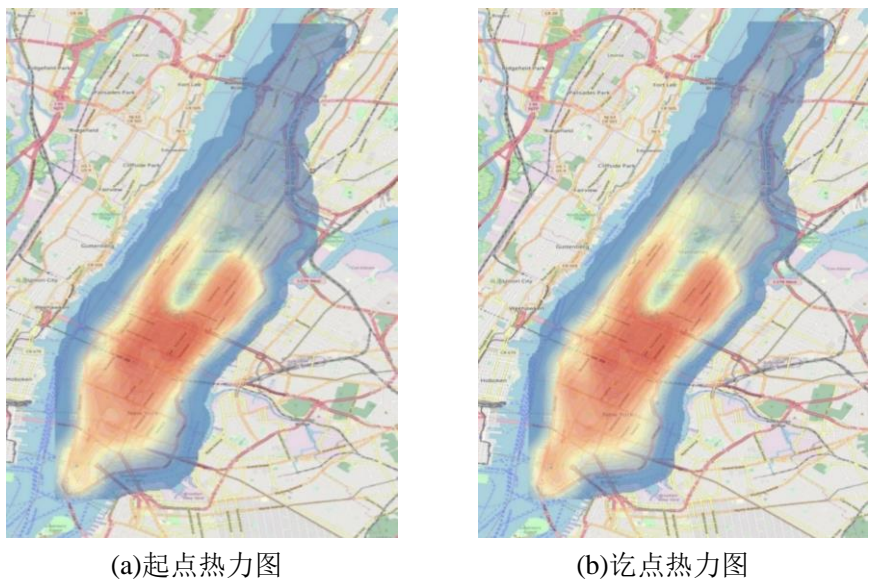


图 4.4 出行起讫点热力图

Fig.4.4 OD heat map of travels

由图可知，曼哈顿区域内出租车出行起讫点集中于区域的中部和南部，中部区域全天出行起讫点的数量最多，是出租车服务的热点。

4.2 道路网络仿真

本文使用 Python 的 OSMNX 库从 Openstreetmap 网站上下载纽约市曼哈顿的道路网络地图，得到的道路网络包含的主要字段如下表所示：

表 4.2 道路网络主要字段表
Tab.4.2 Main field in road network

编号	字段名称	字段含义
1	Osmid	OSM 道路编号
2	Oneway	布尔型，是否是单行道
3	Name	道路名称
4	Highway	道路等级
5	Maxspeed	道路通行最大速度
6	Length	道路长度

其中道路网络中部分道路等级如下表所示：

表 4.3 曼哈顿道路等级表
Tab.4.3 Road hierarchy in Manhattan

编号	等级名称	等级含义
1	Motorway	高速公路
2	Primary	主干道
3	Secondary	次干道
4	Tertiary	三级道路
5	Residential	居住区道路
6	Unclassified	未分类道路
7	Road	不知名道路
8	Living street	街区道路

通过道路网络预处理简化道路网络并删除部分孤立点，得到曼哈顿区域的道路网络图，该网络图包含 4408 个节点，9618 条边。基于不同道路等级的最大通行速度按照道路网络的路段标定算法求得道路网络各个路段的平均通行时间。本文将纽约市曼哈顿地区 2011 年 3 月的出行数据按工作日和休息日的每小时来划分，使用路段标定算法得到包含路段通行时间的网络图。下图以工作日早上 8 点为例显示各个路段的通行速度。

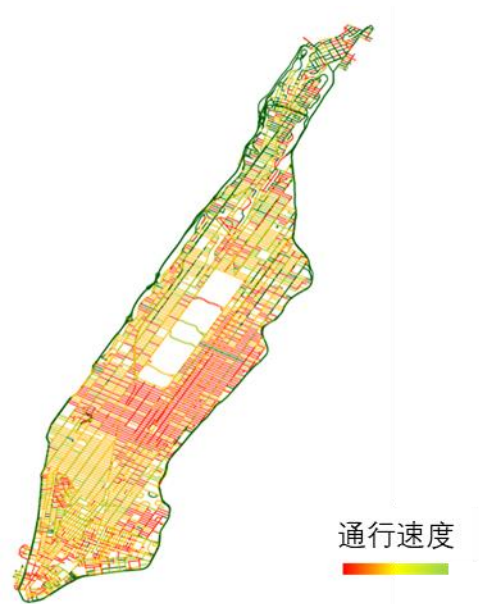


图 4.5 工作日早上 8 点路段通行速度图
Fig.4.5 Speed of the road at 8 am on weekdays

上图所示为工作日早上8点曼哈顿区域内路段的通行速度,红色代表通行速度较慢,绿色代表通行速度较快。曼哈顿中部区域路段的通行速度较慢,边缘区域通行速度较快,与曼哈顿的道路等级分布基本相似。随机从道路网络中选择两个交叉口,使用 Dijkstra 最短路算法计算两交叉口间的行程时间与谷歌地图行程时间对比如下图所示。

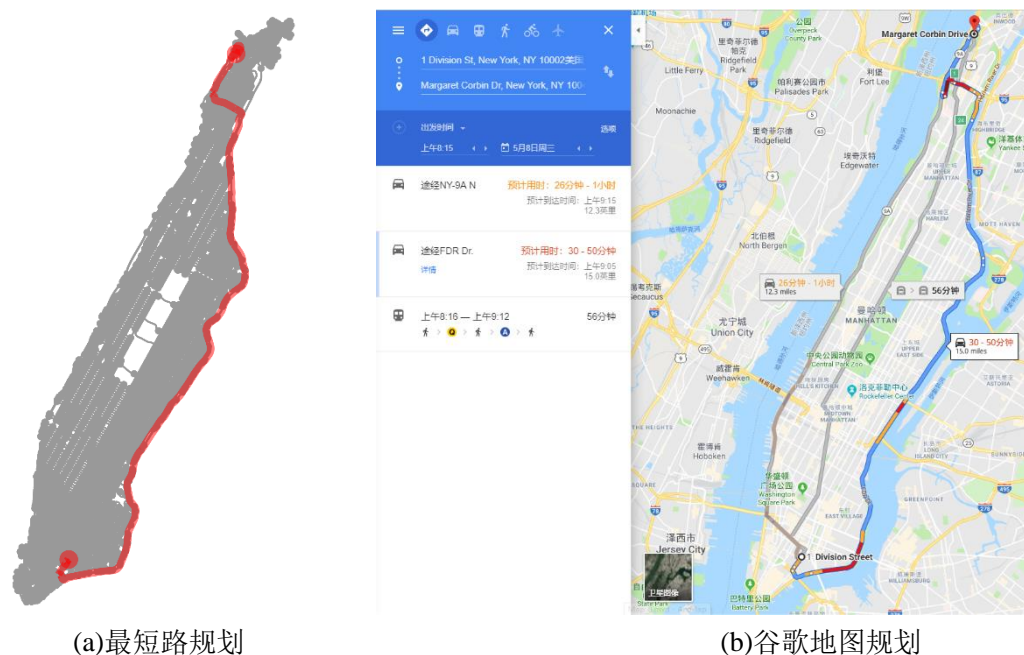


图 4.6 路线规划对比图

Fig.4.6 Comparison of route planning

路径规划对比图中,左侧为已知起讫点使用 Dijkstra 最短路法规划的路线,行程时间为 2519 秒,约 42 分钟。右侧为相同起讫点谷歌地图规划路线,与模型规划路线基本一致,谷歌预测行程时间为 30-50 分钟,预计行程时间为 40 分钟,与模型估计较为接近。因此使用模型仿真真实道路环境具有一定的现实意义。

4.3 交通小区划分

曼哈顿道路网络的地理坐标系为 WGS_1984,将经纬度坐标系投影为以米为单位的地理坐标系 WGS_1984_World_Mercator,按式 2.23 与式 2.24 更新坐标系,下图为道路网络交叉口地理坐标在 X、Y 轴的分布直方图。

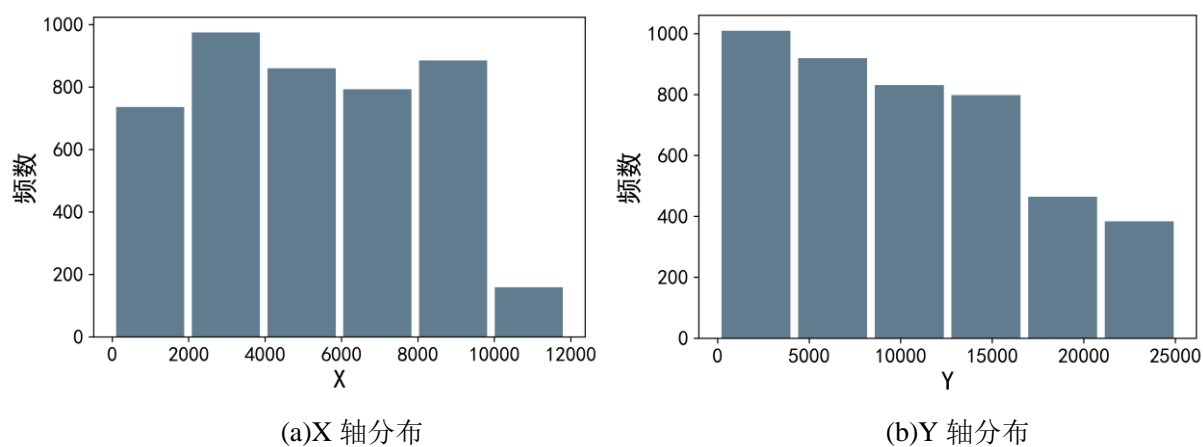


图 4.7 交叉口坐标 X、Y 轴分布直方图

Fig.4.7 Intersection coordinate distributed with X, Y axis histogram

由城市道路交叉口地理坐标在 X、Y 轴的分布可知，曼哈顿区域内交叉口在南北方向上分布更加离散。统计 2011 年 3 月各个交叉口每个小时的平均出行数量，对各个交叉口每个小时的平均出行数量做归一化处理，计算各个交叉口之间的时间序列相关性如下图所示 4.8 所示。

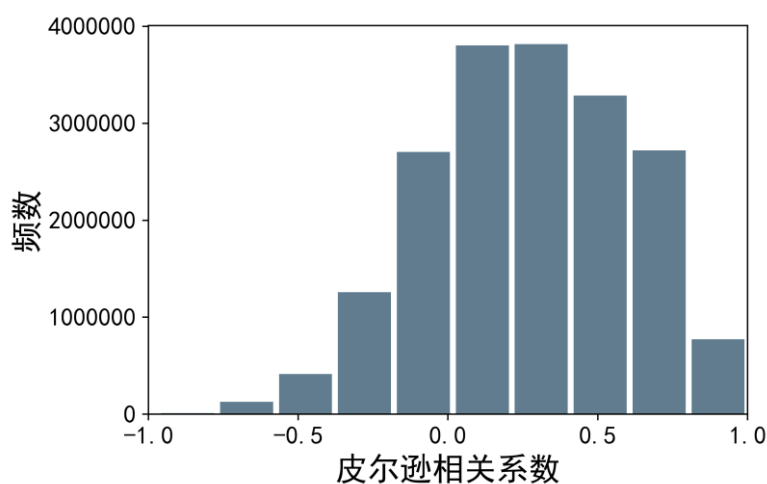


图 4.8 时间序列相关性图

Fig.4.8 Correlation hist of time series

由图可知，大部分交叉口之间时间序列具有正相关性，即大部分交叉口一天中的出行数量在时间上具有相似的分布。设置每个小时出行数量特征的权重为 2500，交叉口 24 小时的出行数量构成 24 维的特征与交叉口地理坐标在 X 轴上分布特征的量级相当，即聚类算法将具有相似时间序列并且地理位置临近的交叉口聚成一类。使用 K-Means 聚类

划分交通小区，K-Means 算法的缺点是需要手动设置 K 值，本文使用手肘法与轮廓系数法确定最佳 K 值。

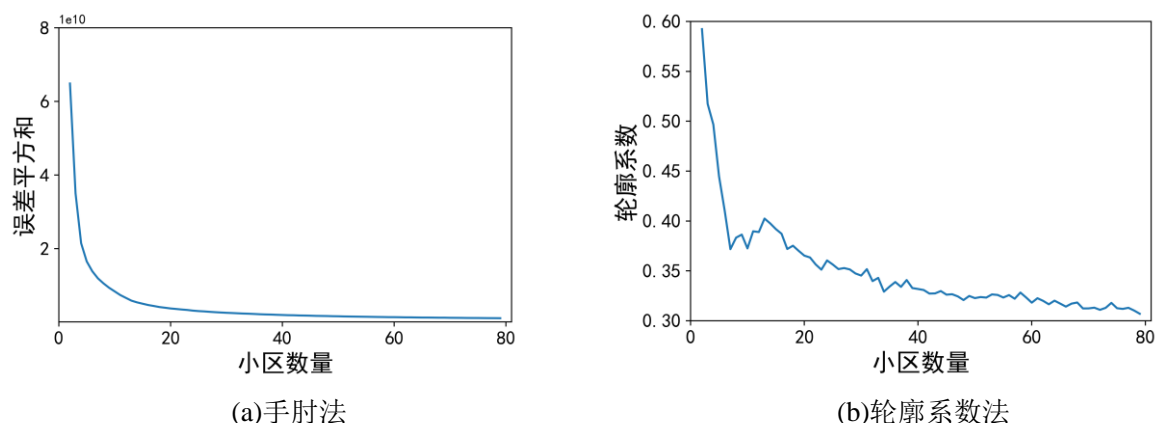


图 4.9 K 值选择图
Fig.4.9 K value analysis

手肘法认为曲线的拐点是 K 值的最佳值，轮廓系数法认为轮廓系数越大越好。综合考虑两种方法，簇的数量最佳取值为 15。考虑到小区大小不能过大，本文选择簇的数量为 30，即曼哈顿区域划分为 30 个小区。设置簇的数量为 30，聚类得到相同标签的交叉口构成交通小区，将交通小区抽象为网络节点，依据小区的空间关系得到交通小区网络图，交通小区网络图如下图所示：

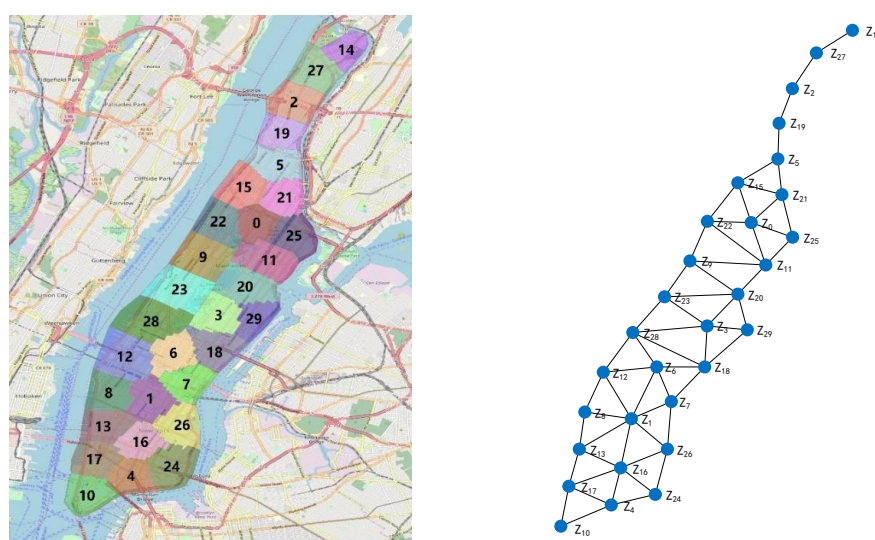


图 4.10 交通小区与区域网络图
Fig.4.10 Traffic zone and zone-based network graph

以工作日早上 8 点为例，计算交通小区内部的各个交叉口间的最短路线通行时间箱型图如下图所示：

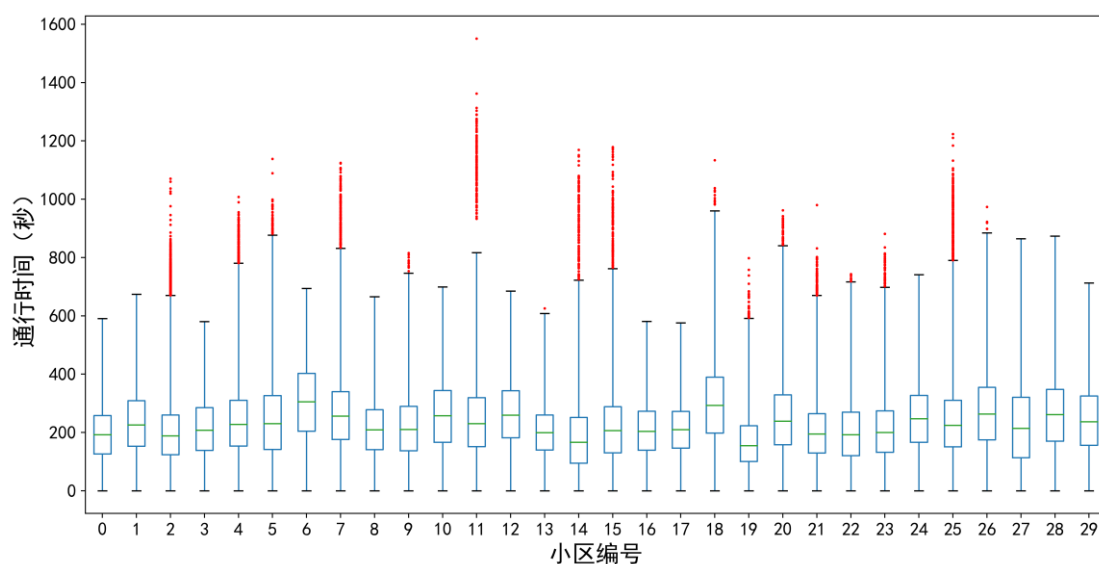


图 4.11 交通小区内各个交叉口间通行时间箱型图

Fig.4.11 Travel time between intersections in traffic zones box diagram

由图可知，工作日早上 8 点区域内任意交叉口间通行时间中位数基本小于 6 分钟，部分区域存在极端值，极端值的产生有以下原因：交通小区内任意一对交叉口间最短行程时间越长则交叉口对的数量越少；区域内部分道路路段过长或通行速度过慢也会造成极端值。交通小区内各个交叉口间通行时间表见附录表 A1。

交通小区间通行时间出行分布图如下：

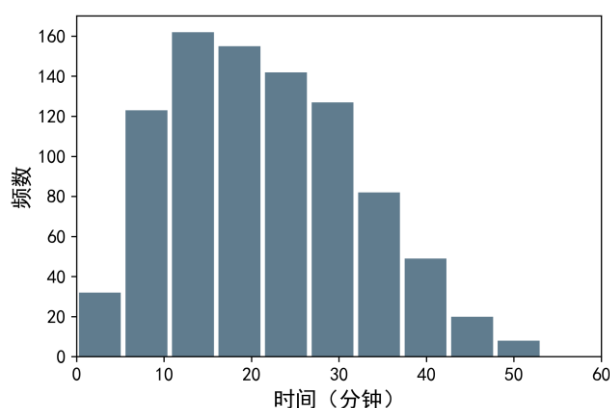


图 4.12 交通小区间通行时间分布直方图

Fig.4.12 Travel time between traffic zones distribution histogram

由图可知交通小区间出行平均行程时间在均 60 分钟以内，绝大多数小于 35 分钟，以此划分短时预测的长度为 60 分钟。交通小区间的交通出行表见附录表 A2。

4.4 共享自动驾驶汽车调度仿真

本文假设乘车出行需求为曼哈顿区域内的出行，不包含境外交通，出行记录中乘客起点上车时间为乘客下单时间。参考 Vazifeh M M 等^[26]的做法，设置曼哈顿区域内共享自动驾驶汽车数量为 7440 辆，乘客最长等待时间为 6 分钟。每天调度平台开始运营前初始化车辆地理位置分布，为避免车辆分布带来的随机性，本文将共享自动驾驶车辆按照交叉口编号由大到小的顺序遍历分布在道路网络交叉口上。研究不同的调度模型对接单率和乘客平均等待时间的影响。

4.4.1 调度模型设计

本文对比以下三种调度模型：

自由调度模型（OTF_Model, On_The_Fly）：

该模型仅考虑派单调度问题，当乘客下单时，系统调度使乘客等待时间最短的可用车辆（包含空闲车辆和即将转为空闲状态的车辆）给乘客；

批次调度模型（Batch_Model）：

该模型仅考虑派单调度问题，参考 Vazifeh M M 等^[26]设置批次间隔为 1 分钟，该调度系统先收集 1 分钟内所有乘客的订单，当下一分钟刚开始时，使用权值二分图最大权值匹配模型调度可用车辆（包含空闲车辆和即将转为空闲状态的车辆）给乘客；

预测调度模型（Predict_Model）：

在批次调度模型基础上添加区域调度功能。由于区域间平均通行时间最长为 60 分钟，设置短时预测时间总长度为 1 小时，将 1 小时划分为 2、3、4、5、6、10 等片段，即预测周期时长为 30 分钟、20 分钟、15 分钟、12 分钟、10 分钟、6 分钟等。以周期时长 20 分钟为例，预测调度模型每 20 分钟进行一次区域调度，使用最小费用最大流算法求解区域调度方案并依照调度规则计算实际的调度方案，派单调度与批次调度模型相同。研究不同周期时长对预测调度模型的影响，同时按照图 3.11 与图 3.12 所示调度优先级的不同分为预测调度模型和改进的预测调度模型。

4.4.2 仿真结果分析

本文仿真以上 14 个不同的调度模型，仿真的订单需求来源于曼哈顿区域内 2011 年 3 月 1 日至 7 日共 7 天 168 小时的出租车出行数据，仿真得到各个模型的 168 个小时仿真数据。其中，预测调度模型、自由调度模型和批次调度模型比较如下所示：

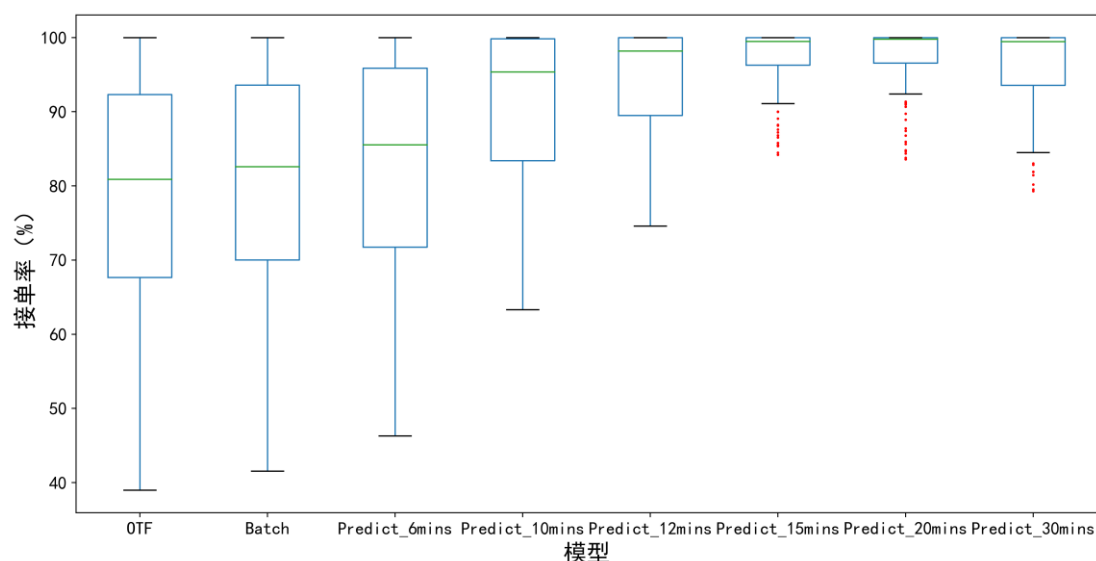


图 4.13 自由调度模型、批次调度模型、预测调度模型接单率箱型图

Fig.4.13 Order-served rate box diagram for OTF_Model, Batch_Model, Predict_Models

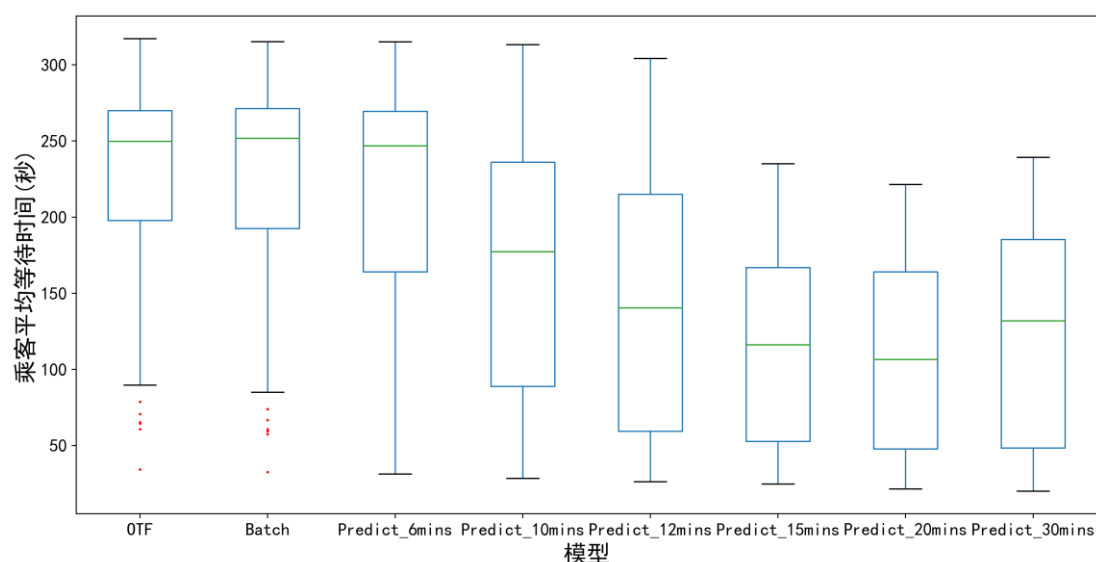


图 4.14 自由调度模型、批次调度模型、预测调度模型乘客平均等待时间箱型图

Fig.4.14 Average waiting time box diagram for OTF_Model, Batch_Model, Predict_Models

由图 4.13 可知，自由调度模型的接单率指标最差，中位数为 81%，上限 100%，下限小于 40%，接单率浮动较大，该模型不够稳定。批次调度模型的接单率中位数为 83%，下限比自由调度模型更高。因此采用批次调度模型能提高系统的接单率优化系统效率、提高系统收入。当周期时长为 20 分钟时，接单率最高且最稳定，接单率下四分位数为 95%。接单率随周期时长的增加先上升后下降。

由图 4.14 可知，自由调度模型与批次调度模型的乘客平均等待时间中位数相同，批次调度模型的乘客平均等待时间相较于自由调度模型更加离散，在乘客等待时间指标上具有一定的优势。其次随着区域调度周期时长的增加，乘客等待时间先下降后增加。当周期时长为 20 分钟，即将 1 小时划分为 3 个时间片段时，每小时乘客平均等待时间最少，7 天内每小时的平均等待时间基本不超过 4 分钟，上四分位数的等待时间约为 3 分钟。

改进后的预测调度模型、自由调度模型和批次调度模型比较如下所示：

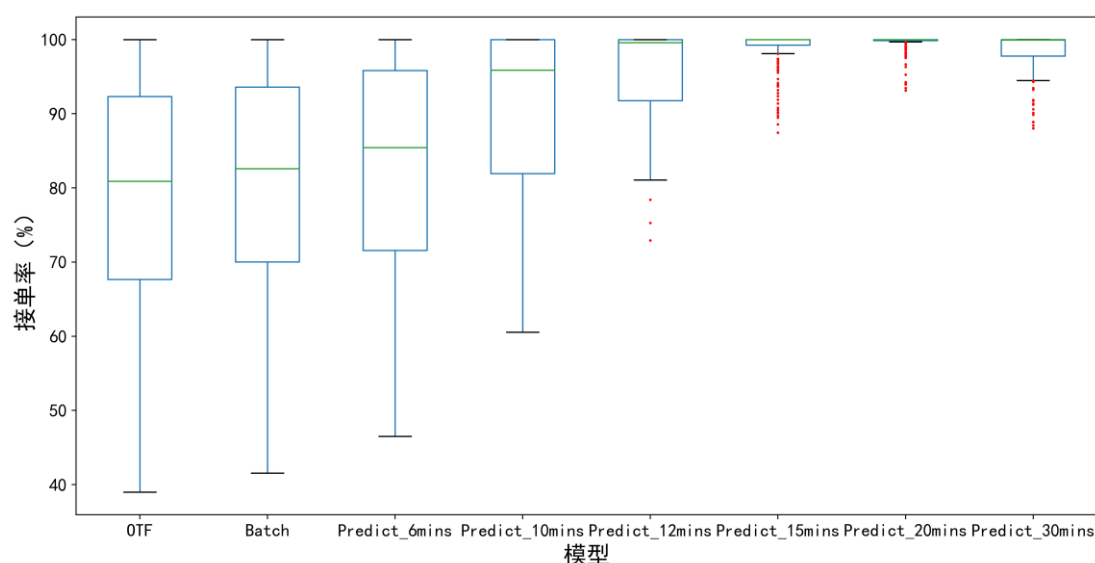


图 4.15 自由调度模型、批次调度模型、改进的预测调度模型接单率箱型图

Fig.4.15 Order-served rate box diagram for OTF_Model, Batch_Model, Improved Predict_Models

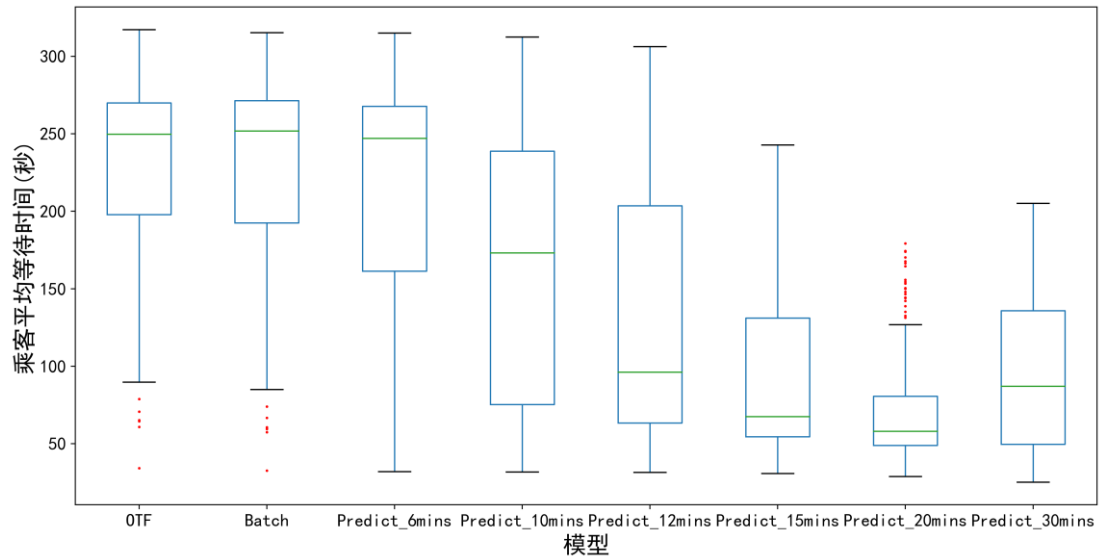


图 4.16 自由调度模型、批次调度模型、改进的预测调度模型乘客平均等待时间箱型图

Fig.4.14 Average waiting time box diagram for OTF_Model, Batch_Model, Improved Predict_Models

由图 4.15 与图 4.16 可知，改进后的预测调度模型相比于改进前的预测调度模型，在接单率与乘客等待时间指标上有了较大的提升。以 20 分钟周期时长为例，改进后的每小时接单率下四分位数达到了 98% 以上，乘客平均等待时间中位数为 1 分钟，上四分位数为 1.5 分钟，7440 辆共享自动驾驶汽车即可满足曼哈顿区域内全天 98% 的出行需求。具体接单率与乘客平均等待时间随时间变化见附录图 B1 至图 B6。

综上所述，批次调度模型相较于自由调度模型，在接单率与乘客平均等待时间上有一定的改进，因为批次调度模型采用权值二分图求最大匹配算法，计算 1 分钟内的最优派单调度，相比单个订单的调度更优。当派单调度使用权值二分图最大权值匹配模型时，同时区域调度周期时长为 20 分钟时，系统的运营效率达到最佳。

直观来说，周期时长越短，区域需求的变化信息掌握的就越充分，调度的接单率与乘客等待时间等指标应该更优，仿真结果与预期相反，分析有以下原因：

第一，需求是不稳定，现实生活中区域的需求曲线并非平滑曲线。周期越短，需求曲线越不平滑。当周期变长时，需求曲线逐渐平滑，调度受到的干扰较小；

第二，受交通小区大小影响。由图 4.11 可知，交通小区的轮廓（小区内交叉口间通行时间的上四分位数）约为 5 分钟，即两个邻近区域形心间的通行时间约为 10 分钟。当周期时长小于 10 分钟，调度车辆刚到达目标小区可能就参与下一轮调度。结合需求曲线不平滑变化较大可知会造成很多无效调度，降低平台调度效率；

第三，当周期时长为 20 分钟以上时，区域调度逐渐失去及时性，虽然可以降低调度成本，但同时也降低了平台的运营收入。

4.5 本章小结

本章以纽约市 2011 年 3 月黄色出租车数据为例，首先预处理出租车 GPS 数据，分析曼哈顿区域内出租车的时空分布特点。结合 Openstreetmap 开源地图资源，使用道路网络路段标定算法计算路段通行时间，标定结果与 Google 地图服务比较通行时间预测相差不大，因此仿真网络具有一定的现实意义。将出租车出行需求匹配到近邻的交叉口上，使用 K-Menas 聚类算法基于交叉口的地理位置和出行需求在时间上的分布计算聚类簇数，将曼哈顿地区划分为 30 个交通小区，研究小区内部交叉口间和区域间的平均通行时间。最后进行共享自动驾驶汽车调度仿真，通过对比不同调度模型的接单率和乘客平均等待时间指标可知派单调度选择权值二分图最大权值匹配模型，设置区域调度周期时长为 20 分钟时得到的结果最好，每小时接单率下四分位数达 98% 以上，7 天 168 小时内每小时平均乘客等待时间基本不超过 3 分钟，每小时乘客等待时间上四分位数约 1.5 分钟。最后分析 20 分钟周期时长效果最佳的原因。

结 论

本文在车联网环境下以共享自动驾驶汽车调度问题为研究对象，首先总结共享自动驾驶汽车、传统出租车和网约车调度研究，发现以下三个较少被重视的问题：第一、共享自动驾驶汽车相关研究缺少寻客路线决策过程；第二、基于短时预测的区域调度研究较少考虑乘客出行时间窗口来设计调度方案；第三、目前研究大多仅考虑单个车辆的最优派单调度与区域调度。

为解决以上问题，本文设计未来共享自动驾驶汽车调度架构，重点解决两个核心问题：派单调度与基于短时预测的区域调度问题。派单调度问题参考传统的派单方法设计权值二分图最大权值匹配模型，使用 KM 算法求解。区域问题，基于历史数据使用 K-Means 算法划分交通小区，假设区域需求预测准确的情况下，构建网络图的最小费用最大流模型，使用 EK 算法求解，并研究不同周期时长对区域调度的影响。

以纽约市曼哈顿为例，本文使用 Python 的 OSMNX 库构建道路网络，提出起讫点匹配算法，改进道路通行时间标定算法，标定得到的路段通行时间与真实路网相近，因此具有一定的现实意义，权值二分图最大权值匹配模型使曼哈顿区域每小时接单率中位数稳定提高 2%，对每小时乘客平均等待时间有一定的优化。将曼哈顿划分成 30 个小区，实验证明在区域出行数量预测准确的情况下使用最小费用最大流模型进行区域调度且周期时长为 20 分钟时，每小时接单率的下四分位数到达 98%，乘客每小时平均等待时间中位数由自由调度模型的 4 分钟减少到 1 分钟。当区域调度周期为 20 分钟时，平台的接单率和乘客平均等待时间达到最优。20 分钟为最佳周期原因是：周期时长较短时需求不平滑；小区的大小会对周期较短的区域调度产生影响；周期过长会造成区域调度不够及时等。

该研究成果具有很好的扩展性，在未来的研究中，需要解决以下问题：第一，目前采用曼哈顿区域内 1 分钟的出行需求和可用车辆分布建立权值二分图模型，该模型求解速度较慢。参考部分学者的做法，可以以交通小区为单位分别建立权值二分图模型，加快派单算法速度；第二，研究需求预测准确的情况下，服务纽约市曼哈顿区域内出行的最佳车队规模；第三，研究不同最长乘客等待时间下的服务质量与成本变化等。

参 考 文 献

- [1] Kornhauser, A., Chang A., Clark C., Gao J., Korac D., Lebowitz B., Swoboda A., 2013. Uncongested Mobility for All: New Jersey's Area-wide aTaxi System. Princeton University. Princeton, New Jersey.
- [2] Ford, Hillary Jeanette, 2012. Shared Autonomous Taxis: Implementing an Efficient Alternative to Automotive Dependency. Bachelors Thesis in Science and Engineering. Princeton University.
- [3] Du, Y., Hall, R., 1997. Fleet sizing and empty equipment redistribution for center-terminal transportation networks. *Manage. Sci.* 43 (2), 145–157.
- [4] Barth, M., Todd, M., 1999. Simulation model performance analysis of a multiple station shared vehicle system. *Transp. Res. Part C: Emerg. Technol.* 7 (4), 237–259.
- [5] Uesugi, K., Mukai, N., Watanabe, T., 2007. Optimization of vehicle assignment for car sharing system. In: *Knowledge-based Intelligent Information and Engineering Systems*. Springer, pp. 1105–1111.
- [6] Ciari, F., Balmer, M., Axhausen, K.W., 2008. Concepts for a Large Scale Car-Sharing System: Modeling and Evaluation with an Agent-based Approach. Eidgenössische Technische Hochschule, Institut für Verkehrsplanung und Transportsysteme.
- [7] George, D.K., Xia, C.H., 2011. Fleet-sizing and service availability for a vehicle rental system via closed queueing networks. *Eur. J. Oper. Res.* 211 (1), 198–207.
- [8] Chauvet, F., Proth, J.M., Sauer, N., 1997. Management of a pool of self-service cars. *J. Intell. Manuf.* 8 (5), 459–465.
- [9] Fan, W.D., Machemehl, R.B., Lownes, N.E., 2008. Carsharing: dynamic decision-making problem for vehicle allocation. *Transp. Res. Rec.* 2063 (1), 97–104.
- [10] Kumar, V.P., Bierlaire, M., 2012. Optimizing locations for a vehicle sharing system. In: *Proceedings of Swiss Transport Research Conference*. <http://www.strc.ch/conferences/2012/Kumar_Bierlaire.pdf> (accessed 30.01.16.).
- [11] Li, X., Ma, J., Cui, J., Ghiasi, A., Zhou, F., 2016. Design framework of large-scale one-way electric vehicle sharing systems: a continuum approximation model. *Transp. Res. Part B: Methodol.* 88, 21–45.
- [12] Levin, M.W., Kockelman, K.M., Boyles, S.D., Li, T., 2017. A general framework for modeling shared autonomous vehicles with dynamic network-loading and dynamic ride-sharing application. *Comput. Environ. Urban Syst.* 64, 373–383. <http://dx.doi.org/10.1016/j.compenvurbsys.2017.04.006>.
- [13] Rigole, P.-J., 2014. Study of a Shared Autonomous Vehicles Based Mobility Solution in Stockholm
- [14] Chen, T.D., Kockelman, K.M., Hanna, J.P., 2016. Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions. *Transp. Res. Part A Policy Pract.* 94, 243–254. <http://dx.doi.org/10.1016/j.tra.2016.08.020>.
- [15] Fagnant, D.J., Kockelman, K.M., 2015. Dynamic ride-sharing and optimal fleet sizing for a system of shared autonomous vehicles. In: *Transportation Research Board 94th Annual Meeting*.
- [16] Fagnant, D.J., Kockelman, K.M., 2016. Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas. *Transportation (Amst)*. 1–16. <http://dx.doi.org/10.1007/s11116-016-9729-z>.

- [17] Martinez, L.M., Viegas, J.M., 2017. Assessing the impacts of deploying a shared self-driving urban mobility system: an agent-based model applied to the city of Lisbon. Portugal. *Int. J. Transp. Sci. Technol.* 6, 13–27. <http://dx.doi.org/10.1016/j.ijtst.2017.05.005>.
- [18] Viegas, J., Martinez, L., 2016. Shared Mobility: Innovation for Liveable Cities
- [19] Bischoff, J., Maciejewski, M., 2016. Simulation of city-wide replacement of private cars with autonomous Taxis in Berlin. In: *Procedia Computer Science*. <http://doi.org/10.1016/j.procs.2016.04.121>.
- [20] Boesch, P.M., Ciari, F., Axhausen, K.W., 2016. Autonomous vehicle fleet sizes required to serve different levels of demand. *Transp. Res. Rec. J. Transp. Res. Board* 2542, 111–119. <http://dx.doi.org/10.3141/2542-13>.
- [21] Zachariah, J., Gao, J., Kornhauser, A., Mufti, T., 2014. Uncongested mobility for all: a proposal for an area wide autonomous taxi system in New Jersey. In: *Transportation Research Board 93rd Annual Meeting*.
- [22] Fagnant, D.J., Kockelman, K.M., 2014. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. *Transp. Res. Part C Emerg. Technol.* 40, 1–13. <http://dx.doi.org/10.1016/j.trc.2013.12.001>.
- [23] Spieser, K., Treleaven, K., Zhang, R., Frazzoli, E., Pavone, M., 2014. Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: a case study in Singapore. In: Meyer, G., Beiker, S. (Eds.), *Road Vehicle Automation, Lecture Notes in Mobility*. Springer International Publishing, pp. 229–245.
- [24] Zhang, W., Guhathakurta, S., Fang, J., Zhang, G., 2015a. The performance and benefits of a shared autonomous vehicles based dynamic ridesharing system: an agentbased simulation approach. In: *Transportation Research Board 94th Annual Meeting*. Zhang, W., Guhathakurta, S., Fang, J., Zhang, G., 2015b. Exploring the impact of shared autonomous vehicles on urban parking demand: an agent-based simulation approach. *Sustain. Cities Soc.* 19, 34–45. <http://dx.doi.org/10.1016/j.scs.2015.07.006>.
- [25] Burns, Lawrence, Jordan, William, Scarborough, Bonnie, 2013. *Transforming Personal Mobility. The Earth Institute – Columbia University, New York. Center for Information and Society, 2013. Automated Driving: Legislative and Regulatory Action. Stanford University.* <http://cyberlaw.stanford.edu/wiki/index.php/Automated_Driving:_Legislative_and_Regulatory_Acti on>.
- [26] Vazifeh M M , Santi P , Resta G , et al. Addressing the minimum fleet problem in on-demand urban mobility[J]. *Nature*, 2018, 557(7706):534-538.
- [27] 李换平. 汽车云计划下城市自动驾驶车辆调度研究[D]. 哈尔滨工业大学 2015
- [28] 谢榕,潘维,柴崎亮介. 基于人工鱼群算法的出租车智能调度[J]. *系统工程理论与实践*. 2017(11)
- [29] 范子明. 基于大连出租车 GPS 调度系统的应用研究[D]. 大连海事大学 2017
- [30] 郭瑞雪. 基于 BP 神经网络的网约车出行需求短时预测[D]. 北京交通大学 2017
- [31] 贾婷. 基于乘客需求及分布的出租车调度方法技术研究[D].成都理工大学,2013.
- [32] 王郑委. 基于大数据 Hadoop 平台的出租车载客热点区域挖掘研究[D].北京交通大学,2016.
- [33] 唐诗韵. 基于 GPS 数据的空驶出租车寻客路线推荐研究[D].西南交通大学,2017.
- [34] 刘丽. 基于出租车 GPS 数据的高效益寻客推荐策略研究[D].浙江大学,2017.

- [35] 陈轶非,李治军,姜守旭.关于出租车载客地点序列推荐技术的研究[J].智能计算机与应用,2013,3(06):70-73.
- [36] 于扬红雪. 出租车需求量预测模型的研究[D].大连海事大学,2017.
- [37] 朱本常. 基于时间序列的网约车在线司机与乘客发单的供需预测[D].华中科技大学,2019.
- [38] Friedman J H , Bentley J L , Finkel R A . An Algorithm for Finding Best Matches in Logarithmic Expected Time[J]. ACM Transactions on Mathematical Software, 1977, 3(3):209-226.
- [39] 罗群. 基于 OpenCV 的包裹体显微自动分析系统[D].成都理工大学,2015.
- [40] Santi P , Resta G , Szell M , et al. Quantifying the benefits of vehicle pooling with shareability networks[J]. Proceedings of the National Academy of Sciences, 2014, 111(37):13290-13294.
- [41] 诸葛雪玉,张文闯,谭娅琦,徐萌甜,樊颖,张小丽.基于 POI 空间聚类的交通小区划分方法改进[J].山西建筑,2018,44(26):7-8.
- [42] 孙惠泉. 图论及其应用[M]. 北京: 科学出版社,2004:89
- [43] J.Edmonds, Paths, trees, and flowers[J], Cand.J.Math,1965,17:449-467.
- [44] Gabow, Harold N . An Efficient Implementation of Edmonds\' Algorithm for Maximum Matching on Graphs[J]. Journal of the ACM, 1976, 23(2):221-234.
- [45] Micali S , Vazirani V V . An $O(\sqrt{|v|} |E|)$ Algorithm for Finding Maximum Matching in General Graphs[J]. Foundations of Computer Science Annual Symposium on, 1980:17-27.
- [46] 张先迪, 李正良. 图论及其应用[M]. 高等教育出版社,2005:111-116.
- [47] 《运筹学》教材编写组. 运筹学(第三版)[M]. 清华大学出版社, 2005.
- [48] Edmonds J , Karp R M . Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems[J]. Journal of the ACM, 1972, 19(2):248-264.
- [49] 高随详. 图论与网络流理论[M]. 北京:高等教育出版社, 2009.

附录 A 区域通行时间数据

表 A1 交通小区内通行时间表

Tab.A1 Travel time in traffic zones

小区编号	交叉口数量	平均值	标准差	25%分位	50%分位	75%分位	最大值
0	139	3.2	1.5	2.1	3.2	4.3	9.8
1	150	3.9	1.9	2.5	3.8	5.2	11.2
2	181	3.5	2.2	2.1	3.1	4.3	17.8
3	111	3.6	1.7	2.3	3.5	4.8	9.7
4	198	4.0	2.1	2.6	3.8	5.2	16.8
5	168	4.1	2.3	2.4	3.8	5.4	19.0
6	121	5.1	2.2	3.4	5.1	6.7	11.6
7	164	4.5	2.2	2.9	4.3	5.7	18.7
8	119	3.5	1.6	2.4	3.5	4.6	11.1
9	139	3.7	1.9	2.3	3.5	4.8	13.6
10	219	4.3	1.9	2.8	4.3	5.7	11.7
11	124	4.1	2.4	2.5	3.8	5.3	25.8
12	151	4.4	1.9	3.0	4.3	5.7	11.4
13	200	3.4	1.4	2.3	3.3	4.3	10.4
14	96	3.2	2.5	1.6	2.8	4.2	19.5
15	127	3.7	2.2	2.2	3.4	4.8	19.7
16	180	3.5	1.6	2.3	3.4	4.5	9.7
17	190	3.5	1.5	2.4	3.5	4.5	9.6
18	168	5.0	2.3	3.3	4.9	6.5	18.9
19	144	3.0	1.9	1.7	2.6	3.7	13.3
20	138	4.2	2.2	2.6	4.0	5.5	16.0
21	134	3.4	1.8	2.2	3.2	4.4	16.3
22	128	3.3	1.8	2.0	3.2	4.5	12.4
23	137	3.5	1.9	2.2	3.3	4.6	14.7
24	177	4.2	1.9	2.8	4.1	5.5	12.4
25	145	4.1	2.5	2.5	3.7	5.2	20.4
26	107	4.6	2.4	2.9	4.4	5.9	16.2
27	119	3.8	2.4	1.9	3.6	5.3	14.4
28	150	4.4	2.1	2.8	4.4	5.8	14.6
29	84	4.1	2.0	2.6	4.0	5.4	11.9

(单位: 分钟)

表 A2 区域间出行行程时间

Tab.A2 Travel time between traffic zones

OD	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0.0	32.1	15.3	18.9	37.3	10.5	25.6	28.2	31.3	10.2	38.5	8.9	26.9	35.7	19.8
1	27.4	0.0	32.2	17.6	13.4	29.8	10.5	7.5	7.9	24.2	16.7	21.0	10.3	8.5	35.9
2	17.7	40.6	0.0	32.2	46.7	9.5	34.4	38.5	36.0	20.2	47.1	24.0	32.6	40.4	8.6
3	16.9	19.4	25.5	0.0	26.6	23.3	12.4	15.2	24.2	13.1	28.0	11.6	19.2	26.3	29.8
4	32.3	13.5	37.6	25.4	0.0	34.9	21.2	15.7	15.0	30.9	7.6	25.7	20.1	10.9	40.9

基于车联网环境的共享自动驾驶汽车调度研究

5	10.5	36.5	8.7	27.4	41.9	0.0	31.3	33.3	33.6	16.5	42.7	17.9	30.3	37.9	12.9
6	23.7	10.7	27.4	11.7	21.7	25.7	0.0	8.6	14.0	17.6	23.9	18.9	9.3	15.8	31.8
7	22.3	7.8	27.9	13.6	15.7	25.1	8.6	0.0	14.4	22.0	17.8	15.7	11.9	15.2	30.9
8	27.3	7.9	29.6	23.0	15.3	28.9	13.5	14.0	0.0	19.8	15.3	27.3	7.3	6.3	34.4
9	9.9	27.4	15.9	14.4	35.1	14.7	19.2	26.1	23.3	0.0	34.9	13.2	19.3	27.7	20.7
10	33.5	17.4	38.5	27.3	8.1	36.0	23.4	18.4	14.5	30.9	0.0	27.1	19.5	11.7	42.1
11	8.2	25.7	19.7	13.1	30.2	16.2	22.5	21.1	31.1	11.9	31.6	0.0	27.8	32.5	23.0
12	23.6	10.7	25.9	18.7	20.7	25.2	9.4	12.2	7.7	16.0	20.6	24.4	0.0	12.0	30.7
13	31.0	8.8	33.5	25.1	11.3	32.6	15.8	15.1	5.8	23.7	12.4	28.3	11.3	0.0	38.3
14	23.9	46.9	10.3	38.6	52.4	15.8	41.1	43.8	42.8	27.1	53.2	29.1	39.4	47.1	0.0
15	7.9	33.8	10.4	24.2	40.6	6.8	27.2	32.8	29.0	11.8	40.5	15.9	25.6	33.4	15.4
16	30.8	8.4	35.9	22.7	7.1	33.4	16.9	12.3	10.5	28.2	11.4	24.3	15.9	6.6	39.5
17	33.8	12.6	36.5	28.3	7.4	35.3	20.0	18.1	9.7	26.7	7.8	30.0	14.8	6.2	41.2
18	19.0	15.0	24.6	7.6	21.9	21.9	9.6	10.0	21.2	17.7	23.6	12.4	17.2	22.2	27.7
19	15.0	38.9	5.2	30.4	45.4	6.3	32.4	37.4	34.1	18.1	45.5	22.6	30.7	38.4	11.2
20	12.0	22.3	21.9	8.1	27.2	19.2	17.8	17.9	27.3	10.7	28.6	6.5	23.6	29.0	25.7
21	6.7	33.0	12.6	22.6	37.5	7.0	29.3	28.4	34.6	15.7	38.8	12.4	31.1	38.0	16.0
22	6.9	31.7	14.4	18.7	38.7	11.4	23.7	29.8	27.8	6.6	39.0	11.8	23.8	32.2	19.3
23	14.0	22.3	17.5	11.4	31.2	16.8	14.0	21.3	18.7	6.2	30.8	16.9	14.3	23.2	22.2
24	28.7	13.1	34.3	22.3	6.7	31.6	19.3	13.3	16.5	29.3	10.5	22.1	21.0	12.5	37.4
25	7.0	28.6	16.2	17.9	33.1	12.8	26.0	24.0	34.1	15.6	34.4	7.4	31.2	35.3	19.3
26	25.0	7.3	30.7	18.1	11.0	27.9	14.3	8.3	12.2	25.5	13.8	18.4	15.9	11.6	33.7
27	21.4	44.3	6.9	35.9	50.1	13.4	38.2	41.8	39.9	24.1	50.7	27.2	36.5	44.2	5.0
28	18.0	17.3	20.7	12.5	26.9	20.0	9.4	16.6	14.3	10.3	26.7	20.2	9.6	18.6	25.4
29	17.0	19.3	24.0	6.7	23.8	21.2	16.3	14.6	25.5	14.8	25.2	10.3	23.1	26.1	27.1

OD	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	7.1	36.2	38.4	23.0	13.3	13.6	7.0	6.5	15.2	33.7	7.2	30.8	17.6	20.0	19.4
1	28.5	8.3	12.5	13.0	31.6	20.1	26.1	27.2	21.0	12.4	24.0	7.3	34.2	16.5	19.1
2	11.9	44.7	43.1	33.8	5.6	28.2	14.1	17.1	23.1	44.5	19.6	41.5	5.5	27.2	31.4
3	20.4	24.9	29.7	8.3	24.1	7.5	19.4	16.9	10.5	23.0	15.3	20.0	27.7	12.2	7.3
4	35.0	7.0	7.6	20.1	37.6	25.2	30.7	33.5	29.4	7.0	28.7	10.8	39.5	26.0	24.3
5	6.8	40.4	40.6	28.7	5.7	22.3	6.8	12.1	20.5	38.9	13.8	35.9	11.1	24.7	25.8
6	22.6	17.3	20.2	9.0	26.1	16.4	23.8	21.2	13.6	19.7	22.0	15.5	29.7	9.3	16.4
7	26.2	12.8	18.4	8.4	28.2	14.9	20.8	24.4	20.1	12.8	18.7	8.9	29.6	15.9	13.7
8	24.4	10.7	10.1	19.3	27.8	25.8	30.1	22.9	16.9	16.5	30.1	12.2	31.9	13.3	25.7
9	9.7	32.4	30.5	20.5	14.1	12.4	15.2	5.7	7.2	34.1	15.2	30.7	18.2	12.4	17.1
10	35.0	12.1	7.7	22.1	38.1	26.7	32.1	33.7	28.7	11.2	30.1	14.7	40.5	25.6	25.9
11	14.3	29.2	34.6	16.3	19.5	7.4	11.1	10.8	16.5	26.5	6.4	23.6	21.6	21.4	12.1
12	20.7	16.6	15.6	16.0	24.1	22.4	26.6	19.2	12.6	21.0	27.1	16.5	28.2	8.8	23.0
13	28.3	6.6	6.8	20.7	31.8	27.4	32.5	26.9	21.0	12.7	31.3	11.4	35.9	17.4	26.7
14	19.1	51.0	49.9	39.3	13.6	33.5	19.2	24.1	29.9	49.4	24.5	46.5	5.8	34.0	36.3

15	0.0	38.0	36.1	28.0	7.9	20.2	8.3	7.4	15.7	39.6	12.8	36.3	12.9	20.0	25.6
16	32.4	0.0	7.8	17.5	35.5	23.8	29.4	31.1	25.9	7.7	27.4	7.4	37.9	22.0	22.7
17	31.3	7.6	0.0	23.4	34.7	29.3	34.6	29.8	24.1	11.7	33.0	13.8	38.8	20.9	28.5
18	22.5	19.6	25.2	0.0	24.7	10.6	17.6	20.6	15.2	18.5	15.5	15.1	26.4	12.8	9.3
19	8.9	43.1	41.2	32.7	0.0	26.8	11.6	14.3	21.2	43.7	18.7	40.7	8.3	25.3	30.5
20	17.3	26.2	31.4	12.3	21.2	0.0	14.4	13.3	12.5	23.6	10.1	20.6	24.0	17.1	7.8
21	7.7	36.4	40.4	23.9	10.6	17.0	0.0	11.3	20.3	33.8	8.1	30.9	14.6	24.8	20.7
22	6.5	36.6	35.0	24.2	12.5	15.4	11.4	0.0	11.8	36.5	12.3	33.5	16.8	16.9	21.0
23	12.2	27.9	26.1	15.8	15.7	13.4	18.4	9.8	0.0	31.1	19.3	27.3	19.8	7.3	15.2
24	33.2	7.6	11.5	17.1	34.8	21.7	27.2	31.1	29.8	0.0	25.1	8.4	36.0	26.2	20.9
25	11.6	32.0	37.5	19.4	16.2	12.1	7.6	12.0	20.5	29.4	0.0	26.5	18.0	25.2	15.7
26	29.4	8.1	14.3	12.6	31.1	18.0	23.5	27.4	25.4	8.1	21.5	0.0	32.4	21.5	17.0
27	16.1	48.4	46.9	37.1	10.3	31.5	17.3	21.2	27.0	47.5	22.6	44.6	0.0	31.0	34.4
28	15.4	23.1	21.9	12.5	18.9	17.2	21.7	13.7	6.6	27.1	22.7	22.8	23.0	0.0	18.0
29	21.6	22.8	28.3	9.3	24.3	6.9	16.9	18.3	13.5	20.1	13.6	17.2	25.7	17.0	0.0

(单位: 分钟)

附录 B 各模型对比图

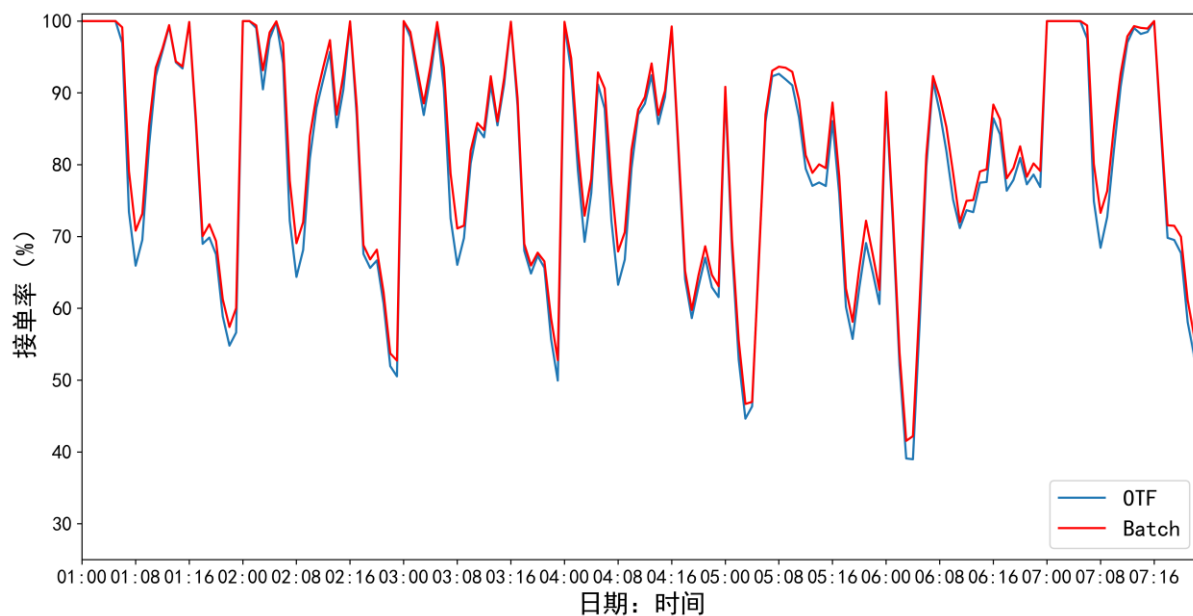


图 B1 直接调度模型与批次调度模型接单率对比图

Fig.B1 Comparison of order-served rate for OTF_Model and Batch_Model

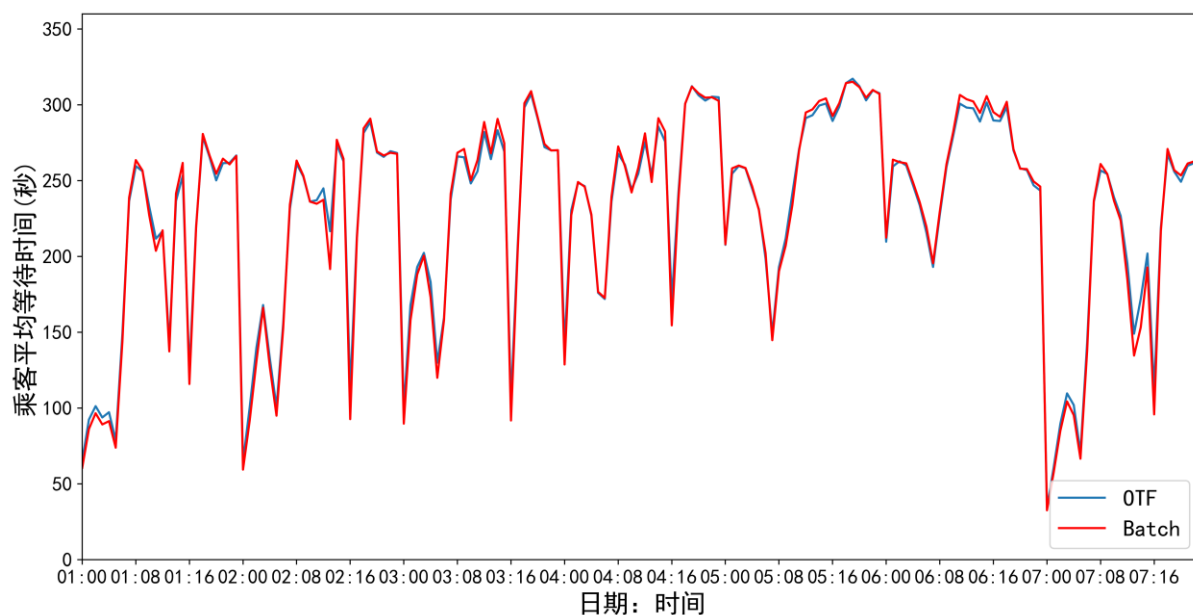


图 B2 直接调度模型与批次调度模型乘客平均等待时间对比图

Fig.B2 Average waiting time for OTF_Model and Batch_Model

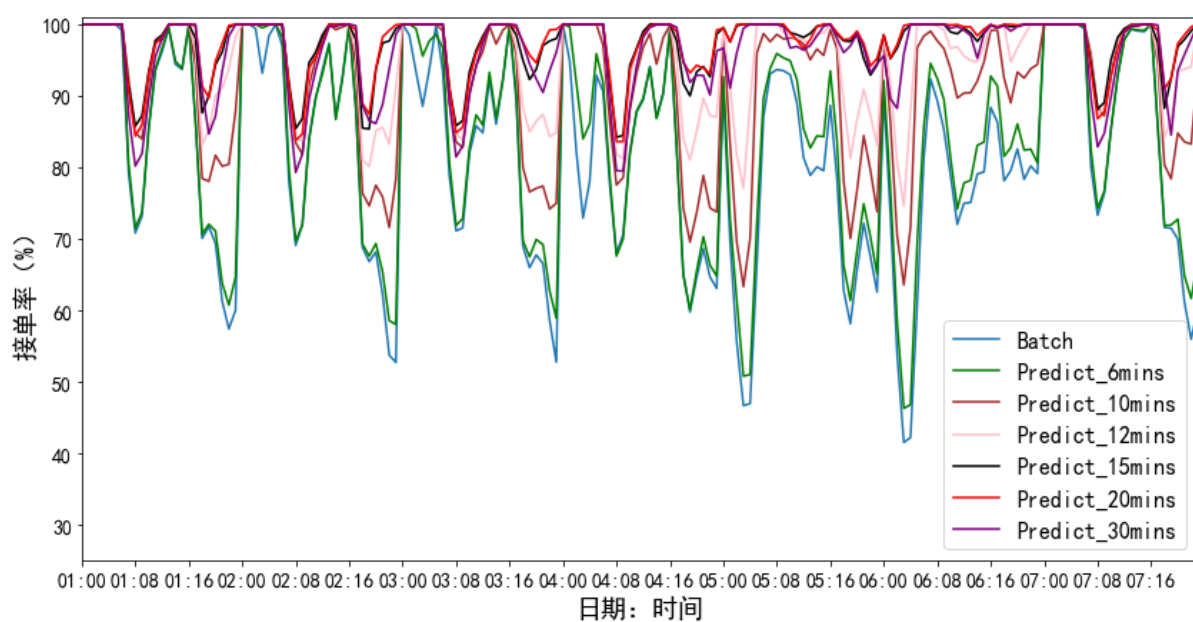


图 B3 批次调度模型与预测调度模型接单率对比图

Fig.B3 Comparison of order-served rate for Batch_Model and Predict_Models

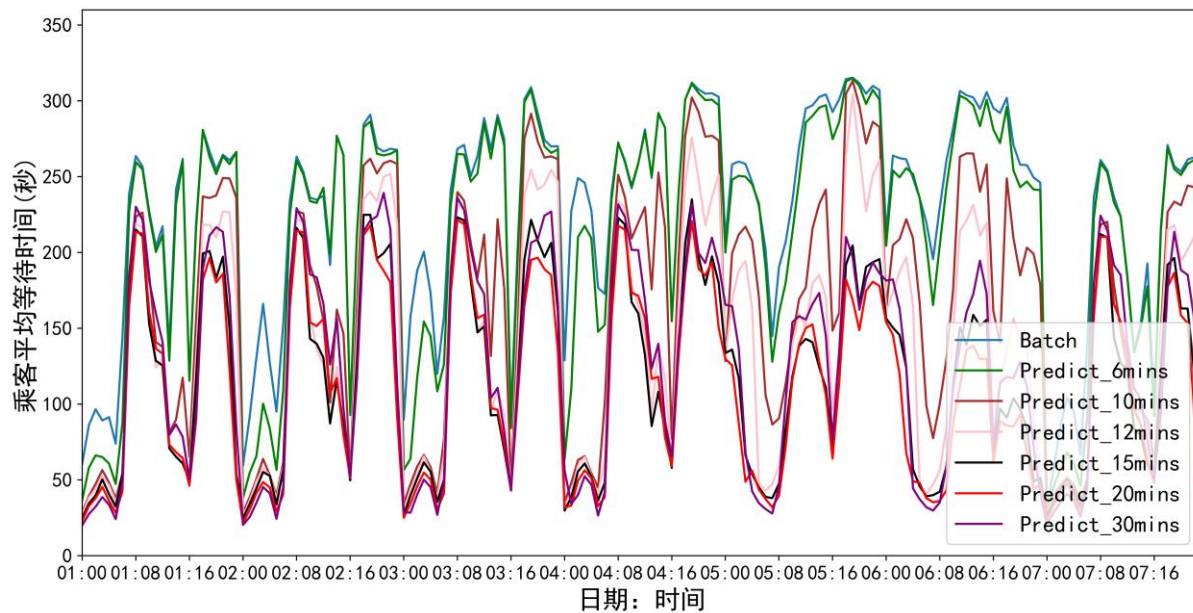


图 B4 批次调度模型与预测调度模型乘客平均等待时间对比图

Fig.B4 Average waiting time for Batch_Model and Predict_Models

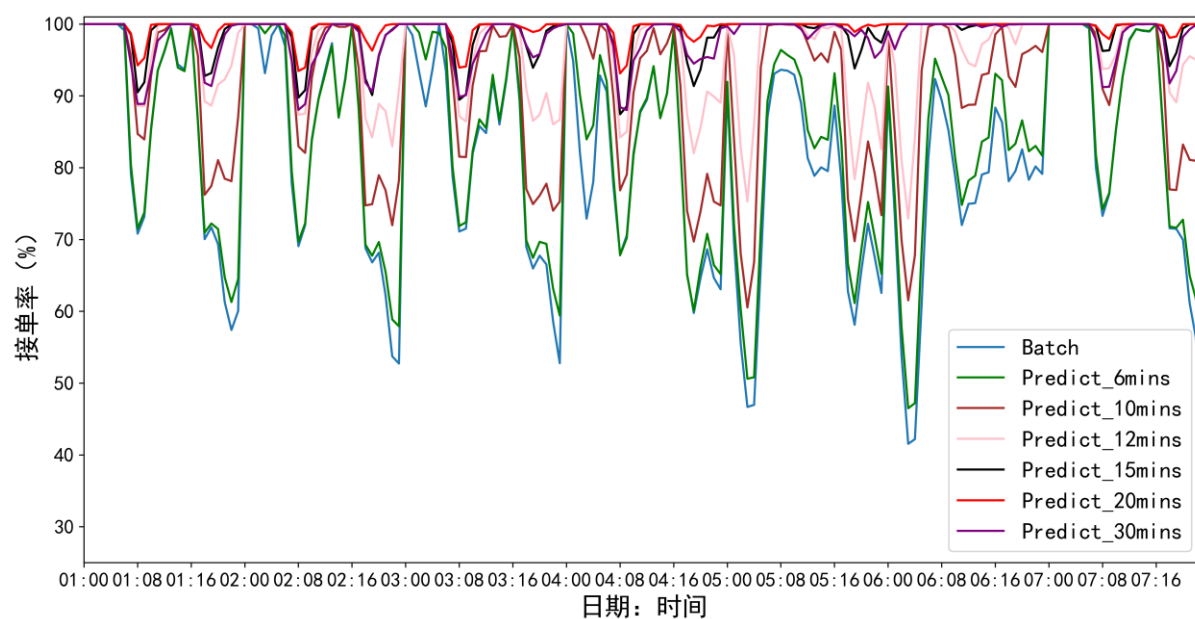


图 B5 批次调度模型与改进的预测调度模型接单率对比图

Fig.B5 Comparison of order-served rate for Batch_Model and improved Predict_Models

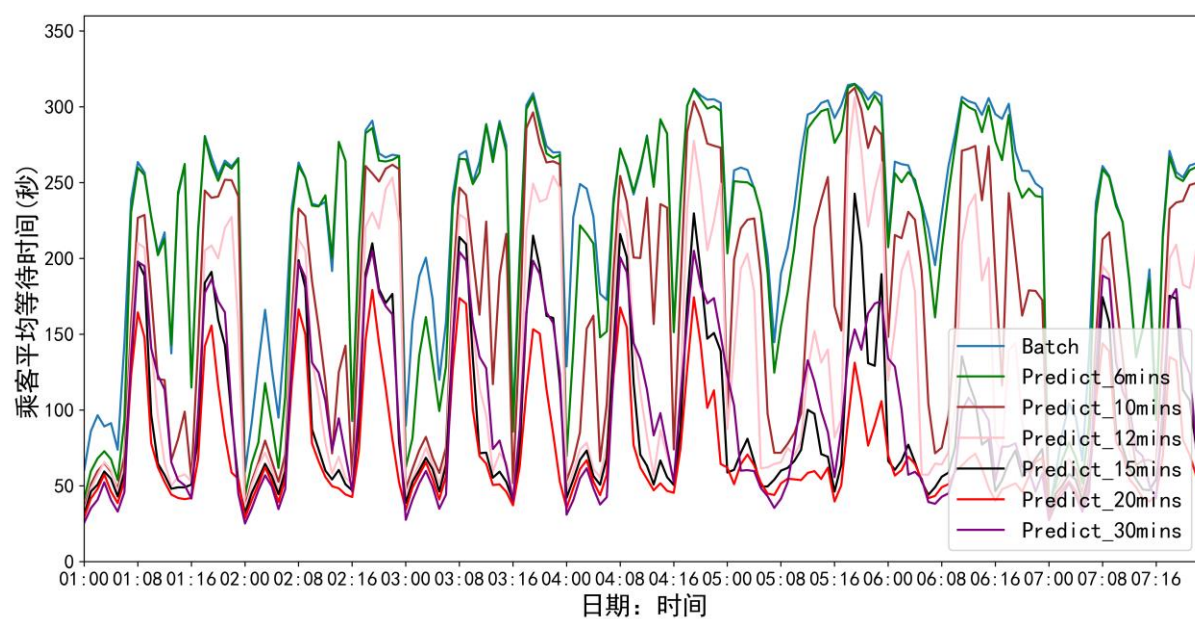


图 B6 批次调度模型与改进的预测调度模型乘客平均等待时间对比图

Fig.B6 Average waiting time for Batch_Model and improved Predict_Models

致 谢

时光似箭，岁月如梭，转眼间两年的研究生生活即将结束。在此期间，很多人帮助过我，在我找工作期间，有人鼓励我给我信心，在我难过的时候有人安慰我，还有很多很多……感谢这两年来帮助过我的所有人。

首先要感谢我的导师王仲老师，感谢您这两年在生活上学习上对我的关心与指导，在我撰写毕业论文时认真负责地指导，在我迷茫时给与我指引，让我想清楚未来要走的道路。您认真负责的工作态度深深影响了我，让我受益匪浅，感谢您！我还要感谢所有帮助过我的老师，所谓传道授业解惑也，谢谢各位老师教授我专业知识，感谢你们！

其次，我要感谢我的父母，在我撰写毕业论文期间给与我安慰与鼓励。还要感谢我的女朋友秦雪璠，在我找工作怀疑自己时一直默默支持着我，为我出谋划策，安慰我静下心来撰写论文。还要感谢师门的各位成员，这是一个团结友爱的大家庭。还要感谢张路、钟季汝、侯琳师姐、宫小雯、王明莉、梁亚林、柳爽、李旭丰、程荣等好朋友对我的帮助，每次在我遇到问题时你们都会十分耐心与我交流帮我解答，与你们相处的时光是研究生生活中最为宝贵的回忆。特别感谢好朋友张路在我撰写论文期间给与的大力支持与帮助。

最后感谢答辩评审委员会的老师，感谢各位老师对我论文的指导！

大连理工大学学位论文版权使用授权书

本人完全了解学校有关学位论文知识产权的规定，在校攻读学位期间论文工作的知识产权属于大连理工大学，允许论文被查阅和借阅。学校有权保留论文并向国家有关部门或机构送交论文的复印件和电子版，可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印、或扫描等复制手段保存和汇编本学位论文。

学位论文题目：_____

作者签名：_____ 日期：_____年____月____日

导师签名：_____ 日期：_____年____月____日