

分类号:U491

10710-2021134084



長安大學

硕 士 学 位 论 文

基于短时需求预测的
出租车区域供需再平衡调度

聂梦婕

导师姓名职称	邵海鹏 教授		
申请学位类别	工学硕士	学科专业名称	交通运输工程
论文提交日期	2024年4月11日	论文答辩日期	2024年5月25日
学位授予单位	长安大学		

Regional Supply-demand Rebalancing Dispatch of Taxis Based on Short-term Demand Forecasting

A Thesis Submitted for the Degree of Master

Candidate: Nie Mengjie

Supervisor: Prof. Shao Haipeng

Chang'an University, Xi'an, China

答辩委员会名单

答辩委员会	姓名	职称	工作单位
主席	白玉	教授	同济大学
委员	宗婷	高级工程师	苏交科集团股份有限公司
委员	颜建强	教授	西北大学
委员	陈建忠	副教授	西北工业大学
委员	李聪颖	副教授	西安建筑科技大学
秘书	高亚楠	副教授	长安大学运输工程学院

摘要

随着自动驾驶技术的发展，结合出租车服务和自动驾驶技术的共享自动驾驶车服务应运而生。在无需考虑司机利益的情况下，提供共享自动驾驶服务的运营商更有可能基于全局对车队进行调度，以实现区域间出租车出行需求和车辆供给的平衡。在此背景下，本文基于出租车短时订单需求预测，针对有人驾驶的出租车和共享无人车组成的混合车队构建区域调度模型，以更好地实现供需匹配。

在对出租车订单数据集进行时空特征分析的基础上，明确出租车订单的短时需求预测任务。引入长短时记忆神经网络、时间卷积网络进行预测，尝试在模型中添加时序模式注意力机制对其进行改进，并用四种损失函数对模型进行训练并评价预测效果。结果表明，在本研究中采用 Huber 损失对模型进行训练时，各个模型的损失值下降速度最快，训练效率最高；同时，引入时序模式注意力机制的 TCN 模型在预测精度、模型稳定性和模型泛化能力等方面表现较优。

在出租车订单需求短时预测的基础上，构建区域供需再平衡调度模型。考虑有司机驾驶的出租车和共享无人车产生的不同调度成本，明确参数、决策变量、目标函数和约束条件，将车辆调度问题抽象为组合优化问题。

最后以曼哈顿和芝加哥为例对本文建立的调度模型进行实例分析，引入 CP-SAT 开源求解器对模型进行求解并输出结果。结果表明，采用本文搭建的出租车短时订单需求预测模型以及混合车队调度模型，对共享无人车和有司机驾驶的出租车组成的混合车队进行区域预调度，能够使车辆在时间限制内完成调度任务，有效满足各区域的用车需求，并在生成调度方案时区分两类车辆，使系统调度成本尽可能小的同时限制了司机的调度时间和距离。

关键词：短时需求预测，深度学习，注意力机制，车辆调度，组合优化问题

Abstract

With the gradual advancement of autonomous driving technology, a shared autonomous vehicle service that combines taxi service and autonomous driving technology has emerged. Without considering the interests of drivers, operators providing shared autonomous driving services are more likely to schedule fleets on a global basis to achieve a balance between inter-regional travel demand and vehicle supply. In this context, based on the prediction of taxi short-term order demand, this paper constructs a regional scheduling model for a mixed fleet of manned taxis and shared driverless cars to better achieve supply and demand matching.

Based on the analysis of spatial and temporal characteristics of taxi order data sets, a short-term demand forecasting task of taxi order is defined. Long short-term memory neural network and time convolution network are introduced to predict, and attention mechanism is added to the model to improve it. Four loss functions are used to train the model and evaluate the prediction results. The results show that when the Huber loss function is used to train the model in this study, the loss value of each model decreases the fastest and the training efficiency is the highest. At the same time, the TCN model with attention mechanism performs better in prediction accuracy, model stability, and model generalization ability.

Based on the short-term forecast of taxi order demand, a regional supply and demand rebalancing dispatch model is constructed. Considering the different dispatch costs incurred by chauffeur-driven taxis and shared driverless vehicles, the parameters, decision variables, objective functions and constraints are clarified, and the vehicle dispatch problem is abstracted as a combinatorial optimization problem.

Finally, Manhattan and Chicago are taken as examples to simulate the area scheduling model established in this paper. The CP-SAT open-source solver is introduced to solve the model and output the results. The results show that using the taxi short-term order demand forecasting model and the hybrid fleet scheduling model built in this paper, the regional pre-scheduling of the hybrid fleet composed of shared driverless vehicles and taxis driven by drivers can enable the vehicles to complete the scheduling task within the time limit, effectively meet the vehicle demand in each region, and distinguish the two types of vehicles when generating the scheduling scheme, so that the system scheduling cost is as small as possible while limiting the driver's scheduling time and distance.

Key words: Short-term demand forecasting, Temporal convolutional network, Attention mechanism, Vehicle dispatch, Combinatorial optimization problem

目 录

第一章 绪论.....	1
1.1 研究背景与意义.....	1
1.2 国内外研究进展.....	3
1.2.1 出租车订单需求预测方法.....	3
1.2.2 传统出租车、共享无人车调度方法.....	4
1.2.3 研究进展小结.....	7
1.3 主要研究内容.....	7
1.4 论文组织结构.....	8
第二章 数据处理及时空特征分析	11
2.1 数据描述与预处理.....	11
2.1.1 芝加哥出租车订单数据.....	11
2.1.2 纽约出租车订单数据集.....	12
2.2 数据时空特征分析.....	14
2.2.1 出租车订单时间特征分析.....	14
2.2.2 出租车行程空间特征分析.....	17
2.2.3 出租车行程时空特征分析.....	19
2.3 本章小结.....	23
第三章 基于深度学习的短时订单需求预测	25
3.1 出租车订单量预测问题描述	25
3.2 基于深度学习的时间序列预测方法	26
3.2.1 长短时记忆神经网络.....	26
3.2.2 时间卷积网络.....	27
3.2.3 注意力机制.....	28
3.3 出租车短时订单需求预测模型构建	30
3.3.1 数据处理方法.....	30
3.3.2 模型细节与超参数确定.....	31
3.4 出租车订单量预测结果与模型评价	34
3.4.1 纽约曼哈顿出租车订单量预测.....	34
3.4.2 芝加哥出租车订单量预测.....	39
3.5 本章小结.....	43
第四章 车辆区域供需再平衡	45
4.1 出租车区域供需再平衡问题描述	45
4.2 出租车区域供需再平衡调度模型构建	46
4.2.1 区域调度相关参数.....	46

4.2.2 区域调度模型决策变量.....	49
4.2.3 区域调度模型目标函数.....	49
4.2.4 区域调度的约束条件.....	49
4.3 出租车区域供需再平衡调度模型求解	51
4.4 本章小结	53
第五章 实例分析	55
5.1 调度模型实例应用参数设置	55
5.1.1 交通小区划分.....	55
5.1.2 路网结构与路径计算.....	57
5.1.3 车辆总数与车辆状态.....	59
5.1.4 调度模型实例分析其他相关参数.....	61
5.2 出租车区域调度结果分析	63
5.2.1 纽约曼哈顿出租车区域调度方案分析	63
5.2.2 芝加哥区域调度方案分析	66
5.3 本章小结	67
总结与展望.....	69
参考文献.....	71
附录 A.....	77
附录 B	81

第一章 绪论

1.1 研究背景与意义

出租车是城市公共交通服务系统的必要组成部分，是城市居民重要的出行方式之一。尽管搭乘出租车的费用远高于乘坐城市公共交通，但因为出租车服务便利、快捷、可定制，所以在城市出行中一直备受青睐。在出现之初，出租车一般采用“巡游”这一寻客和载客模式。在这种模式下，出租车司机驾驶出租车辆在道路上持续驾驶，而有搭乘需求的乘客则在路边等待出租车辆经过。在这种经营模式下，出租车供给和乘客需求之前往往存在着无法匹配的情况，且出租车空驶时间偏长，导致资源的浪费。随着通讯技术的发展，逐渐出现了电话叫车这一服务形式，乘客通过拨打出租电话，将乘车需求告知出租车调度中心或出租车司机，大大提高了司乘匹配的效率。

近年来依托互联网技术，出现网络预约出租汽车经营服务（网约车），进一步拓宽了出租车的经营模式，便捷了人们的日常出行。乘客可以通过第三方网络平台即时打车，或预约打车服务；司机也可以通过平台接取订单，或完成激励任务，实现了供需信息的整合。通过网约车平台和出租车公司的资源整合，传统巡游出租车也加入到网络平台叫车服务的阵营之中，出租车系统的整体运营效率得到进一步优化。

随着车联网和自动驾驶技术的发展，自动化和共享出行（Automation and Shared Mobility）成为未来交通发展的一大趋势。结合出租车服务和自动驾驶技术的共享自动驾驶车（Sharing Autonomous Vehicles，SAV）服务应运而生，成为城市出租车服务的一种新形式。国内，无人驾驶出租车服务于2022年5月率先在北京首钢园落地，至今已有南京、福州等多个城市开展试点运营；并且，在2024年国内最大的网约车平台“滴滴”也与自动驾驶技术科技公司达成合作，推进自动驾驶网约车服务。国外，自动驾驶公司Waymo在对“信任试乘者”测试的基础上，于2022年11月宣布将在美国凤凰城市中心向公众开放自动驾驶网约车服务；2024年3月14日起，将向洛杉矶部分公众免费开放自动驾驶服务。尽管未来几十年自动驾驶对交通系统的影响仍存在很大的不确定性，但从目前的研究和应用的趋势来看，它或许很快将成为交通系统的一部分，进而对传统交通模式产生颠覆性的影响，同时也会带来出租车运营模式的变革。

随着各类出租车服务模式的快速发展，整体市场已经进入到了平台模式主导的发展阶段，是平台经济和算法经济蓬勃发展的典型领域^[1]。作为城市交通系统的重要组成部分，出租车运营不仅关系到运营商和驾驶员的切身利益，也影响着出租车行业的服务水平。

平和城市整体的出行效率。因此，运营平台须立足于服务效率、服务水平等方面，不断提高响应速度，这就要求建立更加科学有效的车辆调度方式。出租车辆的调度主要分为派单调度和区域调度^[2]，派单调度是在乘客发起订单后，在一定的范围内搜索出租车辆并将订单推荐给合适的车辆；区域调度功能则是考虑未来短时间内各区域的出租车需求，对空闲车辆进行调度，减少部分区域用车紧张而部分区域车辆富余的情况，使车辆提前在区域内就位，在乘客发起订单需求后可以尽快满足需求，减少派单调度时间。通过优化派单调度和区域调度，可以减少车辆空驶时间与乘客平均等待时间，并优化资源配置，提高利润空间。

出租车服务一般由司机、乘客、和调度平台共同组成，在进行调度时，应综合考虑司机、平台的收益和乘客的时间效益。当前网约车运营平台对于参与区域调度的车辆会提供一定补贴，司机按照规定的导航路线行驶至调度目的地，且中途未接单的情况下，可以拿到平台的调度补偿奖励。但这种调度补偿机制对司机的实际调度距离以及调度时间考虑不够充分，若司机接取调度任务，获得的补偿奖励往往难以弥补放弃接单而造成的损失，因此网约车和出租车司机参与调度的积极性普遍不高。而如果平台增加调度补偿奖励，则会压缩盈利空间，难以取得平台和司机间的平衡。由于共享无人车没有司机驾驶，因此在完成订单以及参与调度时不会产生人力成本，也无需考虑司机的利益损失。相较于巡游出租车服务和网约车服务而言，不从个体角度而从系统角度决策寻客路线，是共享无人车相较于传统车辆的一大优势。从这种角度看，共享无人车运营平台和巡游出租车、网约车平台的调度方案应有所不同。尤其是进行区域调度时，在无需考虑司机利益的情况下，运营商更有可能基于全局对车队进行调度，以实现各区域出行需求和车辆供给的再平衡。

虽然共享无人车已经在部分城市展开试点，但短期内不能完全替代传统出租车和网约车服务。考虑公众对无人驾驶的接受程度以及共享无人车的投放规模，一段时间内会形成共享自动驾驶服务与巡游出租车、网约车服务并行的局面。由于共享无人车和巡游出租车、网约车服务的调度方案存在差异，盲目采用巡游出租车、网约车服务的调度方案，或者仅考虑共享无人车的调度方案在应用时无法保证效益的最大化。因此，本文考虑共享无人车与有人驾驶的出租车在运营成本等方面的差异，重点关注类出租车服务的区域间供需再平衡，优化区域调度方案的生成。

1.2 国内外研究进展

实现高效率的出租车区域预调度的前提是对短时间内各区域乘客需求的准确预测，因此订单需求预测是出租车运营的一项重要任务。并且，由于本文研究的调度对象为共享无人车与有人驾驶的出租车组成的混合车队，因此应考虑二者在调度方法上的共通点及差异性。

1.2.1 出租车订单需求预测方法

出租车订单需求预测是交通预测类问题中备受关注的课题之一。由于当前未出现大规模投入运营的共享无人车，并且在以往的研究中，传统出租车与网约车服务在订单需求预测任务上没有明显的差异，因此本文在订单需求预测部分的谈论中，不对传统出租车、网约车和共享无人车进行区分。

根据待预测时间段的长度，交通预测问题通常可分为两类：短时预测（5-30分钟）和中长时预测（大于30分钟）^[3]。对于出租车运营来说，长时需求预测可以提供长期的市场趋势和需求情况，帮助平台或运营公司基于未来需求进行车队规模的调整、新车辆的投放和出租车司机的招募等。而短时需求预测主要关注未来较短时间内的需求变化，可以辅助指导出租车辆的实时调度和灵活应对，并帮助司机避开交通拥堵、选择更高效的接客路线。两者结合，可以减少车辆的空驶率，提高车辆利用率，以提高平台的整体收益，并提升乘客的满意度，有助于实现整体运营的最佳化。

国内外学者在交通预测的方法上开展了大量研究。早期的研究大多依赖于模型驱动的方法，这类方法试图建立较为贴合实际情况的交通模型，以预测道路网中的交通流或出行、驾驶等行为。然而交通数据受到多种因素的影响，致使准确的交通模型难以建立，影响了预测精度。随着数据储存能力的飞速提升以及数据处理技术的快速发展，交通大数据、车联网等逐渐兴起，数据驱动的交通预测方法应运而生。数据驱动方法不是通过分析交通系统的物理特性来建立模型，而是根据历史交通数据的变化规律来推断未来的变化趋势，因此具有更高的灵活性^[4]。交通预测的数据驱动方法的相关研究大致经历了三个阶段^[5]：（1）时间序列方法；（2）传统机器学习方法，如支持向量机、随机森林等；（3）深度学习方法，如长短记忆神经网络、时空卷积网络等。

1. 时间序列方法

交通数据在时间维度上具有一定的规律性，因此部分研究者以基本的时间序列预测方法为基础，进行交通预测，如指数平滑法^[7]、趋势移动平均法^[8]、卡尔曼滤波算法^[9]、

谱分析^[10]、差分整合移动平均自回归模型等^[11]。其中差分整合移动平均自回归(Autoregressive Integrated Moving Average Model, ARIMA)模型在交通预测问题上应用广泛，并产生了诸多变形。如 Li 等^[12]提出改进的 ARIMA 预测方法用于预测热点地区乘客的时空变化，降低了司机寻客的时间和距离成本；韦凌翔等^[13]提出基于 RVM 和 ARIMA 的组合预测模型进行短时交通流预测，提高了预测的精度。

这些模型依据时间序列的移动平均性、自相关性，利用统计分析理论提取历史时间序列中的出行规律性和周期性变化特征，适用于较为稳定、线性的场景。然而在实际应用中，交通流数据往往包含复杂的非线性特征，传统的数学模型难以充分提取，因而在预测精度上有一定局限性。

2. 传统机器学习方法

在运用机器学习方法进行出租车需求预测方面，叶锋等^[14]借助回声状态网络、随机森林等机器学习算法预测出租车行程目的地和行驶时间，以辅助出租车资源调度；Liu 等^[15]提出了六个基于反向传播神经网络和极端梯度提升的不同信息模型来预测在线叫车需求，证明在机器学习中选取更多的影响因素可以使出租车叫车需求预测精度提高。

3. 深度学习方法

在运用深度学习方法进行出租车需求预测方面，Chen 等^[16]提出了一种基于图卷积网络、长期短时记忆和多任务学习的深度学习模型，并证明该模型能够在交通网络水平上有效地预测出租车的短时需求；Liu 等^[17]提出了一种上下文时空神经网络进行出租车 OD 需求预测，强调了该模型在预测出租车行程终点上的良好性能；Wang 等^[18]将出租车订单需求预测问题描述为一个起点一目的地矩阵预测问题，提出基于网格嵌入的多任务学习模型来解决这一问题。

机器学习、深度学习类方法主要依靠大量计算来分析出租车需求的变化规律，往往有着复杂的数学结构，且缺少交通系统相关理论的支撑，因而预测结果往往缺乏可解释性，因此有学者致力于对深度学习的预测结果给出可接受的解释。Kim 等^[19]结合机器学习中的线性回归模型和深度学习中的长短期记忆神经网络，开发了一个两阶段机器学习框架用于出租车短时需求预测，强调了这一模型框架的可解释性。

1.2.2 传统出租车、共享无人车调度方法

传统出租车调度和共享无人车的调度有一定的共通之处。首先体现在供需匹配方面，无论是传统出租车还是 SAV，在调度时都需要考虑出行需求和可用车辆的区域供需平衡；

其次是优先考虑乘客等待时间，平台在对车辆进行调度时，应尽量使乘客等待的时间最小。而传统出租车、网约车调度和 SAV 调度也存在一些区别。一方面是运力问题，人工驾驶情况下，整体的运力很大程度上取决于司机的工作意愿，因此平台有时会通过一些奖励制度，激励司机尽可能多地接单，从而维持运力；另一方面是成本问题，在对出租车、网约车进行调度时，平台要考虑到单个司机的空车损失，但对 SAV 调度时，平台不需要考虑人力成本，因此可以更加侧重于整体的收益情况，在进行区域再平衡时更有可能实现系统最优。

1. 巡游出租车、网约车调度方法

国内外关于出租车和网约车的订单调度及车队调度相关研究成果非常丰富，目前可以大致分为基于稳态的调度和实时优化的调度两类^[20]。无论是基于稳态的调度还是实时优化的调度，在建模时大多都是考虑调度成本、调度时间、乘客等待时间等因素，定义变量参数、构建约束条件，但对于调度模型的求解一直是研究的重点。因此出租车调度模型建立和对模型的求解工作往往共同开展。

基于稳态的调度方法中，通常将车辆、乘客等作为主要因素来对调度过程进行建模，然后根据优化算法求解最佳调度方案，部分研究在稳定状态中加入了路网等环境因素。张宗垚等^[21]对机场附近的出租车调度问题进行研究，通过分析出租车载客过程建立模型，使用自适应遗传算法对模型进行优化求解，合理安排上车点，从而使得总的乘车效率最高；Sadrani 等^[22]针对交通系统中不同尺寸的出租车辆组成的混合车队，考虑乘客平均等待时间建立调度模型，并提出了一种结合蒙特卡罗模拟框架的模拟退火算法来求解大规模实际问题；Kim 等^[23]通过将出租车调度问题转化为网络流问题，并利用最小花费最大流算法进行求解。

实时优化的调度方法是基于实时信息建立调度模型。Long 等^[24]将司机的期望收益与其搭载的乘客的等待时间及行程时间之和的比值定义为期望回报率（Expected Rate of Return, EROR），并根据不同 EROR 与实时交通信息分别制定调度策略；Yang 等^[25]首先对未来时间窗内的需求进行预测，并提前将出租车调往可能发生打车请求的区域以满足未来可能的需求，同时平衡区域间的车辆供给水平，以达到实时优化调度的目的。

与此同时，深度学习和强化学习在出租车辆调度问题建模和求解上得到越来越广泛的应用^[26]。Li 等^[27]使用多智能体强化学习方法，遵循点对点乘车共享问题的分布式性质，并采用均值场近似来简化局部交互；Jin 等^[28]使用分层强化学习解决车辆和订单匹配的问题；Zhe 等^[29]采用强化学习的方法，引入马尔可夫决策过程，将网约车订单调度

和司机调度连接在一起，并在滴滴打车平台上进行试验，使区域整体收益提高了 0.5%-5%；同为滴滴打车平台，Tang 等^[30]针对平台拼车订单调度问题，采用深度强化学习的方法，引入半马尔可夫决策，提出了具有新型分布式状态表示层的小脑值网络模型；Lin 等^[31]以最大化平台的总商品量为目标，对大规模在线拼车车队调度问题进行优化，构建了上下文多智能体强化学习框架。

对于巡游出租车和网约车调度，不仅需要考虑乘客等待时间、系统服务水平等因素，还应考虑司机个体收益。Yu 等^[32]对影响司机收入的各种因素进行了探索和研究，Chen 等^[33]在搭建派单调度框架时考虑司机收益确定调度路径，充分考虑司机个体，生成合理的调度方案。

2. 共享自动驾驶（电动）汽车调度方法

共享自动驾驶汽车又称为自动驾驶出租车，不需要考虑车辆个体的收益，使实现系统最优调度成为可能。国内共享无人车 app 打出了“不挑单，不拒载”的口号，表现出共享无人驾驶在乘客角度的便捷之处。

大多数私家车的使用时间不到 10%^[34]，因此汽车共享是一条通往可持续发展的明确道路——尤其是在汽车不需要司机移动的情况下。基于共享自动驾驶的理念，国外提出了自动按需移动（Automated Mobility-on-Demand, AMoD）系统的概念，并广泛研究其控制方法，可大致分为模型驱动类和数据驱动类。

模型驱动类包括排队理论方法^[35]、网络流方法^[36]、整数线性规划和模型预测控制方法^[37]以及基于模拟的方法^[38,39]等。Iglesias 等^[35]在一个封闭的多级 BCPM 网络中解决 AMoD 的再平衡问题，为 AMoD 系统的分析和控制提供了一个统一的框架；Zhang 等^[40]采用了排队理论建立了调度模型，Chua 等^[41]将排队模型运用到乘客上，同样完成了对车辆的调度；Federico 等^[42]指出网络流的一个关键限制是不能捕捉随机效应，特别是不能明确说明顾客到达过程的随机性，并尝试用随机排队网络解决自动按需移动系统和电力系统的相互作用问题。

数据驱动类，是指在没有预定模型的基础上，通过对历史数据的识别出一定的模式，以进行下一步决策。Shi 等^[43]采用分散学习-集中决策的整体框架，先通过强化学习的方法得到车辆的价值函数，然后通过传统的线性规划和网络流方法进行集中决策，生成调度方案；Kim 等^[44]采用深度学习的方法解决 AMoD 系统中闲置车辆迁移的问题，利用出租车大数据构建乘客需求预测模型，并基于预测的需求解决闲置车辆重新定位问题，

估计最优策略; Qian 等^[45]提出了影子价格深度强化学习(shadow-price deep reinforcement learning, shadow-price DRL), 结合了严格的模型和数据驱动的无模型 DRL 算法, 并证明了该方法的有效性。

3. 车队调度仿真模拟方法

为了评估调度模型的效果, 通常会在道路上进行仿真, 一般可以分为抽象道路网上的仿真和真实道路上的仿真^[46,47,48]。抽象道路网一般为网格状的虚构道路网络, 网络中的车流数据亦为模拟器生成; 而真实道路网则多为从开源地图上下载的城市真实路网, 在分析时也多结合出行大数据, 使仿真模拟的结果更为可靠。也有学者在仿真时引入更为贴近现实的交通流模型, 来模拟调度过程中路网上的拥堵现象^[49]。

1.2.3 研究进展小结

在出租车订单需求预测方面, 受到多种因素的影响, 出租车订单量的时间、空间分布虽然有一定规律, 但难以通过传统的建模方式表征。随着大数据、云计算以及人工智能等技术的发展, 许多学者开始尝试将深度学习技术应用于订单需求预测任务上。本文在进行订单预测时, 结合预测任务的特点, 选择合适的深度学习方法, 以提高预测精度, 更好地辅助出租车区域预调度任务的完成, 提高调度的准确性。

在出租车辆调度方面, 国内外关于出租车、网约车调度的研究已经比较成熟, 对自动按需移动系统控制方式的研究成果非常丰富, 但研究对象多为同质车队, 很少有研究以有人驾驶的出租车和共享无人车组成的混合车队为研究对象。本文考虑有人驾驶的出租车和共享无人车组成的混合车队, 研究系统最优条件下出租车辆区域再平衡调度方案的生成方法。并使用真实数据集计算生成调度方案, 检验调度模型的有效性。

1.3 主要研究内容

本研究在前人研究的基础上, 进一步探索出租车订单短时需求预测方法以及出租车队的区域调度方法, 研究中的车队为有人驾驶的出租车和共享无人车组成的混合车队, 主要包括以下内容:

1. 基于出租车订单数据的乘客短时需求预测

在开源网站中收集芝加哥和纽约的城市出租车订单数据集, 对原始数据进行预处理, 保留数据中的上下客时间、上下客区域、上下客经纬度等信息, 并采用出租车订单中原有的上下客区域信息对订单进行分类。采用长短时记忆神经网络和时间卷积网络两种深度学习方法, 并引入时序模式注意力机制对原本的深度学习方法进行改进, 构建短时客

流需求预测模型，对比评估模型的有效性和预测结果的准确性。

2. 建立区域供需再平衡调度模型

在对出租车订单需求进行预测的基础上，考虑有司机驾驶的出租车和共享无人车调度时的成本差异，设置调度时间限制，以系统调度成本最小化为目标，建立模型对两类出租车辆进行区域再平衡调度。在建立调度模型时将区域调度方案生成的任务分为两个阶段，第一阶段确定车辆的调出区域、调入区域以及对应的调度车辆数；第二阶段确定调度车辆编号以及车辆在路网上的调度起点和调度终点，同时也确定车辆的移动路径。

3. 模型实例应用与结果分析

为了验证模型的有效性，在真实的城市场景中应用模型计算区域调度方案，输出调度方案、调度成本、调度完成后各区域出租车辆数等，评估根据调度模型进行预调度后是否能有效满足区域出租车需求，实现出租车在区域间的供需再平衡；以及在分配调度任务时能否有效区分有人驾驶的出租车和共享无人车，尽可能减少司机参与调度的路程和行程时间。

1.4 论文组织结构

全文分为六个部分。第一部分为绪论，介绍了研究背景、国内外研究的进展以及本文的主要研究内容；第二部分为数据处理及时空特征分析，数据主要包括美国芝加哥市区的出租车订单数据、美国纽约曼哈顿出租车订单数据，对两地出租车订单的时间、空间特征进行了分析；第三部分为基于深度学习的出租车短时需求预测，运用深度学习中的长短时记忆神经网络和时间卷积网络，并引入时序模式注意力机制，构建出租车短时需求预测模型，并结合芝加哥数据集和纽约曼哈顿数据集测试模型的预测精度；第四部分建立出租车区域再平衡调度模型，考虑出租车订单需求的空间差异性，以系统最优为目标建立出租车区域再平衡调度模型，将车辆调度问题抽象为一个组合优化问题；第五部分为仿真模拟与结果分析，运用本文提出的出租车订单需求短时预测算法和区域再平衡调度算法，进行真实道路网上的仿真计算，分析算法的有效性；第六部分为总结与展望。

技术路线如图 1.1 所示。

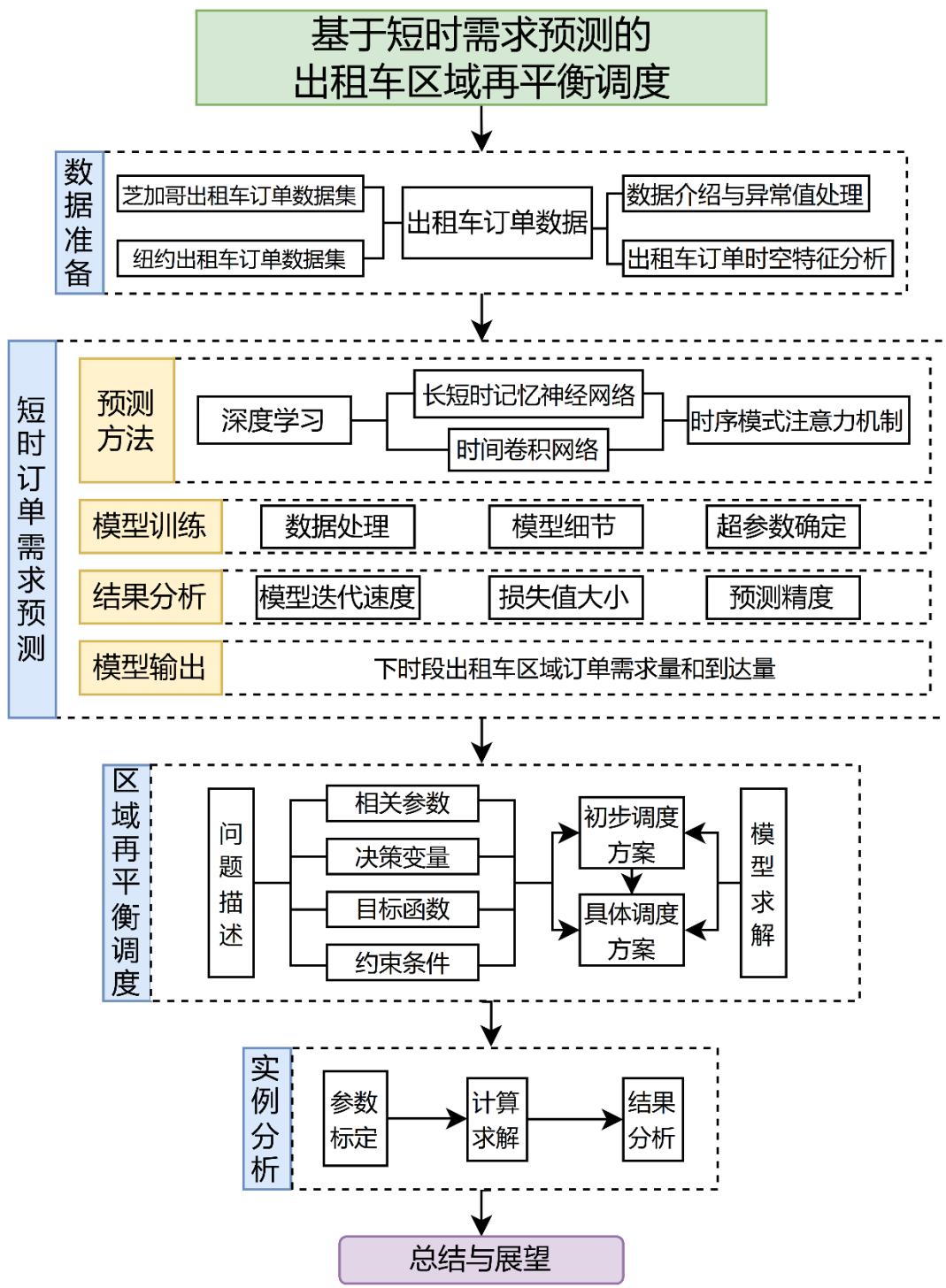


图 1.1 技术路线

第二章 数据处理及时空特征分析

数据是研究工作的基石，对数据进行预处理和时空特性分析有利于提高算法模型的准确度，并便于后续的仿真模拟与检验。在第一章中提到，本文所研究的车队调度任务的对象为有人驾驶的出租车和共享无人车组成的混合车队，而以当前共享无人车投放及试运营的规模，难以收集足够的样本来分析城市居民对这类服务的需求。共享无人车与巡游出租车、网约车的服务模式类似，都是为乘客提供点对点式的定制出行服务。因此在不考虑进一步刺激居民出行的情况下，可以用城市现有的出租车订单数据分析类出租车服务的需求，以供后续进行车队调度任务的研究。

本章将首先对本文选取的出租车行程数据集进行介绍，然后对数据集进行处理，并分析出租车行程的时空分布规律，为后续工作提供参考。

2.1 数据描述与预处理

2.1.1 芝加哥出租车订单数据

本文使用的芝加哥出租车订单行程数据为 Chicago Digital 提供的开源出租车数据集^[50]。为了保护隐私并允许进行汇总分析，数据集中的出租车 ID 与实际的出租车牌照编号不同，但一一对应。数据集中出租车订单行程开始和结束时间均四舍五入到最接近的 15 分钟。

在本研究中，选取时间范围为 2023 年 8 月 7 日至 2023 年 10 月 29 日的出租车订单行程数据，共 84 天。该数据集中，每条数据包含出租车 ID、出租车行程的上下车时间、上下客点的经纬度坐标、上下车点所在分区等 21 个字段。芝加哥出租车订单行程数据部分字段解释如表 2.1 所示。

表 2.1 芝加哥出租车行车数据部分字段信息

字段	类型	示例	备注
Taxi ID	Str	89fc6b1f0628f328cccd1021…	经过隐私处理
Trip Start Timestamp	Timestamp	11/30/2023 11:45:00 PM	以 15 分钟为
Trip End Timestamp	Timestamp	11/30/2023 11:30:00 PM	时间切片
Pickup Community Area	Int	12	
Dropoff Community Area	Int	70	
Pickup Centroid Latitude	Float	41.79409	不是真实上下
Pickup Centroid Longitude	Float	-87.6252	车点，经过聚
Dropoff Centroid Latitude	Float	41.98026	类处理
Dropoff Centroid Longitude	Float	-87.6333	

为了完成后续的模型训练与测试，需要检查数据是否存在异常。在该数据集中存在

两类异常数据，一类是部分字段存在缺失值的数据，另一类是订单的行程时间过短或过长的数据。由于该数据集中的上下客时间以 15 分钟为切片进行了模糊处理，因此无法判断哪些订单行程时间过短。考虑到订单涉及到的城区范围，将行程时间大于 2 小时的订单数据作为行程时间异常数据。在本研究中，采用直接删除异常数据的方式对数据进行数据清洗。

在原始数据中，行程起始时间在 2023 年 8 月 7 日至 2023 年 10 月 29 日之间的共有 1576848 条。经过数据清洗后，仍然保留的数据有 1413290 条。在数据集中出现的上下客区域这两列中，共出现的 77 个区域编号。数据集中出现的上客点共有 395 个，下客点共有 484 个，如图 2.1 所示。

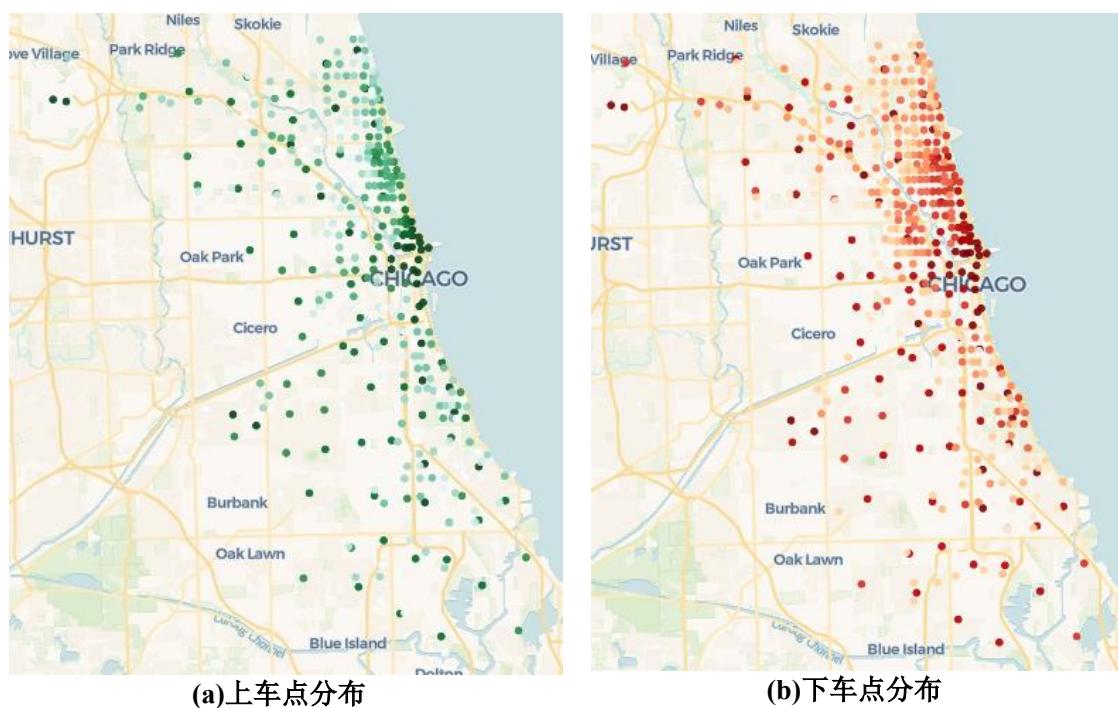


图 2.1 芝加哥数据集中上下客点分布情况

由于出租车服务具有门对门、点对点的特性，因此出租车行程的上下客点分布通常极为分散。而本文选取的 1413290 条出租车订单行程中，出现的上客点坐标仅有 395 个，下客点坐标仅有 484 个。在收集出租车订单数据时，可能对上下客点位按照某种规则进行的聚类处理，并非真实的上下客点位。

2.1.2 纽约出租车订单数据集

除芝加哥出租车数据集外，本文还使用了纽约市 TLC (New York City Taxi & Limousine Commission) 提供的开源出租车数据集^[51]。在本研究中，选取数据集中的黄色出租车订单数据，时间范围同样选取 2023 年 8 月 7 日至 2023 年 10 月 29 日，共 84 天。该数据集中，每条数据包含供应商编号、出租车行程的上下车时间、上下客区域、

行程距离等 19 个字段，部分字段解释如表 2.2 所示。在该数据集中未出现上下客点经纬度相关信息。

表 2.2 纽约出租车订单数据集部分字段信息

字段	类型	示例
VendorID	Int	1
tpep_pickup_datetime	Timestamp	2023/8/7 0:00:29
tpep_dropoff_datetime	Timestamp	2023/8/7 0:05:52
trip_distance	Float	1.27
PULocationID	Int	164
DOLocationID	Int	113
fare_amount	Float	8.6

纽约出租车订单数据集中数据条数极多，考虑到计算速度等因素，按照上下车点所在的区域，选则曼哈顿地区的出租车订单进行提取。提取完成后共保留数据 6857136 条。在纽约出租车订单数据集中，由于上下客的时间较为精确，因此将订单的行程时间小于 3 分钟和大于 2 小时的行程数据作为异常数据。数据清洗完成后保留数据 6602747 条。

结合出租车订单数据集中出现的上下车区域编号，在芝加哥和纽约的数据平台中进一步获取了区域区的划分情况^[51,52]，如图 2.2 所示。其中芝加哥共划分为 77 个区域，纽约曼哈顿共划分为 69 个区域。

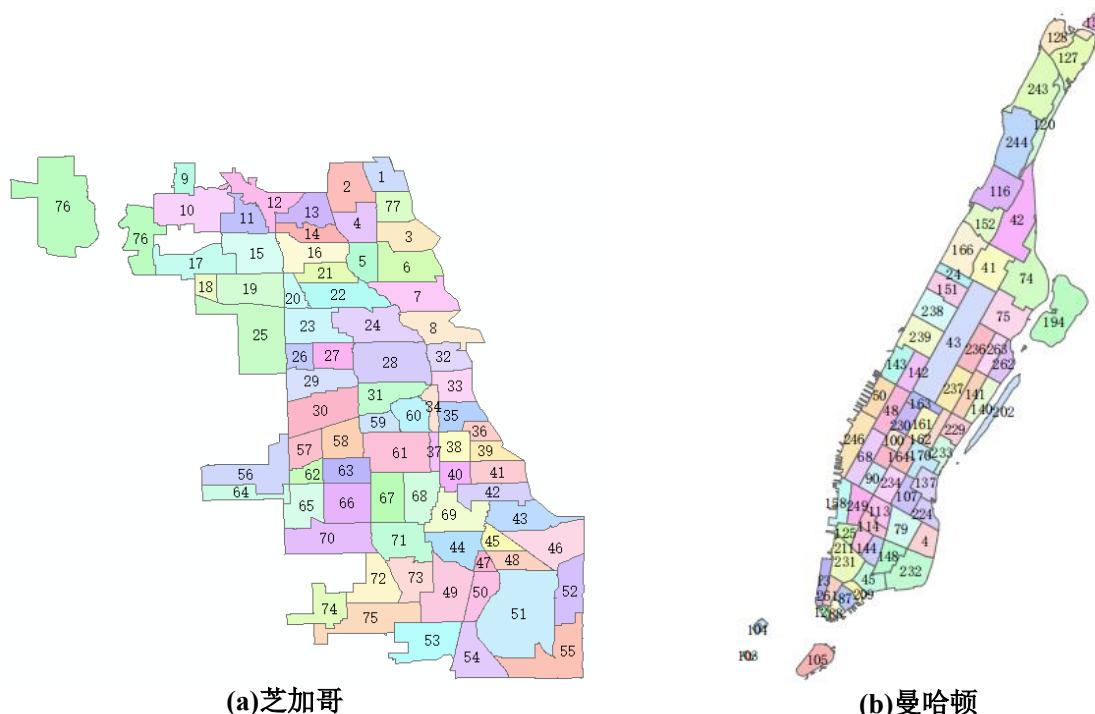


图 2.2 芝加哥与纽约曼哈顿分区图

2.2 数据时空特征分析

2.2.1 出租车订单时间特征分析

城市出租车出行需求在时间上通常表现出一定的规律性，对出租车订单数据集进行时间特征分析，分析出租车订单呈现出的统计规律，有利于辅助构建需求预测模型，提升模型预测的精度。

芝加哥市 2023 年 8 月 7 日至 2023 年 10 月 29 日每日产生的出租车订单量变化趋势如图 2.3 所示，纽约曼哈顿 2023 年 8 月 7 日至 2023 年 10 月 29 日每日产生的出租车订单量变化趋势如图 2.4 所示，图中横坐标标识出来的日期刻度均为周一。对比图 2.3 和图 2.4，可以看出芝加哥和纽约曼哈顿的每日订单量都呈现出周期性，以七天为一个周期，双休日出租车订单量明显低于工作日出租车订单量。

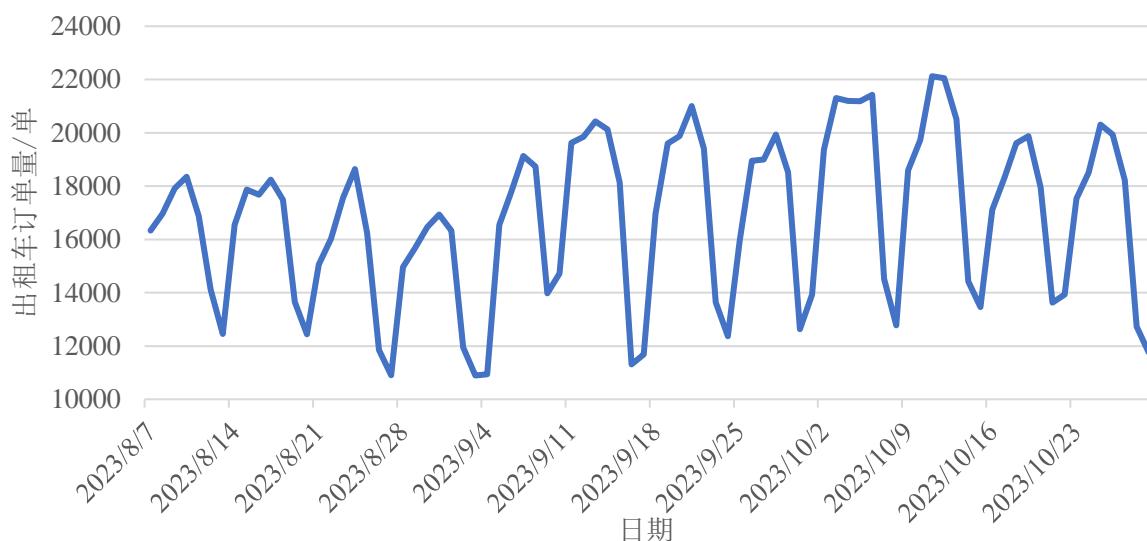


图 2.3 芝加哥每日出租车订单量

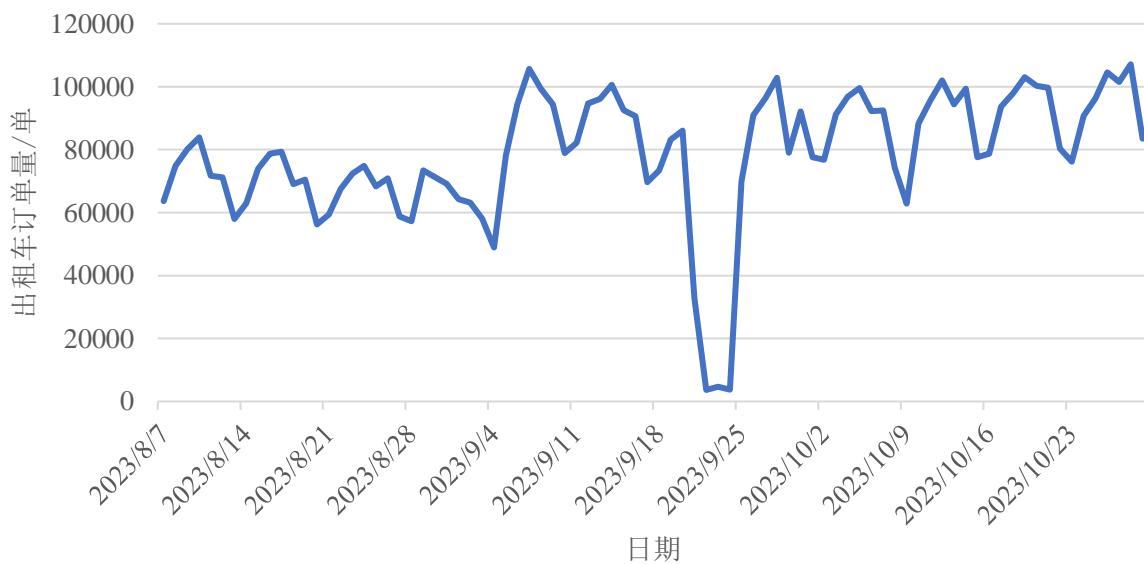


图 2.4 曼哈顿每日出租车订单量

在纽约曼哈顿数据集中，2023年09月22日、2023年09月23日和2023年09月24日三天产生的订单量分别为3662单、4631单和3758单，远远小于其他日期的订单量，可能由于这三天中纽约曼哈顿的出租车活动受到某些现实因素的影响，或者原始数据集在收集、统计的阶段出现了错误。为了防止这三天的异常数据对后续工作产生影响，选择将对应日期的订单数据进行剔除。剔除完成后，仍保留的纽约曼哈顿出租车订单数据有6590696条。

对异常数据进行剔除后，以七天为一个周期对出租车订单数据集进行进一步分析。由于芝加哥出租车订单数据集中，订单量的统计以15分钟为时间间隔，因此同样以15分钟为间隔对纽约曼哈顿的订单量进行分组。计算选取时间段内芝加哥和曼哈顿出租车订单量在一周内各天的平均值，绘制折线图如图2.5和图2.6所示。

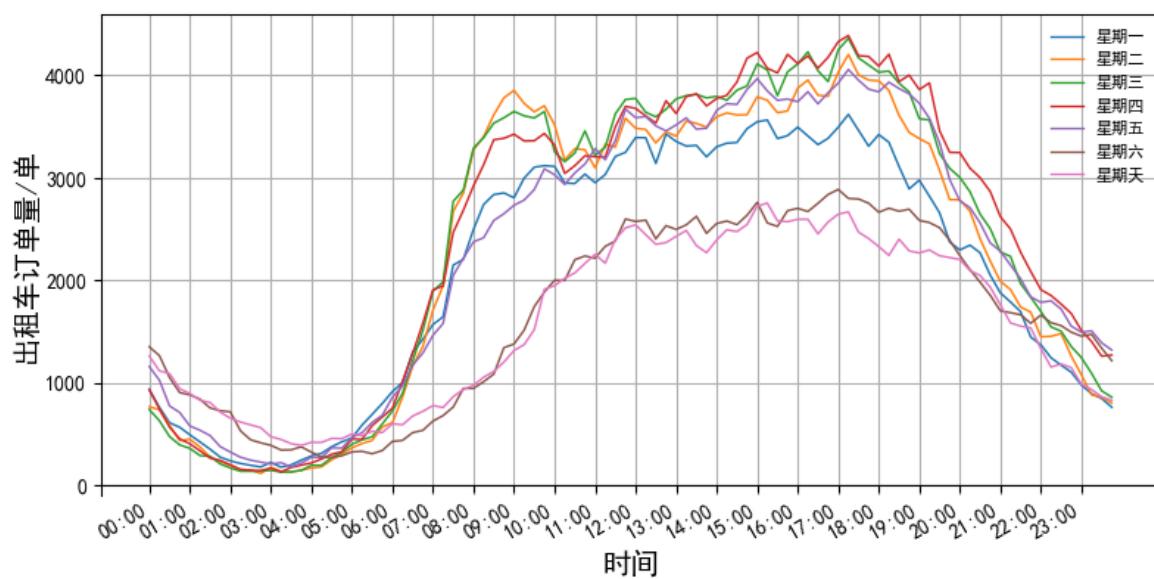


图 2.5 芝加哥各时刻平均出租车订单量

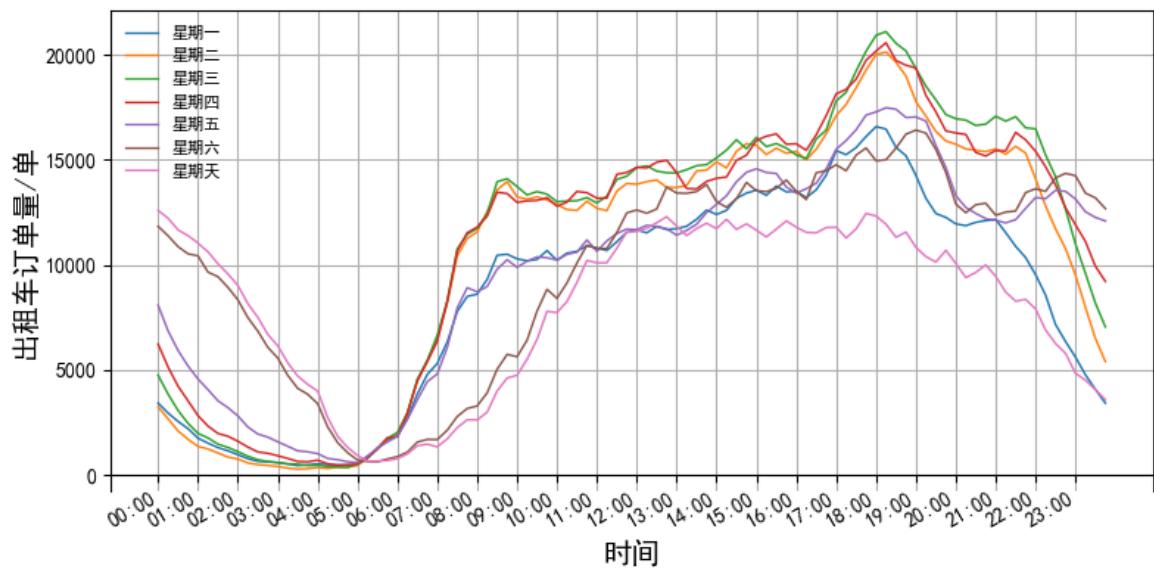


图 2.6 曼哈顿各时刻平均出租车订单量

从图中可以看出，芝加哥和纽约曼哈顿周一到周日各时刻的订单量变化趋势存在一定的相似性，同时也存在差异性。两地工作日的各时刻订单量变化趋势基本一致，订单量在上午 9:00 左右达到第一个峰值，随后订单量有所减少，在 10:00 之后各时段订单量持续增加，在 17:00-19:00 之间达到第二个峰值。其次双休日的出租车各时刻订单量变化趋势也基本一致，在 11:00 至 20:00 之间，订单呼叫量始终保持较高水平。与工作日相比，在白天的 7:00 至 10:00，双休日产生的出租车订单量相对较少，而在凌晨的 0:00 至 4:00 产生的订单量则略微高于工作日相应时段产生的订单量。将两地的工作日出租车订单量与双休日进行比较，芝加哥在双休日全天产生的出租车订单量明显低于工作日，而纽约曼哈顿在双休日全天订单量与工作日没有显著差距，两地工作日订单量随时间变化趋势均与双休日存在一定差异。

上述分析结论都是通过观察折线图的变化情况得出的，虽然符合生活常识，但是缺乏数据本身的支撑证明。因此，需要对数据进行检验。

对周一至周日全天各个时刻产生的订单量变化趋势进行比较。在本研究中，采用皮尔逊相关系数（Pearson Correlation Coefficient,）进行两两之间的相关性检验。皮尔逊相关系数有一个重要的数学特性是，两个变量的位置和尺度的变化并不会引起该系数的改变。也就是说，我们如果把 X 移动到 $a+bX$ 和把 Y 移动到 $c+dY$ （其中 a、b、c 和 d 是常数），并不会改变两个变量的相关系数。因此，皮尔逊相关系数在比较两组变量的变化趋势是否相关上比较有效。两个变量之间的皮尔逊相关系数定义为两个变量的协方差除以它们标准差的乘积：

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (2.1)$$

皮尔逊相关系数的变化范围为 [-1,1]，系数的值为 0 意味着两个变量之间没有线性关系，系数值越靠近 1 或 -1，两个变量之间的相关性越强。

对芝加哥出租车订单和纽约曼哈顿出租车订单的时间序列进行计算，得到皮尔逊相关系数如表 2.3 和表 2.4 所示。

整体来看，无论是芝加哥还是纽约曼哈顿，工作日和工作日之间的皮尔逊相关系数基本在 0.95 以上，出租车订单量随时间的变化趋势呈现出强相关性；而工作日和双休日之间的皮尔逊相关系数偏大，尤其是纽约曼哈顿出租车订单数据，皮尔逊相关系数的最小值达到了 0.570，相关性较弱。说明对于出租车订单数量在一天内的变化趋势，双休日与工作日的变化趋势差异较大，符合前文中的分析。

表 2.3 芝加哥出租车订单量皮尔逊相关系数

皮尔逊相关系数	星期一	星期二	星期三	星期四	星期五	星期六	星期日
星期一	1.000	0.990	0.993	0.985	0.982	0.889	0.911
星期二	0.990	1.000	0.996	0.985	0.970	0.863	0.876
星期三	0.993	0.996	1.000	0.994	0.982	0.885	0.898
星期四	0.985	0.985	0.994	1.000	0.991	0.914	0.920
星期五	0.982	0.970	0.982	0.991	1.000	0.949	0.953
星期六	0.889	0.863	0.885	0.914	0.949	1.000	0.982
星期日	0.911	0.876	0.898	0.920	0.953	0.982	1.000

表 2.4 曼哈顿出租车订单量皮尔逊相关系数

皮尔逊相关系数	星期一	星期二	星期三	星期四	星期五	星期六	星期日
星期一	1.000	0.987	0.975	0.971	0.922	0.708	0.633
星期二	0.987	1.000	0.996	0.991	0.943	0.711	0.570
星期三	0.975	0.996	1.000	0.995	0.958	0.744	0.575
星期四	0.971	0.991	0.995	1.000	0.974	0.770	0.594
星期五	0.922	0.943	0.958	0.974	1.000	0.861	0.644
星期六	0.708	0.711	0.744	0.770	0.861	1.000	0.855
星期日	0.633	0.570	0.575	0.594	0.644	0.855	1.000

2.2.2 出租车行程空间特征分析

首先统计 2023 年 8 月 7 日至 2023 年 10 月 29 日期间，芝加哥市各个人口普查区以及纽约曼哈顿各分区的出租车订单总量，如图 2.7、图 2.8 所示。

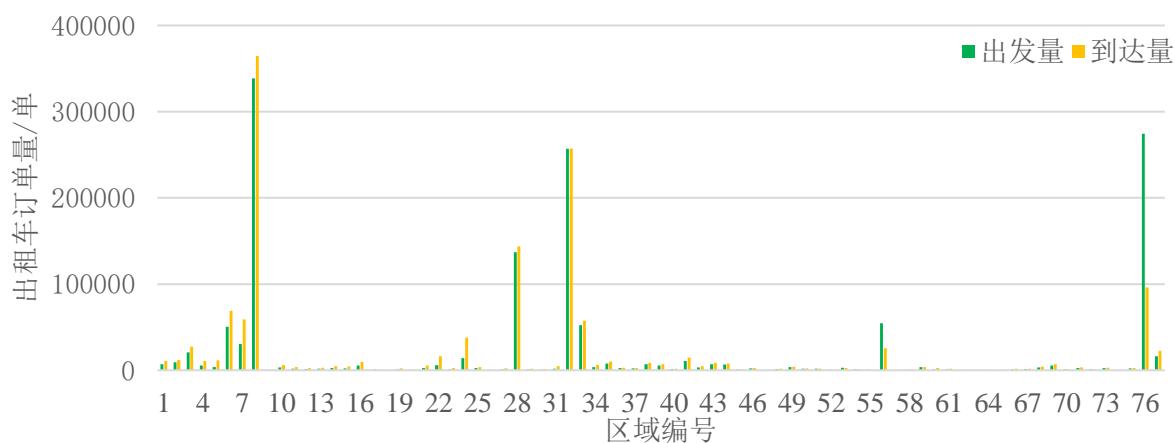


图 2.7 芝加哥各分区出发量和到达量

由柱状图可以看出，芝加哥各区域间的订单量存在明显差异。在本研究选取的 84 天中，绝大多数区域累计出发的出租车订单量以及到达的订单量均在 10000 单以下，而出租车订单量最多的区域，累计出发的订单量达到了 338478 单，到达量达到了 364797 单。

大多数分区的订单出发量和到达量较为接近，但也存在部分区域订单出发量和到达量存在显著差异，例如编号为 76 的区域，累计订单出发量为 274556 单，而到达量仅为 96202 单，出发量超过了到达量的 2.85 倍。

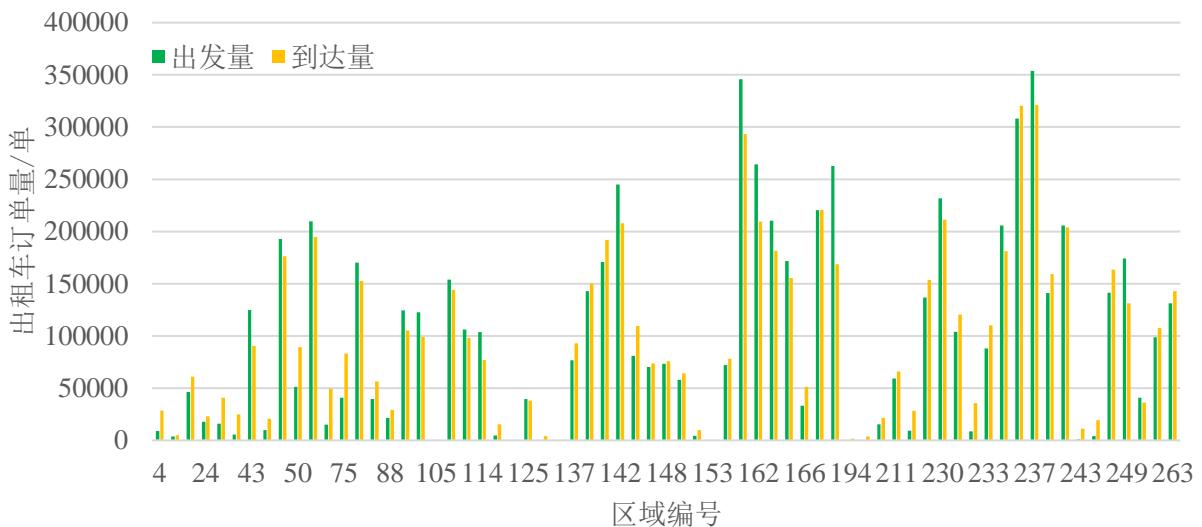


图 2.8 曼哈顿各分区出发量和到达量

相比于芝加哥，纽约曼哈顿各分区的订单量较为均匀，但最大值和最小值仍存在很大差距。在统计的时间段内，有 3 个分区订单的出发量和到达量均超过了 25 万，但也存在部分分区的订单出发量和到达量均不到 1000。与芝加哥出租车订单类似，纽约曼哈顿各分区的出租车出发量和到达量存在明显差异。

为了更直观的显示订单在空间上的分布情况，将出租车订单量在分区图上进行可视化。由于各分区之间的出租车订单量差异过大，为了更好地展示订单量分级情况，采用自然间断点分级法（Jenks）将订单量划分为 10 个类别，在此基础上展示各分区订单量差异，得到图 2.9 和图 2.10。

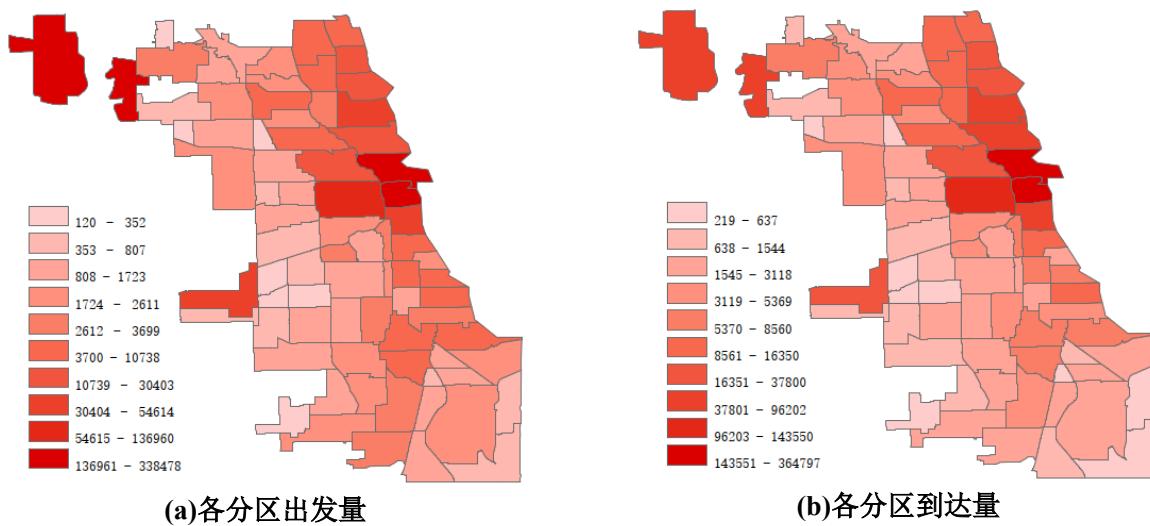


图 2.9 芝加哥市各分区订单出发量和到达量

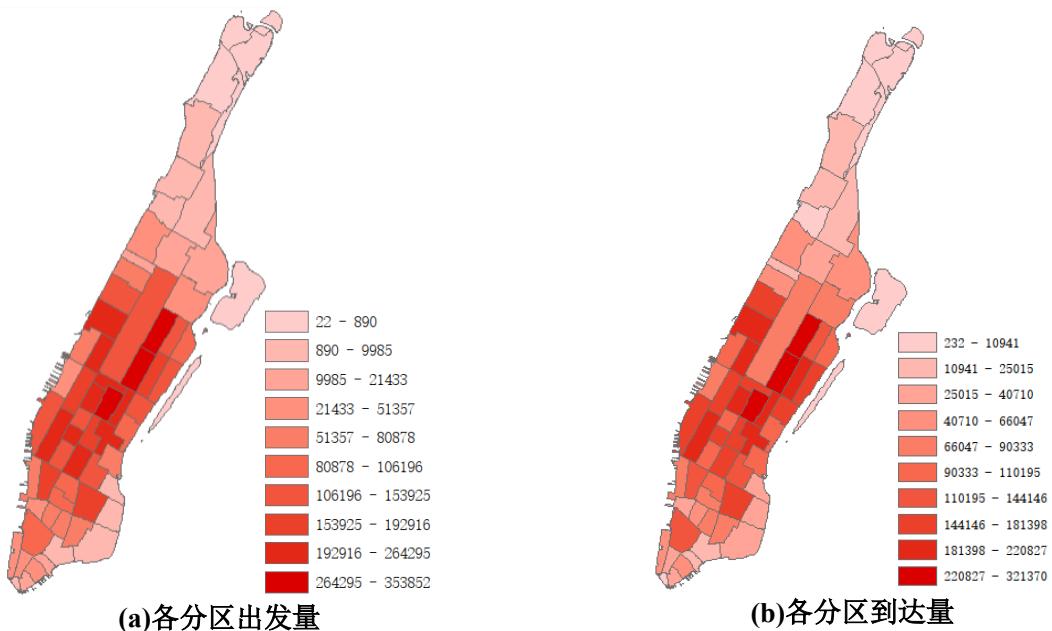


图 2.10 曼哈顿各分区订单出发量和到达量

由图中可以看出，无论是芝加哥还是纽约曼哈顿地区，不同分区的订单量都存在明显的差异。其中芝加哥的出租车订单主要集中在东部中心区域以及西北角，曼哈顿的出租车订单则主要集中于南部。无论是芝加哥还是纽约曼哈顿，上车量和下车量在各区域的整体分布都较为相似，订单数始发量较多的区域，在该区域内结束的订单量也相应较多；但也有部分区域出发量和到达量分布存在差异。

出发量和到达量的分布差异也会导致区域出租车供需失衡，因此需要对出租车进行及时调度，以满足区域需求。

2.2.3 出租车行程时空特征分析

在 2.2.1 节出租车订单量时间特征分析和 2.2.2 节订单量空间特征分析中，纽约曼哈顿和芝加哥城市范围内的出租车订单量都表现出一定的时间规律性，同时部分行政区域间订单量差异较大。由于本文研究的是出租车区域供需再平衡问题，因此需要对城市中不同片区出租车订单量随时间变化特征进行分析。

为了更好地分析各区域出租车订单量随时间的变化趋势，探究其规律性，以纽约曼哈顿为例，从所有区域中选取 10 个分区，绘制 2023 年 10 月 11 日不同分区订单出发量变化趋势图如图 2.11 所示；从所有日期中选取 2023 年 10 月 9 日至 2023 年 10 月 15 日连续一周，绘制 48 号区域和 236 号区域不同日期订单出发量变化趋势图如图 2.12 所示。

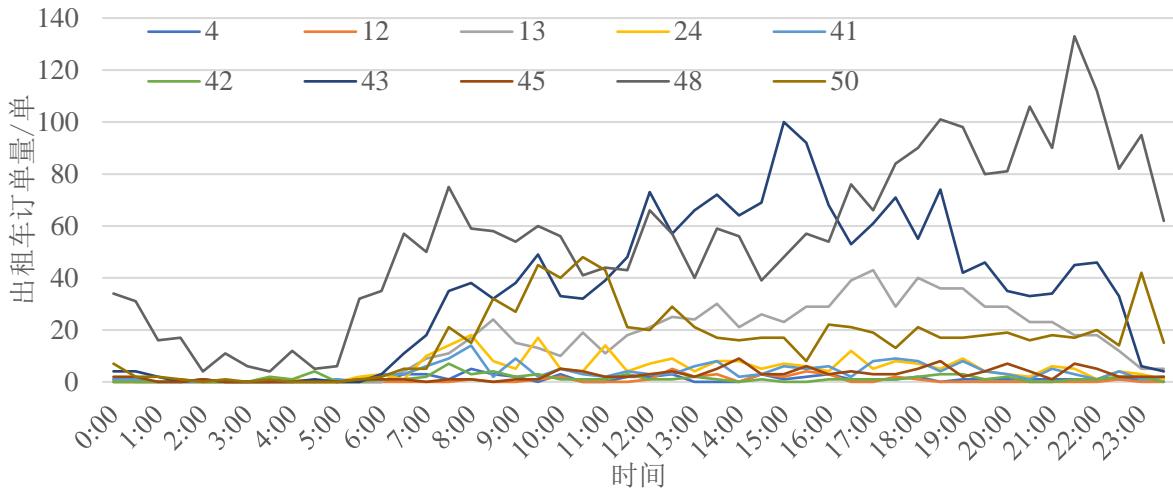


图 2.11 曼哈顿单日不同区域订单出发量变化趋势

由图 2.11 可以看出，对于订单量整体较少的区域，一天内的订单量随时间变化没有表现出明显的规律。而对于订单量较大的区域，如图 2.13 中的 48 号分区，订单量出现早晚两个高峰，与人们的日常出行习惯较为一致。

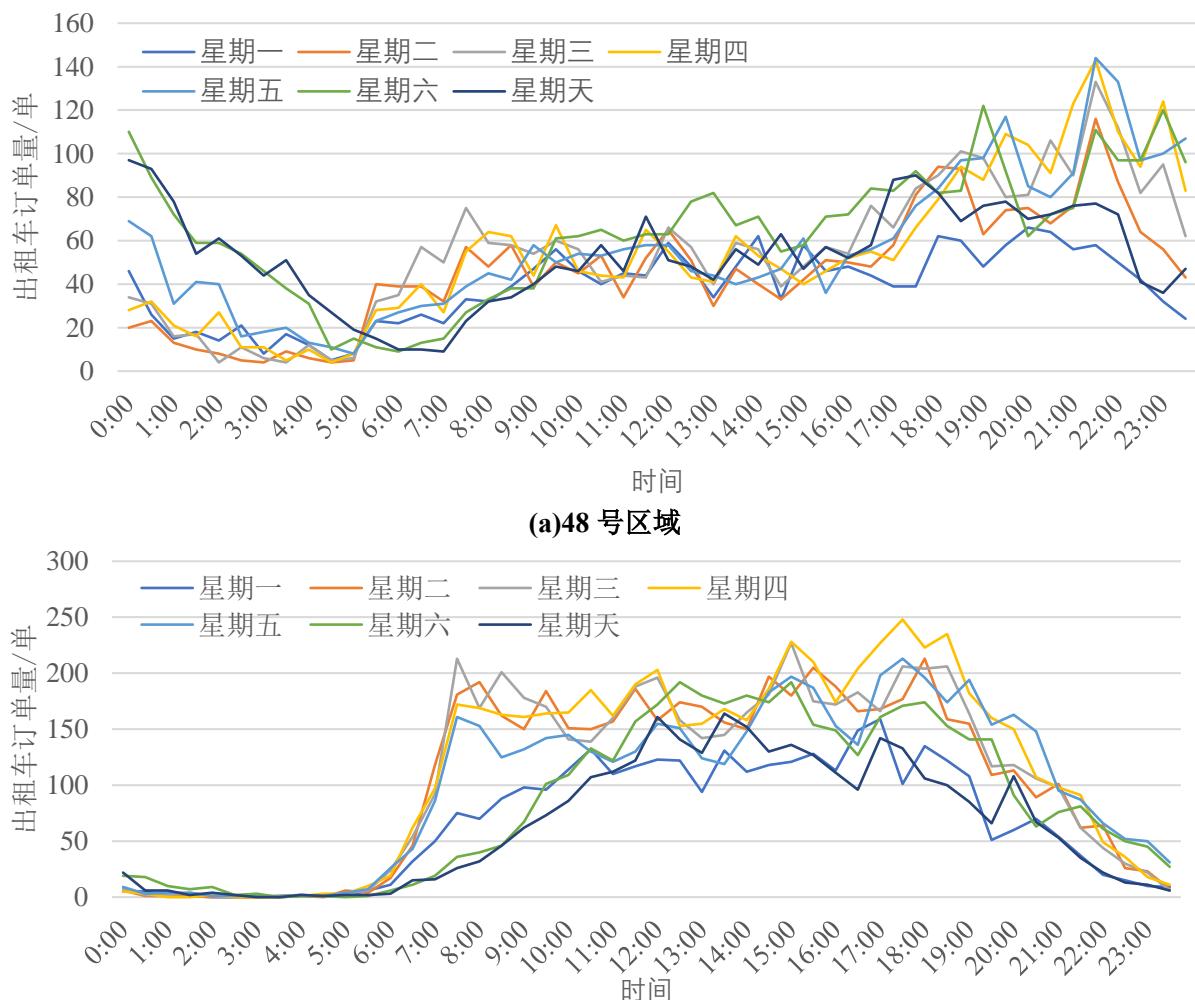


图 2.12 一周内订单出发量变化趋势

在图 2.12 中，虽然选取的均为 2023 年 10 月 9 日至 10 月 15 日的订单数据，但 48 号区域和 236 号区域的订单量走势呈现出明显的差异性。并且对于同一区域，工作日和双休日的订单量走势也有所不同。

本研究需要在出租车订单量预测的基础上进行出租车的区域再平衡调度，而预测的前提是各个区域的订单量变化趋势具有规律性。由于研究对象是出租车订单量随时间的变化趋势，因此在这部分内容中，可以将数据作为时间序列进行研究。

自相关检验是一种辅助判断数据是否具有某种规律性或变化趋势的重要统计分析方法，用于检测时间序列数据中的自相关性，即一个时间序列中的值与其自身滞后值之间的相关性，对于时间序列分析、预测模型的构建和验证非常重要。若通过自相关检验发现时间序列是白噪声，则说明序列是完全随机的，过去的行为对未来的发展没有丝毫影响，故而没有必要再深入分析。

白噪声是一种理想的随机信号，其特点是：

(1) 均值恒定，即

$$E(\varepsilon_t) = \mu \quad (2.2)$$

(2) 方差恒定，即

$$\text{Var}(\varepsilon_t) = \sigma^2 \quad (2.3)$$

(3) 无自相关性，即

$$\text{Cov}(\varepsilon_t, \varepsilon_s) = 0, \quad t \neq s \quad (2.4)$$

在本研究中采用 Ljung-Box 检验判断序列是否为白噪声，其计算步骤如下：

(1) 计算自相关系数 r_k

对于一个时间序列 x_t ，自相关系数 r_k 可以表示为：

$$r_k = \frac{\sum_{t=k+1}^n (x_t - \bar{x})(x_{t-k} - \bar{x})}{\sum_{t=1}^n (x_t - \bar{x})^2} \quad (2.5)$$

其中， \bar{x} 是 x 序列的均值， n 是序列的长度。

(2) 计算检验统计量

Ljung-Box 检验统计量 Q 计算公式为：

$$Q = n(n+2) \sum_{k=1}^m \frac{r_k^2}{n-k} \quad (2.6)$$

其中， m 是检验的滞后期数。

(3) 计算 p 值

统计量 Q 近似服从一个自由度为 m 的分布。根据 Q 和 χ^2 分布，可以计算出 p 值：

$$p\text{-value} = P(\chi_m^2 \geq Q) \quad (2.7)$$

其中， $P(\chi_m^2 \geq Q)$ 是 χ^2 分布的右尾概率。

Ljung-Box 检验的原假设认为数据是独立的。如果 p 值小于显著性水平(例如 0.05)，则拒绝原假设，认为数据具有显著的自相关性。

对一天内各分区订单量随时间的变化趋势是否呈现出规律性进行 Ljung-Box 检验，计算纽约曼哈顿和芝加哥各分区内出租车订单量时间序列的 p 值，如图 2.13 和图 2.14 所示。

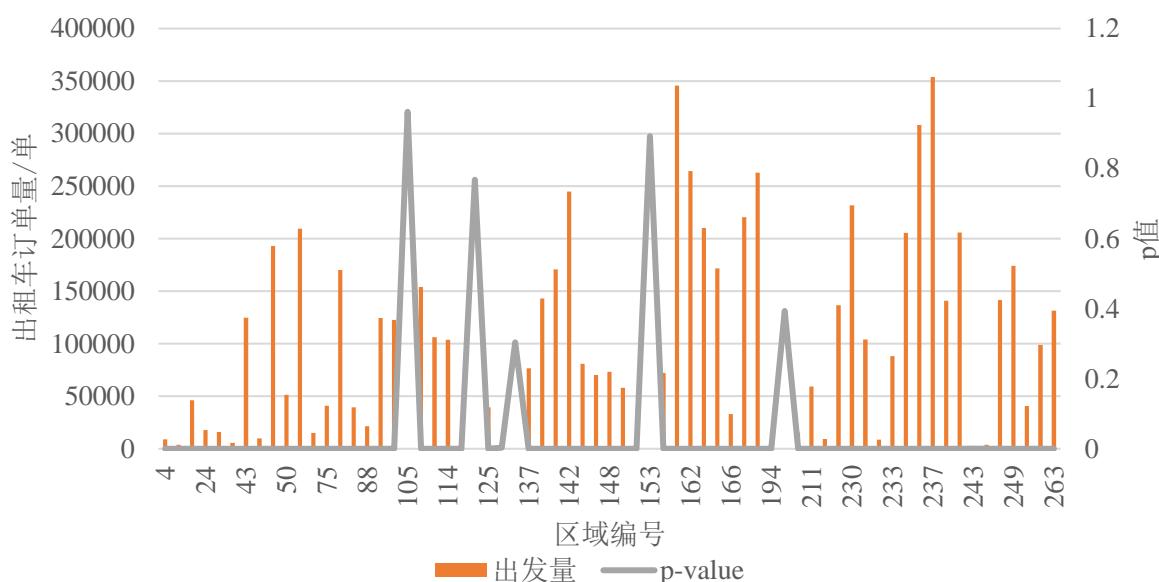


图 2.13 纽约曼哈顿序列 p 值与订单出发量

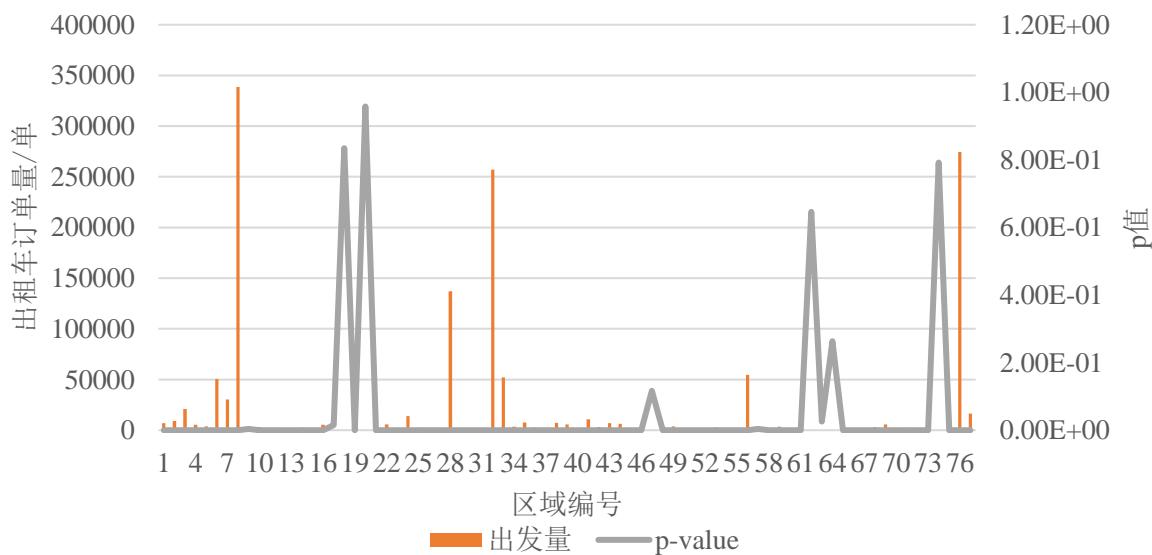


图 2.14 芝加哥序列 p 值与订单出发量

对于纽约曼哈顿和芝加哥的多数区域， p 值接近于 0，区域内出租车订单量序列有明显的、确定的变化趋势，不属于白噪声。而少数区域，如纽约曼哈顿地区编号为 105、120、128、153、202 的区域，以及芝加哥地区编号为 18、20、47、62、64、74 的区域，由于分区内产生的出租车订单量过小，订单量随时间变化区域不明显，在进行订单预测时也会增加预测难度。

总体来看，纽约曼哈顿和芝加哥整体范围内订单量随时间变化的规律性强，可以建立预测模型进行订单量预测。

2.3 本章小结

本章首先对研究选取的两个出租车订单数据集——芝加哥出租车订单数据集和纽约黄色出租车订单数据集进行介绍、词条筛选和异常值值处理。接着对订单数据的时空特征进行分析。在时间上，分析了纽约曼哈顿和芝加哥两地一天内不同时间段以及一周内不同订单需求量的变化趋势；在空间上，分析了两地不同行政区域内的出租车订单量差异；最后分析了同一时间不同区域、同一区域不同日期订单量变化趋势以及早晚高峰区域间出租车出发量和到达量的差异，并通过自相关性分析，证明各个分区订单量随时间变化具有一定的规律性，为后续工作提供参考。

第三章 基于深度学习的短时订单需求预测

在进行出租车区域调度时，往往存在调度滞后等问题，进而导致整个服务系统的效益和乘客满意度降低。如果能预测出租车的出发地需求和乘客的目的地，可以更有效地进行出租车的预调度^[53]，因此建立高效、可靠的出租车短时需求预测算法十分必要。本研究基于深度学习，结合历史出租车行程数据，对未来短时间内的出租车订单需求量进行预测，并检验预测结果的可靠性。

3.1 出租车订单量预测问题描述

本研究关注的是区域内出租车的订单需求预测，因此首先将需要预测的总范围划分为 n 个交通小区，将交通小区的质心定义为承接出租车出发和到达的虚拟站点 s ，虚拟站点集合为 $S = \{s_1, s_2, \dots, s_n\}$ 。同时，本文研究的是一段时间内的出租车订单需求，而不是某个瞬间产生的出租车订单量，因此以一定的时间间隔 τ 将时间进行切片，时间切片集合被定义为 $T = \{t_1, t_2, \dots, t_m\}$ 。

使用元组 (t, s) 表示出租车上客点和下客点的时空坐标，其中 t 代表时间， s 代表虚拟站点。使用 (p, q) 表示上/下车点对，其中 p 表示上车点， q 表示下车点，所有的上/下车点对集合 $P = (p_1, q_1), (p_2, q_2), \dots, (p_n, q_n)$ 。某一个目标行程的上车点定义为 $p = (t_p, s_p)$ ，下车点定义为 $p = (t_q, s_q)$ 。给定所有的上下车对 P ， $x_{s,t}^p$ 和 $x_{s,t}^q$ 分别表示虚拟站点 s 在时间间隔 t 内的出发车辆数量和到达车辆数，定义如下：

$$x_{s,t}^p = |(p, d) \in P : s_p \in S \wedge t_p \in T| \quad (3.1)$$

$$x_{s,t}^q = |(p, d) \in P : s_q \in S \wedge t_q \in T| \quad (3.2)$$

在本文中，对出租车订单需求的预测，是用已经观测到的 H 个时间切片内的出租车历史行程，计算历史出发车辆数量 $x_{s,t}^p$ 和到达车辆数 $x_{s,t}^q$ ，预测下一个时间切片的 $x_{s,t}^p$ 和 $x_{s,t}^q$ 。因此，可以将本文中的预测问题看作时间序列的预测，如图 3.1 所示。

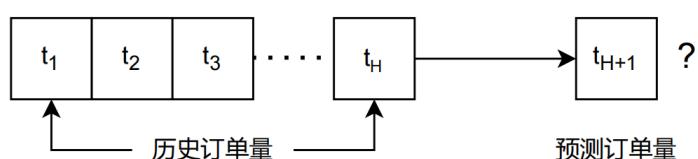


图 3.1 出租车订单预测示意图

3.2 基于深度学习的时间序列预测方法

3.2.1 长短时记忆神经网络

长短期记忆神经网络(Long Short-Term Memory, LSTM)^[54]是具有长期记忆能力的一种循环神经网络(Recurrent Neural Network, RNN), 由于具有时间递归能力, 经常被用在具有时间序列特性的数据和场景中。其网络结构由一个或多个具有可遗忘和记忆功能的单元组成, 原理如图 3.2 所示^[55]。

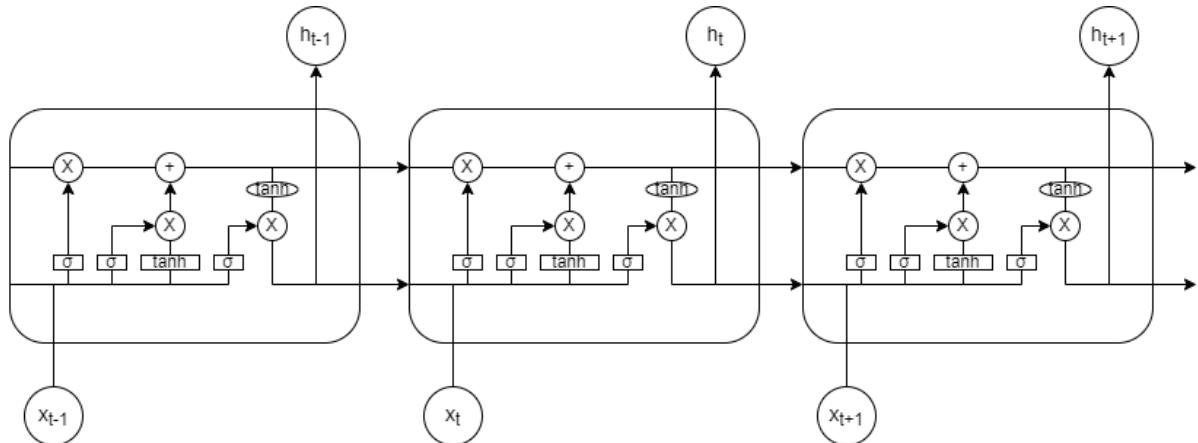


图 3.2 LSTM 原理示意图

LSTM 是一种循环神经网络, 因此需要经过重复的循环模块进行计算, 每个模块中有四个神经网络层。在图 3.2 中, 最上方的横线为单元状态 (cell state), 在 LSTM 中, 可以通过门向单元中添加信息。每个模块中有三个门, 每个门由 sigmoid 神经网络层和点乘法运算组成。sigmoid 层输出介于 0 和 1 之间, 输出为 0, 表示完全丢弃序列信息; 输出为 1, 表示完全保留序列信息。

(1) 遗忘门 f_t 通过观察 t 时刻的输入 x_t 和 $t-1$ 时刻的隐藏层状态 h_{t-1} , 决定 $t-1$ 时刻的单元状态 C_{t-1} 有多少信息可以被保留。因此, 遗忘门反映了前一时刻的记忆单元对当前时刻的重要程度。

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3.3)$$

(2) 输入门 i_t 有两个输入, 一个输入通过观察 t 时刻的输入 x_t 和 $t-1$ 时刻的隐藏层状态 h_{t-1} , 决定向信息流添加多少 t 时刻的输入 x_t ; 另一个输入通过 tanh 层产生一个新的候选单元状态 \tilde{C}_t 。输入门 i_t 决定了当前时刻的输入 x_t 中有多少信息要添加到新的候选单元状态 \tilde{C}_t 中。

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3.4)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3.5)$$

根据遗忘门和输入门的计算结果，对单元状态 C_t 进行更新：

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \quad (3.6)$$

(3) 输出门 o_t 决定当前时刻的单元状态 C_t 中有多少信息要传递到隐藏层状态 h_t 中：

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3.7)$$

根据输出门计算结果更新隐藏层：

$$h_t = o_t \circ \tanh(C_t) \quad (3.8)$$

经过三个门的作用，最终得到输出层的计算为：

$$y_t = \sigma(W_y \cdot h_t) \quad (3.9)$$

以上就是 LSTM 网络的计算过程。其中， W 为各状态的权重矩阵； b 为各个门的偏置项； $[h_{t-1}, x_t]$ 表示将 h_{t-1} 与 x_t 拼接作为整体向量；函数 $\tanh(\bullet)$ 和 $\sigma(\bullet)$ 为两种激活函数；符号“ \circ ”表示哈德玛内积。

通过引入新的单元状态和门控机制，LSTM 可以根据需要选择性地保留或遗忘历史序列信息，同时也可以选择性地添加或过滤输入信息。这种设计有效地解决了长序列信息处理中可能出现的梯度消失或爆炸问题。

3.2.2 时间卷积网络

时间卷积网络（Temporal Convolutional Network, TCN）^[56]是 2018 年提出的一种可以处理时间序列数据的卷积网络模型。与 LSTM 不同，时间卷积网络并不是循环神经网络，而是一种卷积神经网络（Convolutional Neural Network, CNN），不需要递归计算，因此训练速度更快，并且其神经元能够对一定范围内的周边单元产生响应。卷积神经网络并不是由多个相同的单元组成，而是由卷积层、池化层和全连接层组成。卷积层用于提取局部特征，池化层用于降低特征维度和位置敏感性，全连接层用于进行分类或回归等任务。相比于一般的 CNN 模型，TCN 通过以下几方面的改进来提高对时间序列的建模能力：

1. 扩展感受野（receptive field）。TCN 通过堆叠多个卷积层来逐步增加感受野，使得每个神经元能够捕捉到更长期的依赖关系，有助于处理长序列信息，从而提高了模型对时间依赖关系的建模能力。
2. 因果卷积（causal convolution）。在因果卷积中，卷积操作只能访问当前即之前时刻的信息，对未来的信息则无法查询，避免了信息泄漏，确保了时间的先后顺序。
3. 残差连接（skip connect）。残差连接是将网络下层的特性信息拿到高层一起作为输

入，被证明能够减小深度网络的梯度消失/爆炸问题的影响。残差连接：

$$o = \text{Activation}(x + \mathcal{F}(x)) \quad (3.10)$$

4. 感受野可调节。TCN 中的卷积核大小和步幅通常是可调节的，这使得模型能够根据具体任务的需求来灵活地调整感受野大小，从而适应不同长度的输入序列。在对长序列信息进行训练时，需要增大网络深度或卷积核尺寸。为了避免增大网络深度、增加参数数量，TCN 通常使用空洞卷积（dilated convolution）来增加感受野。对于输入的时间序列 x 、输入序列中的元素 s 以及滤波器 f ，空洞卷积操作 F 定义为：

$$F(s) = (X *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot X_{s-d \cdot i} \quad (3.11)$$

其中， d 为扩张率（dilation rate）， k 为滤波器大小， $s-d \cdot i$ 代表过去的方向。通过间隔采样，空洞卷积在增大网络感受野的同时又不会增加网络的计算量。

通过引入因果卷积、残差连接和可调节的感受野等，TCN 模型在处理时间序列数据时表现出优秀的建模能力和性能。

3.2.3 注意力机制

人类在观察和思考时，由于注意力有限，通常在快速了解全景的基础上寻找感兴趣的部分重点关注，从而产生注意力焦点，并剔除无效或干扰信息。通过这种方法，人们可以从海量信息中快速提取出高价值的信息，同时选择性忽略其他不重要的信息。注意力机制^[57]正是模仿了人类这一行为方式。

在深度学习中，注意力机制对输入序列的不同部分赋予不同权重，通过“打分”的方式使模型能更有效地把握住重要信息，从而提高训练效率。打分过程如图 3.3 所示。

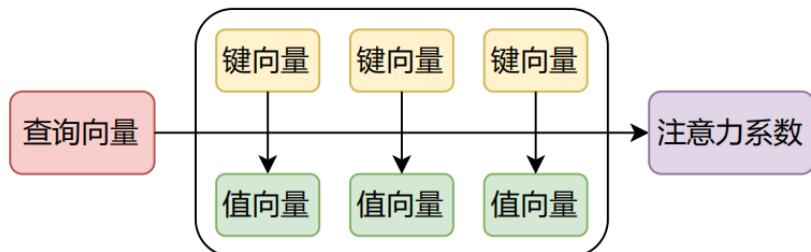


图 3.3 注意力机制

对于查询向量（Query），计算其与键向量（Key）的注意力分布，得到注意力系数后将其附加到对应的值向量（Value）上。在这个过程中，模型不再同等地考虑所有输入信息，而是依据权重，即重要程度选取部分信息参与模型训练。

注意力机制可以分为软注意力机制（Soft attention）和硬注意力机制（Hard attention）^[58]。软注意力机制计算出注意力权重后，将权重与相应的输入特征进行加权求和，得到

加权平均值作为输出，能够动态地为所有输入分配不同的注意力权重，通常应用于连续的输入序列，如文本序列、时间序列等。硬注意力机制中的注意力权重不是连续的，而是根据一些条件或策略从输入中选择特定元素输出，而不是对所有输入进行加权求和，通常应用于离散的输入，如图像、音频等。

由于本研究中主要对时间序列数据进行学习，因此采用软注意力机制对 LSTM 和 TCN 模型进行优化，具体计算过程如下：

(1) 给定一组输入信息 $X = \{x_1, x_2, \dots, x_n\}$ 和查询向量 q ，首先将输入 X 映射成键向量 $K = \{k_1, k_2, \dots, k_n\}$ 和值向量 $V = \{v_1, v_2, \dots, v_n\}$ 。

(2) 使用打分机制衡量 q 和 k 之间的相似性或相关性，常见的打分机制有^[59]：
加性模型：

$$s(q, k_i) = v_a^T \tanh(W_q q + W_k k_i) \quad (3.12)$$

点积模型：

$$s(q, k_i) = q^T k_i \quad (3.13)$$

双线性模型：

$$s(q, k_i) = q^T W_a k_i \quad (3.14)$$

缩放点积模型：

$$s(q, k_i) = \frac{q^T k_i}{\sqrt{d_k}} \quad (3.15)$$

其中 v 、 W_q 、 W_k 、 W_a 是网络的权重矩阵或向量， d_k 是键向量的维度。

(2) 引入 soft max 函数，对计算得到的分数进行数值转换。对于每个值向量 v_i ，其注意力分布：

$$\alpha_i = \text{soft max}(s(q, k_i)) = \frac{\exp(s(q, k_i))}{\sum_{j=1}^n s(q, k_j)} \quad (3.16)$$

(3) 将值向量 v_i 与注意力分布 α_i 进行加权求和，得到输出向量：

$$V = \sum_{i=1}^n \alpha_i v_i \quad (3.17)$$

随着注意力机制在时序预测方面的研究和应用逐渐深入，Shih 等^[60]提出了时序模式注意力机制（Temporal Pattern Attention mechanism, TPA），使用 CNN 滤波器提取输入信息中的定长时序模式，公式如下：

$$H_{i,j}^C = \sum_{l=1}^{\omega} H_{i,(t-\omega-l+1)} \times C_{j,T-\omega+l} \quad (3.18)$$

其中, H_i 表示第 i 个行向量, C_j 表示第 j 个卷积核, $H_{i,j}^C$ 表示第 i 个行向量和第 j 个卷积核作用得到的键向量值。T 代表注意力机所覆盖的范围, 通常令 $T=\omega$ 。

在本研究中尝试在 LSTM 和 TCN 模型中加入时序模式注意力机制对模型进行改进。

3.3 出租车短时订单需求预测模型构建

在本研究中, 共搭建四个模型对出租车订单进行预测, 分别为长短时记忆神经网络 (LSTM)、时间卷积网络 (TCN)、引入时序模式注意力机制的长短时记忆神经网络 (TPA-LSTM) 以及引入时序模式注意力机制的时间卷积网络 (TPA-TCN)。在 TPA-LSTM 以及 TPA-TCN 中, 将 TPA 层添加在 LSTM 层/TCN 层之后, 对 LSTM 层/TCN 层的输出向量计算权重, 然后再将预测结果输出。在本节中介绍预测任务中的数据处理方法以及模型细节。

3.3.1 数据处理方法

在本研究中, 出租车订单需求预测的目标是通过给定的先前观测到的各区域出租车的出发量和到达量, 来预测未来各区域的出租车的出发量和到达量。在划分区域时, 以原本数据集中所标注的订单上下客点所在分区进行划分, 再进行下一步的预测任务。

1. 时间切片

通常情况下, 进行时间切片时选择的时间间隔越小, 时间切片内的订单量越少, 订单量变化趋势的不确定性也会相应增大, 与此同时预测模型在进行时间序列预测时精度及效率会有所降低。根据其他学者的研究, 若乘客提前 30 分钟提交出行订单, 匹配调度后出租车的空载率会有较为明显的降低^[61]。考虑到车辆在区域间调度所需时间以及短时预测的精度, 本文在进行出租车短时需求预测及结果评估时, 分别以 15 分钟、30 分钟为时间间隔, 预测各区域的出租车出行需求量以及出租车到达数量。在进行数据处理时, 分别以 15 分钟、30 分钟为间隔对数据进行重分组。

2. 特征数量

将不同区域产生的订单量作为数据集中的不同特征输入到预测模型中进行训练和预测。因此处理完成后时序数据的维度应为(时间切片数, 分区数)。因此在进行训练时, 完整的数据长度为时间切片数, 特征数量为分区数。这样处理可以分别对每个分区的订单量变化趋势进行拟合, 同时也能考虑到其他分区订单量变化对单个分区订单量的影响。

3. 时间步长 (sequence length)

时间步长的选择直接影响预测模型的性能和行为。步长的大小取决于数据的特性、问题的复杂度以及应用场景。时间步长较小时，模型训练具有以下特点：①较高的分辨率。较小的步长意味着模型对时间序列的变化更敏感，可以更精细地捕捉数据中的波动和短期趋势。②较高的模型复杂性。小步长可能需要更复杂的模型来捕捉高频率的变化。③计算量较大，需要处理更多的时间步。与之相对的，较大的时间步长会使模型的分辨率降低，并使模型更关注较长时间范围内的趋势，可能更好地捕捉长期的周期性变化，同时模型简化，计算量减少。

考虑在本研究中，需要对出租车短时需求进行预测以辅助车辆实现区域间预调度，对模型预测精度、拟合程度要求高，综合考虑模型精度以及模型的训练速度和成本，选择 16 个时间步长作为时间窗口大小，16 个时间步内的订单量数据即为预测任务所需的特征变量 X ；将每个时间窗口的下一个时间步对应的订单量作为预测的目标值 y 。

4. 数据标准化

为了减小模型偏差、提高收敛速度，采用 Z-score 标准化方法，通过去除均值和按单位方差缩放来实现上车量和下车量的归一化。经过归一化后的上下车量：

$$x^* = \frac{x - \bar{x}}{\sigma} \quad (3.19)$$

其中 x 代表上/下车量， $\bar{x} = \text{mean}(x)$ 是 x 的均值， $\sigma = \text{std}(x)$ 为 x 的标准差。

5. 训练集测试集划分

根据 2.2 节的分析，出租车订单数据在时间上具有周期性，为了更好地分析预测结果，在本研究中不随机抽取训练集和测试集，而是将全部数据按照时间序列，选择前 80% 的数据作为训练集，后 20% 的数据作为测试集。

3.3.2 模型细节与超参数确定

1. 优化器

自适应优化器是在训练神经网络等机器学习模型时使用的优化算法，它动态地调整每个参数的学习率，以便更有效地优化模型的参数。在本研究中，选择 Adam 优化器作为预测模型的优化器。

2. 正则化技术

在深度模型的训练中，有时会出现过拟合 (Overfitting) 情况，即模型训练过程中，对训练集的模拟和学习过度贴合，但将模型运用至测试样本时拟合效果变差，即泛化能

力 (generalization ability) 不强。为了对过拟合状况进行改善，在实际的模型训练中常采用正则化技术。“dropout”便是一种正则化技术，在每个训练步骤中以概率 p 随机将部分神经元的输出置为零，从而减少神经元之间的依赖性，特别是在训练数据不足的情况下。本研究中在 LSTM 中的隐藏层和 TCN 中的全连接层加入 dropout 正则化技术以减少过拟合风险，保证模型的泛化能力。

Dropout 率指的是在每次训练迭代中随机丢弃的神经元的比例，过高或过低的 dropout 率都可能导致训练误差和测试误差之间的差距增大。通常情况下，提高 dropout 率可以降低模型复杂度，防止模型记忆训练样本的特定噪声或细节；但也会降低训练速度，导致模型需要更多的训练迭代才能收敛。dropout 率过高还会降低模型的拟合程度，使预测精度偏低。

3. 损失函数 (loss function)

深度学习模型训练的过程就是不断调整模型参数以最小化损失函数的值过程。通过反向传播算法，计算损失函数相对于模型参数的梯度，从而更新参数，使得损失函数的值逐步减小。在模型训练完成后，通过计算测试集的损失值，可以衡量模型预测结果与实际结果之间的差距，量化模型的预测效果。在进行模型训练和预测结果评价时，采用均方误差 (MSE)、均方根误差 (RMSE)、平均绝对误差 (MAE) 和 Huber 损失四种损失函数。假设预测值 $\hat{y} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n\}$ ，真实值 $y = \{y_1, y_2, \dots, y_n\}$ ，则：

平均绝对误差：

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (3.20)$$

均方误差：

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.21)$$

均方根误差：

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3.22)$$

Huber 损失：

$$L_\delta = \begin{cases} \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2, & |y_i - \hat{y}_i| \leq \delta \\ \delta |y_i - \hat{y}_i| - \frac{1}{2} \delta^2, & |y_i - \hat{y}_i| > \delta \end{cases} \quad (3.23)$$

在梯度下降时 MSE 较 MAE 更为准确，而在异常值出现时 MAE 较 MSE 更加鲁棒。RMSE 是将 MSE 的结果进行平方根，以便结果与原始数据的量纲一致，与 MSE 相比，RMSE 对异常值更加敏感。Huber 损失结合了 MSE 和 MAE 的优势，下降速度介于 MAE 与 MSE 之间，可以减少离群值对损失函数的影响，且能有效避免梯度爆炸的问题。超参数 δ 对 Huber 损失的影响主要体现在其对于大误差的惩罚程度上。较大的 δ 会导致在大误差时的惩罚更小，使得损失函数更加接近于 MSE；而较小的 δ 则会导致在大误差时的惩罚更大，使得损失函数更加接近于绝对误差 MAE。

4. 隐藏层数量 (hidden size)

隐藏层数量是 LSTM 模型结构的一个关键超参数，直接影响着模型的复杂性和预测能力。增加隐藏层数量可以增加模型复杂程度，使其能够更好地学习复杂序列，有助于捕捉更细粒度的序列变化特征，提高模型对数据的拟合能力，同时也会增大计算量，增加训练时间。对于复杂的时间序列数据，较少的隐藏层可能无法捕捉到数据中的复杂模式，但在数据量较小的情况下有助于防止出现过拟合。

5. 通道数量 (num channels)

在 TCN 中，卷积操作是主要的时间建模方式，每个卷积层中的通道数量决定了每层卷积的输出特征图的深度。增加通道数量可以增加模型的表示能力和学习的特征的复杂度，从而提高模型的拟合能力。但如果将通道数量设置得太大，则会导致模型过拟合，并增加计算成本。

6. 卷积核大小 (kernel size)

在 TCN 中，卷积核的大小决定了模型在时间维度上捕捉信息的范围。较大的卷积核可以使模型考虑更长的时间跨度，从而更好地捕捉时间序列中的长期依赖关系。但同时也会增加模型的参数量和计算复杂度。通常，卷积核大小的选择取决于数据的时间尺度和序列中重要模式的持续时间。

7. 学习率 (learning rate)

学习率决定了模型在每一轮训练中参数更新的幅度，会直接影响模型的收敛速度、收敛质量、稳定性和泛化能力。如果学习率过大，可能会导致参数更新跨度大，使得模型在训练过程中波动幅度过大，甚至无法收敛。相反，学习率过小会导致收敛速度降低。与此同时，过大的学习率可能导致模型无法稳定地收敛到最优解，而过小的学习率可能使模型陷入局部最优解。在本研究中，使用 Adam 自适应学习率算法来帮助调节学习率，提高模型训练的效率和性能。

8. 批次大小 (batch size)

在深度学习中, batch size 指的是在训练过程中一次批量处理的样本数量。通常情况下, 数据集会被分成若干个批次, 每个批次包含一定数量的样本, 这个数量就是 batch size。较大的 batch size 通常能提高模型计算和收敛的速度, 因为在每次迭代中, 计算梯度和更新模型参数时使用的样本数量更多, 能使模型更加注重全局而非局部。但 batch size 增大的同时也会导致内存需求和计算压力增大。而较小的 batch size 能使模型更频繁地更新参数, 探索数据的变化模式。但 batch size 太小会导致模型容易受样本噪声的影响而变得不稳定。

9. 迭代次数 (epoch)

Epoch 指的是在整个训练集上迭代的次数, 1 个 epoch 完成时, 模型已经使用了整个训练数据集进行了一次训练。一般情况下, 随着迭代次数的增大, 模型对于训练集的拟合程度会逐渐提高。但受限于模型的表达能力以及原始数据的复杂程度, 在达到一定的迭代次数后, 模型产生的损失值会逐渐趋于稳定, 不再继续增强对训练集的拟合程度。如果 epoch 过小, 可能导致模型尚未收敛, 对数据的拟合能力没有完全表达, 导致模型欠拟合 (underfitting); 而有些模型本身比较复杂, 提高迭代次数后仍能继续提高对训练集的拟合程度, 但在测试集上的泛化能力降低, 导致模型过拟合。

在确定部分超参数的值时, 本文采用网格搜索法, 以确保模型的整体性能和预测精度达到最优。最终确定 LSTM 模型中, 隐藏层数量为 20; TCN 模型中, 通道数量为 64, 卷积核大小为 3。对于所有预测模型, dropout rate 确定为 0.5, 学习率为 0.001, Huber 损失中的超参数 δ 为 0.1, batch size 确定为 336, epoch 确定为 25。

3.4 出租车订单量预测结果与模型评价

本节主要列出本研究选择的四个预测模型基于四种评价指标的实验结果, 分析各个模型在两个订单数据集上各个模型的表现情况。

3.4.1 纽约曼哈顿出租车订单量预测

图 3.4 展示了以 30 分钟为间隔进行时间切片时, 纽约曼哈顿的出租车数据训练的过程中各个模型的损失值随训练回合数的变化情况。由于在使用数据训练模型之前, 对数据进行了标准化, 因此得到的损失值较小。

从损失值的下降速度看, 相对而言, LSTM 和 TPA-LSTM 模型的损失值下降速度较慢, 说明模型的拟合能力较弱且训练效率较低; TCN 模型的损失值下降速度最快, 说明

模型的拟合能力较强；相较于 TCN 模型，TPA-TCN 的损失值下降速度较慢，且在训练后期出现了损失值的小幅波动。并且在引入 TPA 注意力机制后，LSTM 和 TCN 模型的损失值下降速度都有所减缓。分析原因，引入注意力机制后，在训练的早期阶段，注意力机制所提供的权重等参数还没有被充分调整和利用，因此模型可能需要更多的训练时间才能收敛到最优解。

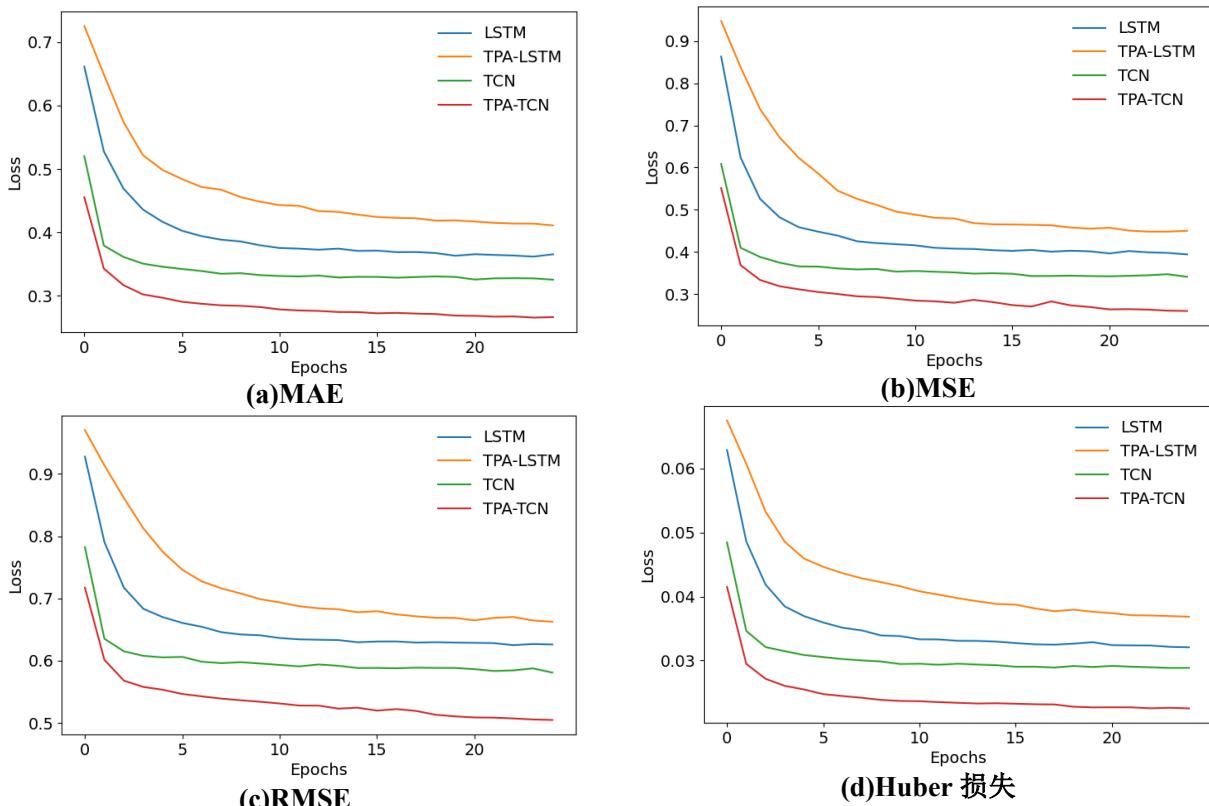


图 3.4 纽约曼哈顿训练集损失值变化

从损失值的大小来看，TPA-LSTM 模型的各项损失值最高，TPA-TCN 模型的损失值则最低。说明经过在测试集上的训练，TPA-LSTM 模型对数据的拟合程度较低，TPA-TCN 模型对训练集数据的拟合程度较高。值得注意的是，在引入 TPA 后，LSTM 模型的损失值提高，模型的拟合能力下降；而 TCN 模型的损失值降低，模型的拟合能力提升。由于出租车订单量变化规律较为复杂，搭建的 LSTM 模型本身复杂度低，不足以捕捉数据中的复杂关系，引入注意力机制难以提高模型的性能。而 TCN 模型本身在捕捉长距离序列特征方面有所欠缺，引入 TPA 后模型的全局分析能力得到提升，模型拟合程度提高。

从损失函数的选择来看，选择 Huber 损失进行训练时，各个模型的损失值下降速度最快，因此选择 Huber 损失训练模型，更有利于模型快速收敛、在有限的迭代次数内充分拟合。

模型训练迭代 25 次后，其他三个模型的各项指标基本趋于稳定，但 TPA-TCN 模型的损失值还在缓慢降低，尤其是用 RMSE 进行训练时，说明 TPA-TCN 模型在训练集上还未充分拟合。但在对深度学习模型进行训练时，如果只观察训练集上的损失值，可能会忽略过拟合的风险。因此为了进一步分析模型在训练的过程中是否存在过拟合情况，选择模型训练最后一次迭代的损失值与测试集预测结果的损失值进行比较，得到表 3.1。

表 3.1 纽约曼哈顿训练集与测试集损失值

模型	LSTM		TPA-LSTM		TCN		TPA-TCN	
	损失	训练集	测试集	训练集	测试集	训练集	测试集	训练集
MAE	0.3653	0.3717	0.4109	0.4466	0.3252	0.3354	0.2662	0.3212
MSE	0.3939	0.4099	0.4501	0.4991	0.3408	0.3763	0.2599	0.3528
RMSE	0.6261	0.6324	0.6626	0.6899	0.5809	0.5966	0.5048	0.5791
Huber	0.0321	0.0328	0.0368	0.0395	0.0289	0.0289	0.0225	0.0281

由表 3.1 可以看出，在模型迭代 25 次的前提下，所有模型的测试集损失值均高于训练集最后一次迭代产生的损失值。其中，LSTM 模型测试集的损失值与训练集最为接近，并且从图 3.4 中可以看出，LSTM 模型的损失值已经趋于稳定，说明在四个预测模型中，LSTM 模型过拟合的风险最低；其次为 TCN 模型，在用 Huber 损失进行训练时，训练集和测试集的损失值几乎没有产生差距。而引入 TPA 的两个预测模型训练集和测试集的损失值差距较大，其中 TPA-TCN 模型的训练集和测试集损失值差距最大，测试集各项损失值均比训练集损失值高 10% 以上。并且在图 3.4 中，TPA-TCN 模型的损失值随着迭代次数增加仍在缓慢降低，如果继续增加迭代次数并不能使模型训练更为充分，只会增加过拟合的风险。虽然在训练时已经引入了 dropout 正则化技术，可以一定程度上防止模型过拟合，提高泛化能力，但由于引入了 TPA 注意力机制后，模型的复杂程度进一步提高，对训练集的拟合程度提升的同时也会导致模型的泛化能力降低。

对比各个预测模型在测试集上的损失值可以发现，虽然存在过拟合风险，但 TPA-TCN 模型在测试集上的损失值仍然最低，模型的拟合能力最佳。相比于 TCN 模型，引入 TPA 注意力机制后，模型的 MAE 损失降低了 4.23%，MSE 损失降低了 6.25%，RMSE 损失降低了 2.93%，Huber 损失降低了 2.77%。

选择曼哈顿编号为 79 的区域，输出用 Huber 损失训练的模型在测试集上的一部分预测结果，得到图 3.5。整体来看，TPA-TCN 模型得到的预测值与真实值曲线最为贴合，尤其是对于图 3.5 中出现的两个峰值，TPA-TCN 模型的预测结果与真实值最为接近。而 LSTM 以及 TPA-LSTM 模型对于该段序列中出现的前两个峰值的预测与真实值差别较

大，对数据变化趋势的捕捉能力较差。

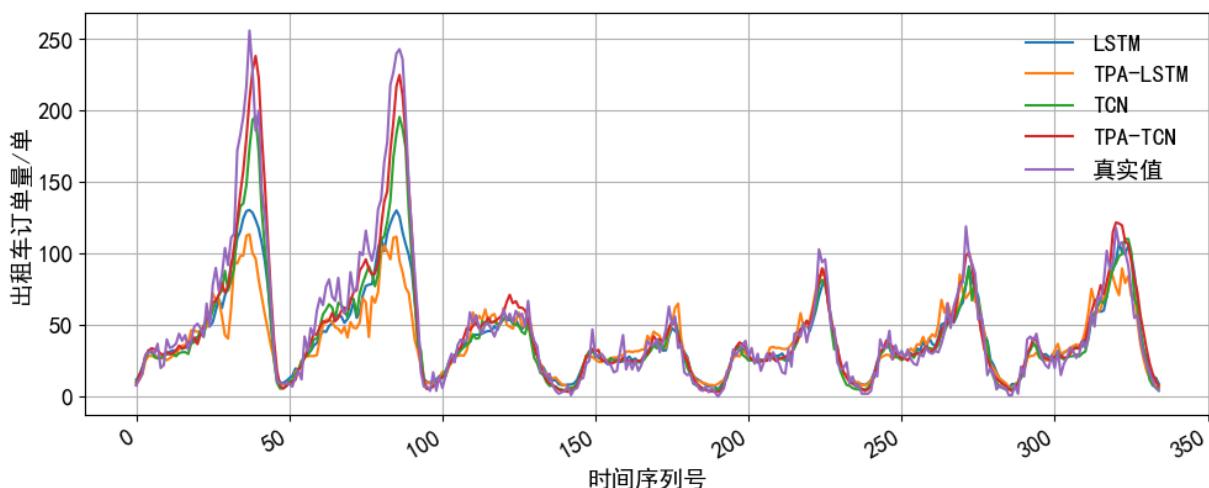


图 3.5 纽约曼哈顿测试集真实值与预测值

在第二章数据分析中，纽约曼哈顿不同分区产生的订单量存在一定差异。绘制各分区订单量以及预测结果在测试集上的损失值如图 3.6 所示。

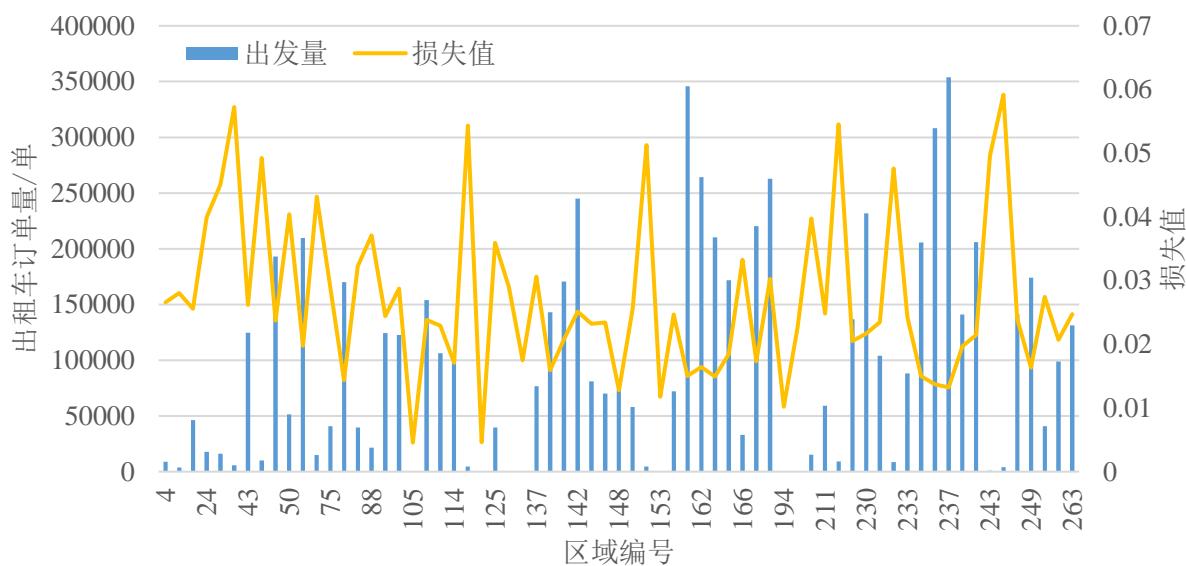


图 3.6 纽约曼哈顿各分区订单出发量与预测结果损失值

在图 3.6 中，订单量较大的分区订单量预测结果的损失值偏低，而订单量较小的分区订单量预测结果的损失值往往偏高。由于部分区域出租车订单量较少，订单量随时间变化的规律具有更大的随机性，因此在采用深度学习方法进行训练时容易出现过拟合情况，进而导致预测模型在测试集上表现不佳，预测精度偏低。

以 15 分钟为间隔进行时间切片的情况下，纽约曼哈顿的出租车数据训练过程中各项损失值变化趋势与以 30 分钟为间隔进行切片时类似，而训练集和测试集上的损失值大小相对来说略高，如表 3.2、图 3.7 所示。

表 3.2 纽约曼哈顿训练集与测试集损失值(时隙为 15 分钟)

模型	LSTM		TPA-LSTM		TCN		TPA-TCN	
	训练集	测试集	训练集	测试集	训练集	测试集	训练集	测试集
MAE	0.3775	0.3884	0.3930	0.4219	0.3475	0.3725	0.2967	0.3470
MSE	0.4380	0.4564	0.4531	0.5143	0.3928	0.4428	0.3150	0.4168
RMSE	0.6587	0.6655	0.6736	0.7084	0.6229	0.6417	0.5546	0.6353
Huber	0.0337	0.0348	0.0350	0.0382	0.0309	0.0322	0.0257	0.0308

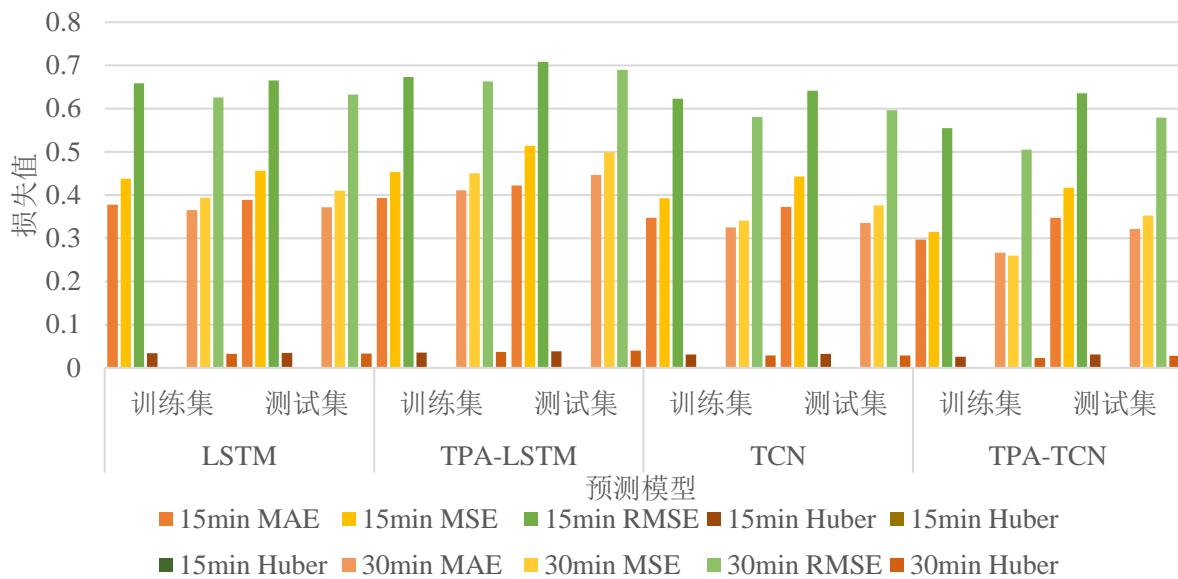


图 3.7 15min 间隙与 30 分钟间隙损失值对比

与时间间隔为 30min 的时间切片预测结果相比,以 15min 为间隔进行时间切片时,预测模型的模型在测试集上的 MAE、MSE、RMSE 和 Huber 损失分别增大了约 3.72%、11.73%、6.12% 和 5.18%。

订单预测结果与真实值的比较如图 3.8 所示。

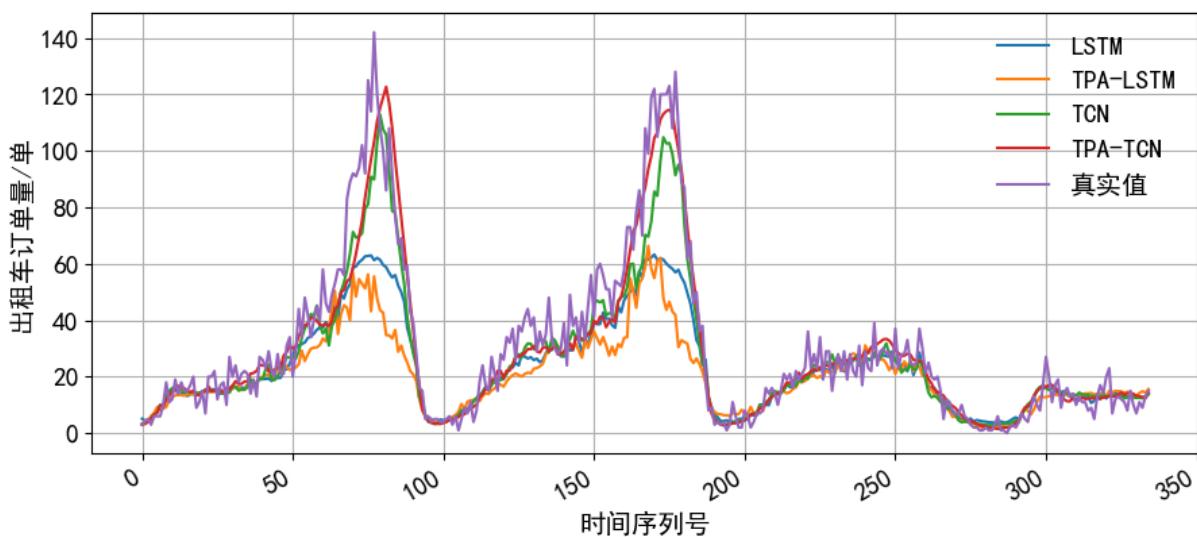


图 3.8 15min 间隙下测试集真实值与预测值

对比图 3.5 可以看出,以 15 分钟为间隔进行时间切片后,相邻时序出租车订单量的

波动性更为明显，与此同时单个时间切片内的出租车订单量更少，因此在进行预测任务时拟合难度会有所提高，预测精度也会相应降低。

3.4.2 芝加哥出租车订单量预测

为了进一步考察各个模型的性能，以及区域订单数量等因素对预测精度的影响，对芝加哥数据集进行训练，损失值如图 3.9 所示。

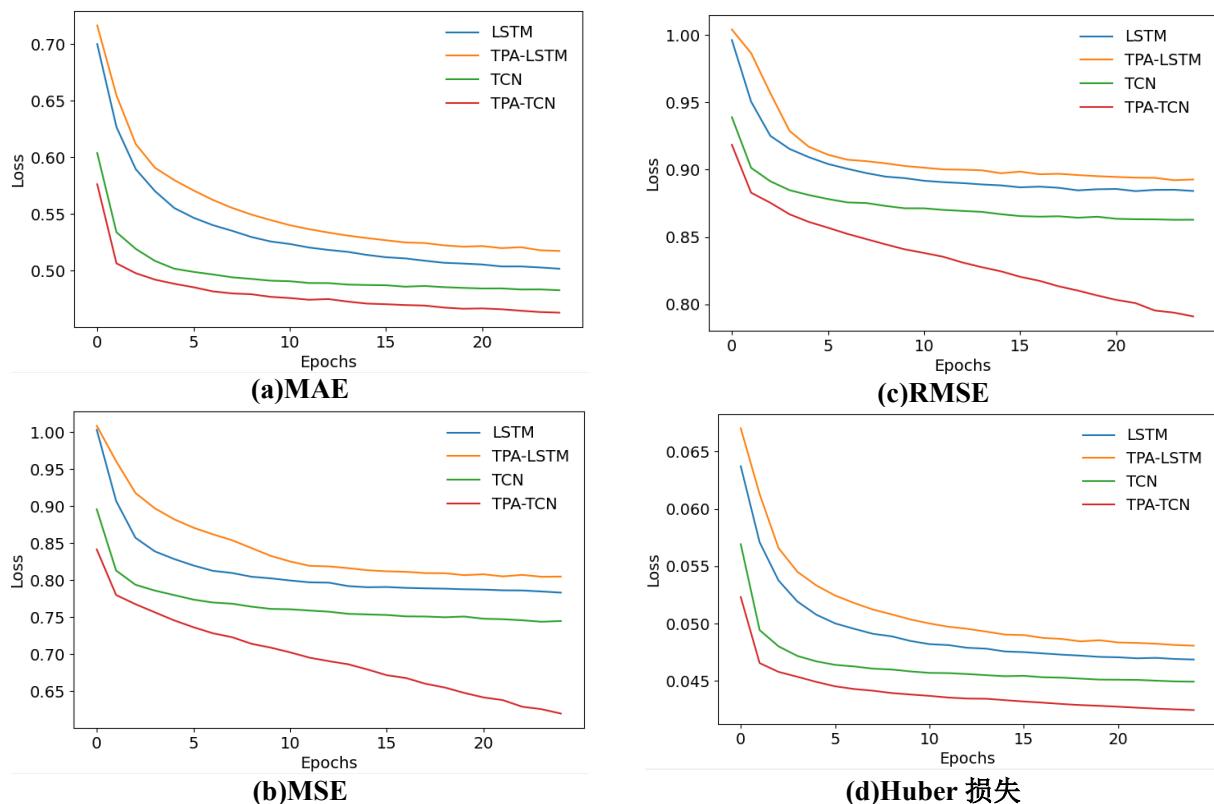


图 3.9 芝加哥训练集损失值变化

与在纽约曼哈顿数据集上训练模型的情况类似，在芝加哥数据集上进行训练时，TPA-LSTM 模型的损失值最高，TPA-TCN 模型的损失值最低。但在芝加哥数据集上进行训练时，随着训练迭代次数的增加，TPA-TCN 模型的损失值并未趋于稳定，尤其是用 MSE 和 RMSE 训练的模型，损失值仍以较快的速度在降低，说明模型仍在缓慢收敛。

这种情况出现一般有两种原因：①模型训练还不够充分，增加训练迭代次数之后模型还可以继续进行有效学习；②随着训练的进行，模型越来越拟合训练集，出现过拟合情况，导致模型泛化能力越来越差；③梯度消失/爆炸。在循环神经网络（如 LSTM）中，可能会出现梯度消失或爆炸的问题，导致模型训练困难。

TCN 通常采用的是直接连接(residual connection)或者扩张卷积(dilated convolution)，可以使得梯度更容易传播，减轻梯度消失的影响。同时，由于 TCN 中的卷积操作是局部操作，并且参数共享的机制使得参数数量相对较少，因此也不容易出现梯度爆炸的问

题。在 3.4.1 节中，经过与测试集损失值的对比，发现 TPA-TCN 模型已经出现过拟合风险，因此对于在芝加哥数据集上出现的情况，认为随着迭代次数增加，模型已经不再有效训练，继续迭代只会使模型的泛化能力降低。

将训练集和测试集损失值进行对比，如表 3.3 所示。

表 3.3 芝加哥训练集与测试集损失值比较

模型 损失	LSTM		TPA-LSTM		TCN		TPA-TCN	
	训练集	测试集	训练集	测试集	训练集	测试集	训练集	测试集
MAE	0.5017	0.4960	0.5174	0.5126	0.4827	0.4869	0.4628	0.4791
MSE	0.7833	0.7714	0.8049	0.7982	0.7449	0.7628	0.6198	0.7685
RMSE	0.8841	0.8745	0.8926	0.8912	0.8628	0.8708	0.7910	0.8725
Huber	0.0469	0.0463	0.0481	0.0476	0.0449	0.0454	0.0425	0.0452

在模型训练迭代 25 次的前提下，TPA-TCN 模型在测试集上的各项损失值指标均高于训练集，说明模型已经出现了过拟合的情况，如果继续增加训练回合数，不但无法增加预测精度，反而会出现更严重的过拟合。用 MAE 和 Huber 损失进行训练时，TPA-TCN 模型在测试集上的损失值仍为最低，但用 MSE 和 RMSE 训练的模型，测试集损失值已经超过 TCN 模型，说明过拟合已经对模型的泛化能力以及预测精度产生了不利影响。

纽约曼哈顿和芝加哥的训练集测试集损失值对比如图 3.10 所示。

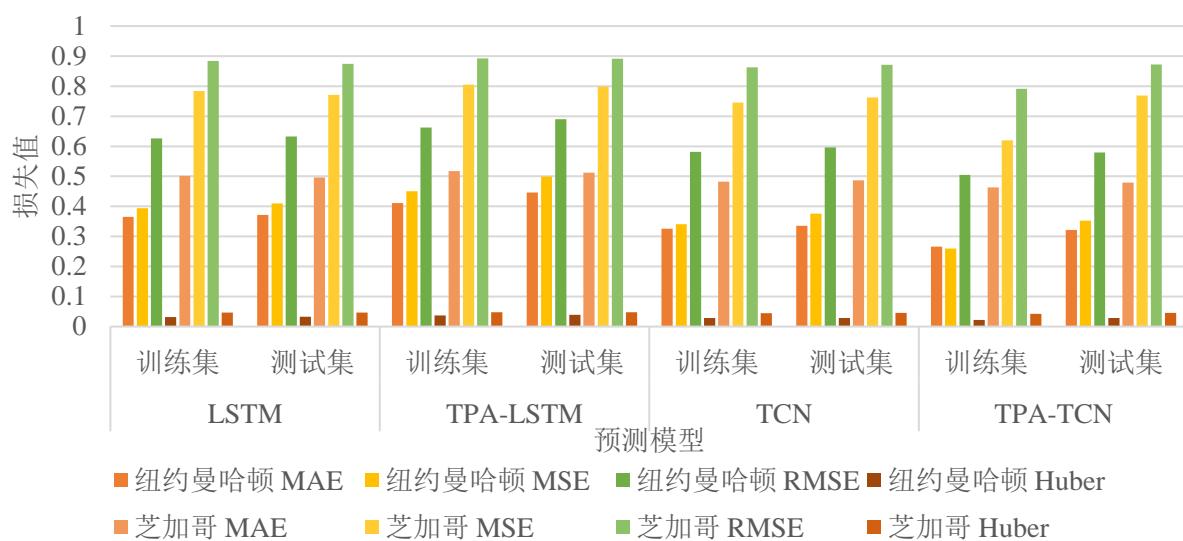


图 3.10 纽约曼哈顿与芝加哥损失值对比

观察图 3.10 并对比表 3.1 与表 3.3 发现，芝加哥数据集训练预测时的各项损失值均大于纽约曼哈顿数据集上的损失值。预测模型的模型在测试集上的 MAE、MSE、RMSE 和 Huber 损失分别增大了约 33.88%、89.30%、40.47% 和 42.69%。

结合数据集情况，分析用芝加哥数据集进行训练时过拟合更为严重以及损失值大幅增大这两种情况产生的原因。芝加哥数据集和纽约曼哈顿数据集的时间跨度均为 84 天，

芝加哥数据集中总的出租车订单数量为 1413290，而纽约曼哈顿数据集中总的出租车订单数量为 6590696（除去数据异常的三天产生的订单），单个时间切片内产生的订单数量远远高于芝加哥。且在 2.2 节的数据空间特征分析中可以看出，芝加哥出租车订单在各区域的分布几不均衡，在全部 77 个分区中，只有四个分区在 84 天内的订单产生量超过 100000 条，其他区域样本量严重不足，对预测的准确度会产生不利影响。绘制芝加哥编号为 7 的区域一周内订单量真实值和预测值，得到图 3.11。虽然在训练阶段，TPA-TCN 模型的损失值偏低，但从测试集上的预测结果以及损失值来看，TPA-TCN 模型的预测精度相比于其他三个模型没有明显优化。

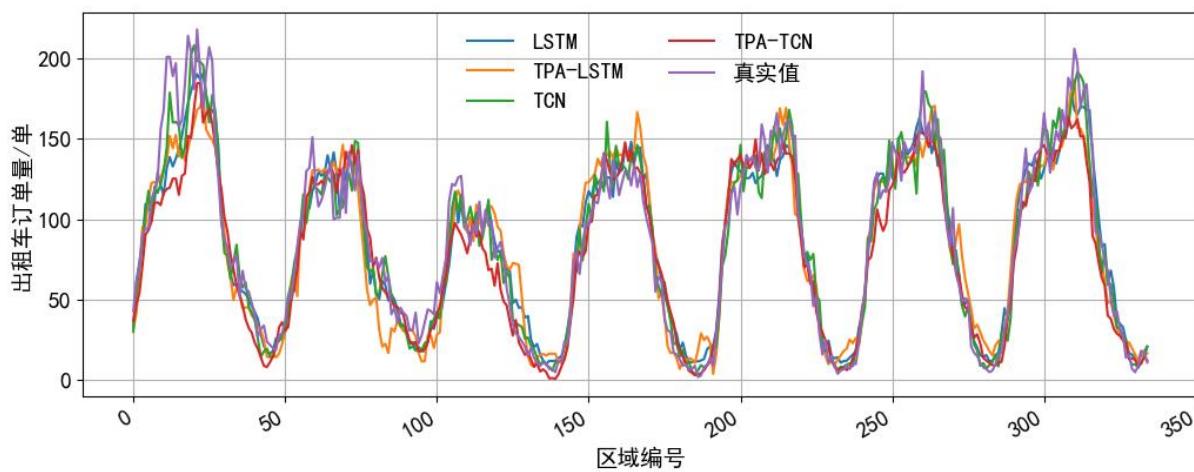


图 3.11 单个区域一周内订单量真实值和预测值

绘制各分区订单量以及预测结果在测试集上的损失值如图 3.12 所示。

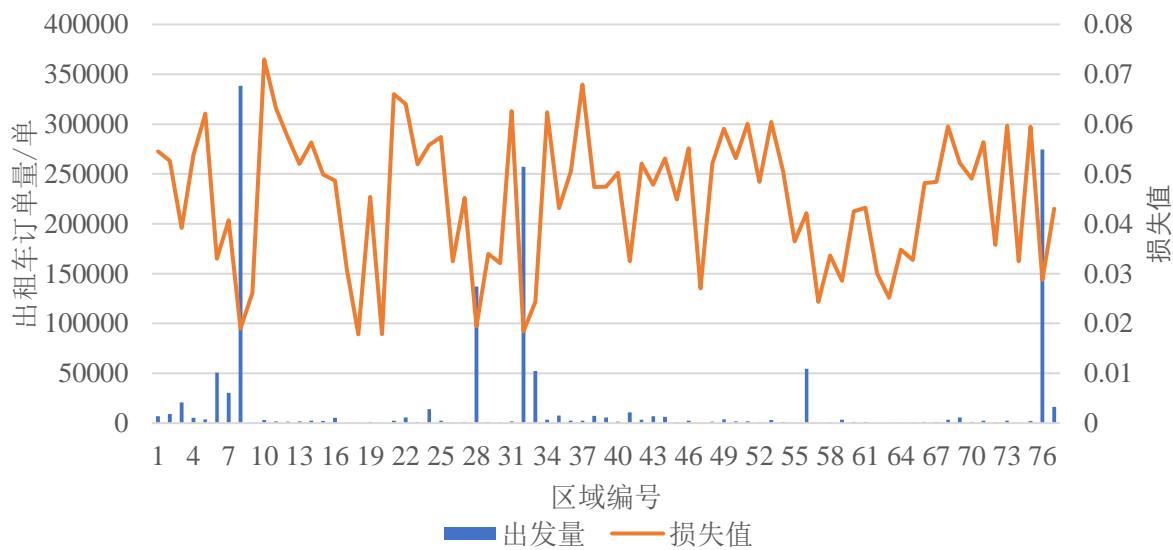


图 3.12 芝加哥各分区订单出发量与预测结果损失值

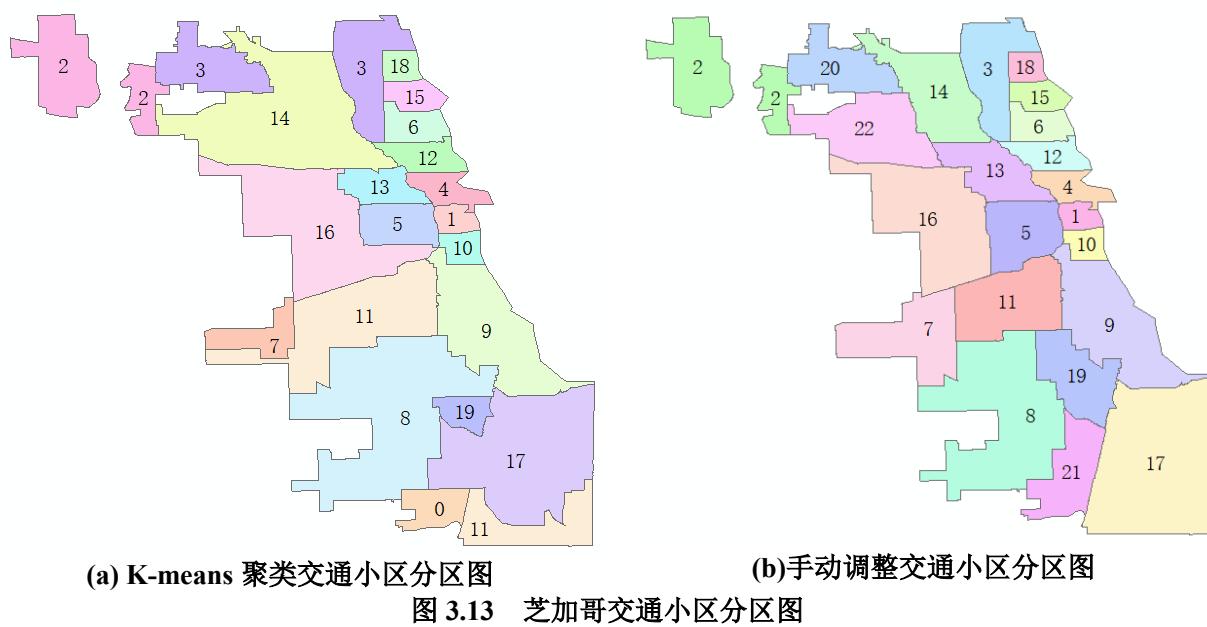
与 3.4.1 节中纽约曼哈顿的预测结果类似，对于订单量较多的区域，订单变化趋势表现出较强的规律性，模型的预测结果也与真实值较为接近，损失值相比于订单量小的

区域要更低。全部 77 个分区中只有 5 个分区在单个时间切片内的订单量超过了 20 单，在实际的预测任务中，可以选择部分重点片区进行预测或者按照一定的规则对区域进行重新划分、合并，以更好地对区域需求订单数量进行预测，并在此基础上执行预调度任务。

在本研究中，采用 K-means 聚类^[62]的方法重新划分交通小区，对芝加哥订单进一步预测。K-means 是一种常用的聚类算法，用于将数据集划分为 k 个不同的类别。其核心思想是通过迭代优化一组初始聚类中心，将数据点分配到最近的聚类中心，并更新聚类中心以最小化每个聚类内部的方差。

K-means 聚类算法的基本步骤为：①初始化：随机选择 k 个数据点作为初始聚类中心 $a = a_1, a_2, a_3, \dots, a_k$ ；②数据分配：对于数据集中的每个数据点 x_i ，计算其与 k 个聚类中心的距离，并将其分配给距离最近的聚类中心所在的类别；③更新聚类中心：对每个类别 a_j ，重新计算其所有数据点的平均值 $a_k = \frac{1}{|c_i|} \sum_{x \in c_i} x$ ，作为新的聚类中心；④重复迭代：重复步骤②和步骤③，直到聚类中心不再发生变化或达到最大迭代次数。

K-means 聚类算法易于实现，且适用于大规模数据集。但对初始聚类中心敏感，且聚类数目需要手动设置。在本研究中，运用 K-means 聚类算法将具有相似时间序列且位置接近的小区聚为一类。选择 20 个初始聚类中心，得到交通小区分区图如图 3.13(a)所示。经过聚类之后共划分 20 个交通小区，交通小区编号为 0 至 19。



通过观察可以发现，用 K-means 聚类算法聚类得到的交通小区存在分区不合理的情况，如编号为 3 的小区和编号为 11 的小区在空间上不连续，14、16 等小区过大，如果

用图 3.13(a)的分区进行预测并指导车辆调度，不利于调度方案的计算和生成。因此手动调整交通小区分区，最终共设置了 22 个交通小区，并将小区编号修改为从 1 开始至 22，如图 3.13(b)所示。

在此基础上重新对芝加哥数据集进行预测，并于重新划分小区前的测试集损失值进行比较，得到表 3.4 和图 3.14。

表 3.4 重新划分小区前后测试集损失值比较

模型	LSTM		TPA-LSTM		TCN		TPA-TCN	
	77 小区	22 小区	77 小区	22 小区	77 小区	22 小区	77 小区	22 小区
损失								
MAE	0.4960	0.4318	0.5126	0.4676	0.4869	0.4295	0.4791	0.4322
MSE	0.7714	0.3829	0.7982	0.4257	0.7628	0.3829	0.7685	0.3765
RMSE	0.8745	0.6144	0.8912	0.6481	0.8708	0.6162	0.8725	0.6136
Huber	0.0463	0.0384	0.0476	0.0433	0.0454	0.0384	0.0452	0.0370

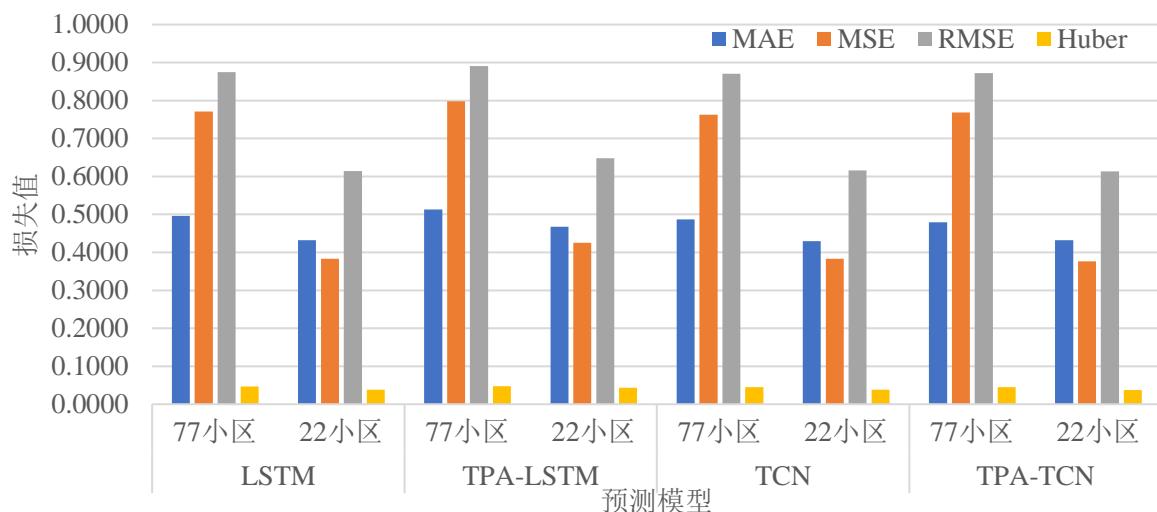


图 3.14 重新划分小区前后测试集损失值比较

重新划分交通小区之后，各项损失值有所降低，模型预测结果与真实数据的拟合度提高。

综合考虑模型的稳定性、预测精度、过拟合情况等，选择用 Huber 损失训练的 TPA-TCN 作为预测模型，预测结果参与到下一步的车辆调度任务中。

3.5 本章小结

本章主要对出租车订单的短时需求预测任务进行研究。首先确定了本研究在订单预测阶段的任务，即将城市划分为多个区域，以一定的时间间隔将时间划分为多个时间切片，针对每个区域内出租车出发量和到达量，用已经观测到的时间切片内的出租车历史行程，预测下一个时间切片该区域的出租车出发数量和到达数量，将其抽象为时间序列预测问题。随后介绍了几种常见的时间序列预测的深度学习方法，考虑 LSTM 模型、

TCN 模型和时序模式注意力机制，确定预测模型构建和训练时数据的处理方式、超参数确定。最后对模型的预测结果进行评价，认为在本研究中涉及的出租车订单量预测问题中，用 Huber 损失进行训练时，模型的收敛速度最快；引入时序模式注意力机制的 TCN 模型在预测精度、模型稳定性和模型泛化等方面表现较优。

第四章 车辆区域供需再平衡

出租车运营过程中，有时存在区域供需不匹配的情况。尤其在高峰期，城市整体用车紧张，只有在对未来短时间内的各区域的出行需求准确预测的基础上，进行合理地区域再平衡，使空闲车辆在即将产生大量订单的片区提前就位，才能在用户发起订单后在较短距离内匹配到可用车辆，减少乘客等待时间。因此，当前网约车平台会设置激励机制，鼓励司机接取调度任务，但由于收到经济效益、感知成本等因素的影响，出租车司机参与调度积极性不高。共享无人车投入运营后，平台在对共享无人车指挥调度时无需考虑对司机的奖励，只需考虑车辆和能源的损耗等因素，一定程度上可以缓解平台在车辆调度方面的压力。

在第三章中，通过建立出租车订单预测模型，我们能够较为准确地预测下个时间段各个交通小区的订单需求量和即将到达该交通小区下客的运营中车辆。在实际场景中，通过收集和分析历史出租车订单数据，采用实时预测的方式识别车辆供需不平衡的趋势，可以提前启动区域调度以优化车辆分配。本研究在短时需求预测的基础上，考虑共享无人车和有司机驾驶的出租车产生的不同调度成本，从系统最优的角度出发，同时尽可能对有司机驾驶的出租车的调度时间、调度距离进行限制，构建区域调度模型，为车辆分配调度任务，以达到出租车区域供需再平衡的目的。

4.1 出租车区域供需再平衡问题描述

本文所研究的出租车调度问题为区域间的供需再平衡，因此开展调度的前提是城市内不同区域存在供需失衡的趋势。在判断下个时段各区域的出租车供需是否失衡时，应充分考虑下时段各区域可能产生的订单总需求以及出租车辆总供给。下时段区域内的供给车辆主要有两类：一类是空闲车辆，另一类是预计在下时段在该区域下客的车辆。

假设城市可以根据某些原则方法被划分为一定数量的交通小区，交通小区的总数量为 l ，城市交通小区集合为 $S = \{1, 2, \dots, l\}$ 。对于编号为 p 的交通小区，该区域的下个时段的出租车订单需求数为通过短时需求预测得到的区域出租车出发量，用 O_p 表示；区域下个时段空闲车辆数用 U_p 表示；区域即将到达的出租车辆数为预测的下个时段该区域的出租车到达量，用 D_p 表示。区域调度的目的即为，假设预测的各区域出租车订单需求数据准确，对各交通小区的出租车出行需求数量和区域供给数量进行区域间调配，尽可能使各区域出租车辆总供给减去总需求不小于 0，即 $U_p + D_p - O_p \geq 0$ 。

在某些情况下，城市出租车辆的总供给可能无法满足总需求，即无法保证全部区域的富裕车辆数 $U_p + D_p - O_p \geq 0$ 。在实际出租车运营中，区域调度的目的是调整城市内各片区的供需关系，辅助派单调度的高效开展。因此无需将全部区域富裕车辆数大于 0 作为区域调度任务必须达成的目标。但为了便于计算调度方案，需要遵循一定的原则。在实际出租车运营调度中通常会考虑需求优先级、调度时间、运营收益等因素，在本研究中，如果可调度的车辆总数无法满足全部区域下时段的订单需求，则通过调度使每个区域的空缺车辆数尽可能均匀分布，避免出现车辆缺口集中于单个区域的情况。为此，构建一个线性规划模型来调整交通小区需求，将实际需求数转化为可供分配数，确保每个交通小区的车辆需求尽可能接近目标值，同时不超过总供给车辆数。假设在车辆总供给不足的情况下，各个交通小区调整后的需求数为 o_p ，线性规划模型如下：

$$\text{Min} \sum_{p=1}^l [O_p - o_p] \quad (4.1)$$

$$\text{s.t. } \sum_{p=1}^l o_p \leq \sum_{p=1}^l (U_p + D_p) \quad (4.2)$$

$$0 \leq o_p \leq O_p, \forall p \in V \quad (4.3)$$

求解该模型后，得到调整后的各交通小区车辆需求，在此基础上求解调度方案。

在调度时还应满足以下几点原则：

1. 调度时优先调度共享无人车，对于有司机驾驶的出租车，调度时间和调度距离尽可能小；
2. 车辆调度应在一定的时间限制内完成；
3. 本文期望得到的出租车辆区域调度方案为系统最优方案而不是用户最优方案，因此在所有满足以上条件的调度方案中，选择使整体调度费用最低的方案。

4.2 出租车区域供需再平衡调度模型构建

在这部分内容中，对调度流程中的初步调度方案生成和最终调度方案生成进行建模，定义调度过程中涉及到的参数，根据调度方案包含的信息确定决策变量，并对车辆调度需要满足的目标函数和约束条件进行定义和说明，将实际中的车辆调度问题转化为数学问题。

4.2.1 区域调度相关参数

首先根据调度任务的需要，对调度模型中涉及的参数进行定义。

1. 交通小区

在进行区域再平衡调度时，首先应该将城市划分为若干大小适宜的交通小区，以交通小区为单位进行出租车订单预测，在此基础上制定调度方案。

(1) 交通小区集合为 $S = \{1, 2, \dots, l\}$ ，其中交通小区的总数量为 l 。

(2) 交通小区 p 的需求量 O_p 和即将到达车辆数 D_p 为通过第三章中的预测模型得到的区域出发量和到达量，交通小区 p 的空闲车辆数 U_p 为当前区域存有的空驶车辆数。

(3) 用 dis_{pq} 表示交通小区 p 和 q 质心间的直线距离。

2. 路网

在模型算法的构建过程中，需要将城市路网抽象为图结构，因此在此处介绍图的基本概念：对于图 $G = (V, E)$ ， V 为节点的集合， E 为边的集合，对于每个节点 i ，均有其特征 x_i ，可以用矩阵 $X_{N \times D}$ 表示。其中 N 表示节点数， D 表示每个节点的特征数，也可以说是特征向量的维度。各个节点之间的关系会形成一个 $N \times N$ 维的矩阵 A ，称为邻接矩阵（adjacency matrix）。

对于城市路网所抽象成的图结构，道路交叉口即为图结构中的节点，连接各个路网节点的路段即为边的集合。在路网结构的基础上建立两个邻接矩阵，其中一个邻接矩阵记录节点之间路段的长度，另一个邻接矩阵记录节点之间路段的速度。在构建调度模型时，需要明确每个路网节点所属的分区，将车辆位置与距离最近的路网节点进行匹配，实现车辆在路网节点之间的调度，进而实现车队的区域调度。

(1) 路网节点集合为 $V = \{1, 2, \dots, n\}$ ，节点总数量为 n ；为每个路网节点定义其所在区域，编号为 i 的路网节点所在区域编号为 s_i 。因此每个交通小区中包含的路网节点数

(2) 定义 $n \times n$ 维的邻接矩阵 $length$ 和 $n \times n$ 维的邻接矩阵 $speed$ ，其中 $length_{ij}$ 表示节点 i 和节点 j 之间路段的长度， $speed_{ij}$ 表示节点 i 和节点 j 之间路段的通行速度。用 L_{ij} 定义车辆从节点 i 调度至节点 j 的行程距离，用 T_{ij} 定义车辆从节点 i 调度至节点 j 花费的总时间。 L_{ij} 和 T_{ij} 应由邻接矩阵 $length$ 和 $speed$ 计算得出。

3. 车队及车辆属性

(1) 混合车队中包含的车辆的集合 $K = \{1, 2, \dots, m\}$ ，车辆总数为 m 。

(2) 用 0-1 变量 a_k 定义车辆是否有司机驾驶，若 $a_k = 0$ ，则为共享无人车；若 $a_k = 1$ ，则为有司机驾驶的出租车。

(3) 用 0-1 变量 b_k 定义车辆的行驶状态。 $b_k = 0$ 为载客状态； $b_k = 1$ 为空驶状态。

(4) 定义车辆调度成本。定义编号为 k 的车辆在路网上移动单位距离产生的成本为 c_k ，对于有司机驾驶的出租车，进一步考虑在路网上移动单位时间产生的成本为 t_k 。

(5) 将车辆匹配至距离当前地理位置最近的路网节点，编号为 k 的车辆所在路网节点编号为 v_k ，编号为 k 的车辆所在的区域编号则为 s_{v_k} 。因此，区域空闲车辆数

$$U_p = \sum_{\forall s_{v_k} = p} b_k$$

4. 调度状态

用 h_{ij}^k 定义编号为 k 的单个车辆从路网节点 i 调度至路网节点 j 花费的总成本。

全部的参数定义如表 4.1 所示。

表 4.1 车辆调度模型变量和参数定义

符号	含义
S	交通小区集合
l	交通小区总数
O_p	交通小区 p 下个时段出租车需求量
D_p	交通小区 p 下个时段出租车到达量
U_p	交通小区 p 当前时段空驶车辆数
dis_{pq}	交通小区 p 和 q 质心间的直线距离
V	路网节点集合
n	路网节点总数
s_i	路网节点 i 所在交通小区编号
$length_{ij}$	节点 i 和节点 j 之间路段的长度
$speed_{ij}$	节点 i 和节点 j 之间路段的通行速度
L_{ij}	从节点 i 调度至节点 j 的行程距离
T_{ij}	从节点 i 调度至节点 j 的行程时间
K	车辆集合
m	车队中的车辆总数
v_k	车辆 k 所在路网节点编号
a_k	车辆 k 是否有司机驾驶
b_k	车辆 k 是否空驶
c_k	车辆 k 调度单位距离成本
t_k	车辆 k 调度单位时间成本
h_{ij}^k	编号为 k 的车辆从节点 i 调度至节点 j 花费的总成本

4.2.2 区域调度模型决策变量

实际中城市内的出租车辆组成的车队往往规模较大，在求解调度方案时容易因计算量过大而难以求解。因此本研究将区域调度方案的求解划分为两个阶段，第一阶段求解交通小区间调度的车辆数，得到初步调度方案；第二阶段在初步方案的基础上为具体的车辆分配点对点调度任务，得到具体调度方案。

首先定义一个元素均为非负整数的二维矩阵 X ，矩阵 X 的大小为 $l \times l$ (l 为交通小区总数)，其中 X_{pq} 的值为从交通小区 p 向交通小区 q 调度的出租车辆数。通过求解 X ，得到初步调度方案。

在求解 X 的基础上，对矩阵 X 中的值进行遍历，计算参与调度的车辆总数 x 。若 $x > 0$ ，建立一个含有 x 个元组的元组列表 Y ， Y 中每个元组的格式为 (i, j, k) ，其中 i 为调出路网节点编号， j 为调入路网节点编号， k 为调度车辆编号。对 Y 进行求解，得到最终的具体调度方案。

4.2.3 区域调度模型目标函数

在求解初步调度方案 X 时，由于没有具体的调度成本，只有区域质心之间的直线距离，因此将目标函数设置为使系统中全部车辆在质心间移动距离最短，即：

$$\text{Min} \sum_{p,q=1}^l dis_{pq} X_{pq} \quad (4.4)$$

对于具体调度方案 Y ，在求解时目标函数为系统调度成本最小，即：

$$\text{Min} \sum_{\forall(i,j,k) \in Y} h_{ij}^k \quad (4.5)$$

4.2.4 区域调度的约束条件

1. 初步调度方案

(1) 定义每个区域可调出的车辆数。

对于任意一个交通小区 p ，从该区域调出的车辆总数应不超过该交通小区当前的空闲车辆数，即：

$$\sum_{q=1}^l X_{pq} \leq U_p \quad (4.6)$$

(2) 定义每个交通小区可供车辆调入和调出的总次数。

为了使生成的调度方案更加灵活，在本研究中，并不是只将区域富余量大于 0 的交通小区作为调出区域、将富余量小于 0 的交通小区作为调入区域，而是所有区域都可以

参与调度。在完成调度后，每个区域的区域富余量应不小于 0 即可：

$$U_p + D_p - O_p - \sum_{q=1}^l X_{pq} + \sum_{q=1}^l X_{qp} \geq 0, \quad \forall p \in V \quad (4.7)$$

其中， $\sum_{q=1}^l X_{pq}$ 为从交通小区 p 调出的全部车辆数， $\sum_{q=1}^l X_{qp}$ 为从其他交通小区调入交通小区 p 的全部车辆数。供给车辆数不小于下时段实际需求数，则式(4.7)中的 O_p 为实际需求数；若供给车辆数小于下时段实际需求数，则式(4.7)中的 O_p 为按照式(4.1)、式(4.2)和式(4.3)所示的线性规划模型求解得到的调整后需求数 o_p 。

(3) 定义交通小区之间可调度车辆的最大距离。

为了防止在生成初步调度任务时，车辆调度距离过远，因此对于可参与调度的调度起始区域和调度目标区域质心间距离进行限制。如果交通小区 p 和交通小区 q 的质心间距离超过 ρ ，则不能车辆不能直接在交通小区 p 和 q 之间进行调度，即：

$$X_{pq} = 0, \quad \forall dis_{pq} > \rho \quad (4.8)$$

2. 具体调度方案

(1) 定义每辆车可以被调度的次数。

若车辆的行驶状态 $b_k = 0$ 为载客状态，则车辆不可被调度；若 $b_k = 1$ 为空驶状态，则车辆在一个调度周期内最多可以被调度一次。因此对于调度方案 Y 中出现的所有元组 (i, j, k) ，相同的 k 值最多只能出现 b_k 次，即：

$$\sum_{\forall(i,j,k) \in Y} [k = k_0] \leq b_{k_0}, \quad \forall k_0 \in K \quad (4.9)$$

其中 $[k = k_0]$ 为一个指示函数，当 $k = k_0$ 时取值为 1，否则为 0。

(2) 定义车辆 k 能否从节点 i 开始调度。如果车辆 k 当前所处的节点位置 $v_k \neq i$ ，则车辆 k 不能从编号 i 的节点开始调度，即：

$$v_k = i, \quad \forall (i, j, k) \in Y \quad (4.10)$$

(3) 全部车辆完成调度时的用时应不超过 τ ，即：

$$T_{ij}^k \leq \tau, \quad \forall (i, j, k) \in Y \quad (4.11)$$

(4) 对司机调度时间进行限制。定义一个新的 0-1 变量：

$$z_{ij} = \begin{cases} 0, & T_{ij} \leq \mu \\ 1, & T_{ij} > \mu \end{cases} \quad (4.12)$$

调度成本：

$$h_{ij}^k = c_k \cdot L_{ij} + t_k \cdot a_k \cdot T_{ij} + a_k \cdot z_{ij} \cdot M \quad (4.13)$$

在本研究中，定义共享无人车的调度成本=单位距离成本×调度距离，有司机驾驶的出租车的调度成本=单位距离成本×调度距离+单位时间成本×调度时间。通过定义不同的调度成本，来区分有人驾驶的出租车和共享无人车之间的调度差异，引导模型求解时尽可能地优先对共享无人车进行调度。并且，如果调度的目标车辆有司机驾驶，且该车辆的调度时间大于限制时间 μ ，则进一步增加调度成本。

与其他约束不同的是，该约束条件是软性约束而不是硬性约束。采用增加调度成本的方式，使方案对应的系统总调度成本增大。由于该模型的目标函数为使总的调度成本最小，因此在搜索解的时候，如果有能使系统总调度成本更小的方案，会将违反第五项约束条件的调度方案舍弃。而如果区域间用车紧张，无法搜索到满足所有约束的更优方案，可以将违反该约束条件的方案输出。

4.3 出租车区域供需再平衡调度模型求解

在本研究中要解决的车辆调度问题是一个组合优化问题 (Combinatorial optimization problem, COP)，旨在有效地分配车辆和任务，并优化调度策略以最大程度地满足一组约束条件和目标函数。组合优化问题是一类在离散状态下求极值的最优化问题，其数学模型如下所示：

$$\min F(x) \quad (4.14)$$

$$s.t. G(x) \geq 0 \quad (4.15)$$

$$x \in D \quad (4.16)$$

其中 x 为决策变量、 $F(x)$ 为目标函数、 $G(x)$ 为约束条件， D 表示离散的决策空间，为有限个点组成的集合。

组合优化问题属于 NP-hard 问题^[63]，在实际应用中，组合优化问题的建模往往并不复杂，但对大规模组合优化问题的求解一直是一个难点。当前，组合优化问题求解算法主要包括精确方法 (Exact approaches) 和近似方法 (Approximate approaches)^[64]。精确方法是可以得到全局最优解，最常见的为分支定界法。随着问题规模的扩大，精确方法的计算量会呈指数级增长，对计算速度和计算性能的要求也随之提高。相比于精确方法，近似方法退而求其次，寻找局部最优解，主要分为近似算法 (Approximate algorithms) 和启发式算法 (Heuristic algorithms)。近似算法通常有性能保证，通过定义近似比例等参数，衡量算法找到的解决方案与最优解之间的差距，常见的近似算法有贪婪算法、近

似比例算法等；而启发式算法则依赖于经验和启发性规则来生成解决方案，能够在可行时间内找到一个较好的解，但是对解的质量没有保证，常见的有粒子群算法、遗传算法、蚁群优化算法等。

在本研究中，尝试使用运筹优化工具 OR-Tools 中的 CP-SAT(Constraint Programming - Satisfiability) 求解器^[65]对建立的车队调度模型进行求解。

OR-Tools 是由 Google 开发的开源求解器，用于解决车辆路径、流程、整数和线性规划以及约束编程等问题。CP-SAT 是 OR-Tools 中的一种约束编程求解器，由 Google Brain 团队推出，结合了约束编程(Constraint Programming)和布尔可满足性问题(SAT)求解技术。在解决约束程序设计和布尔可满足性问题时，CP-SAT 利用搜索和剪枝技术，采用自适应搜索策略、局部搜索和全局搜索相结合的方式，在大型和复杂的约束问题上表现出良好的性能。CP-SAT 的求解过程如下：

1. 预处理：将约束规划问题转化为一个整数线性规划(ILP)问题，并进行一些简化和优化，如删除冗余变量和约束、合并等价变量和约束、检测对称性和支配性等。
2. 搜索：使用启发式搜索策略在可行解空间中寻找最优解或证明无解。搜索过程中会使用冲突分析、重启、剪枝等技术来加速求解过程。
3. 输出：输出最优解或无解的结果，并给出一些求解过程的信息，如冲突数、支配数、重启数等。

CP-SAT 的搜索策略是基于有限域上的布尔可满足性问题(SAT)和伪布尔优化(PBO)。它将整数变量和布尔变量都编码为二进制位向量，并使用布尔逻辑运算来表示约束条件。在求解时，CP-SAT 使用了一种叫做 CDCL (conflict-driven clause learning) 的算法来进行搜索。CDCL 算法步骤如下：

1. 选择一个未赋值的变量，并给它一个随机或启发式的值，这叫做猜测(guess)。
2. 根据已赋值的变量和约束条件，推导出其他变量的值，这叫做传播(propagate)。
3. 如果没有发生冲突，即所有约束都被满足，那么继续选择下一个未赋值的变量并重复上述步骤；如果发生了冲突，即有某个约束被违反了，那么回溯到上一个猜测点，并改变该猜测点的值，这叫做回溯(backtrack)。
4. 在回溯过程中，还会根据冲突原因学习出新的约束条件，并加入到原始问题中，这叫做学习(learn)。学习出来的新约束可以帮助剪枝去掉一些不可行或次优的分支，从而缩小搜索空间。

本研究在 Python 中调用 ortools 库对调度模型进行求解，核心代码见附录 A。对车

辆调度模型求解后,得到车辆调度方案的调出区域编号、调入区域编号、调出节点编号、调入节点编号、调度车辆编号、调度车辆类型、调度车辆路径、每条调度行程的行程距离和行程时间、调度完成后各区域的供给车辆数以及调度过程产生的总成本。

本研究所建立的基于短时需求预测的出租车区域再平衡调度框架如图 4.1 所示。

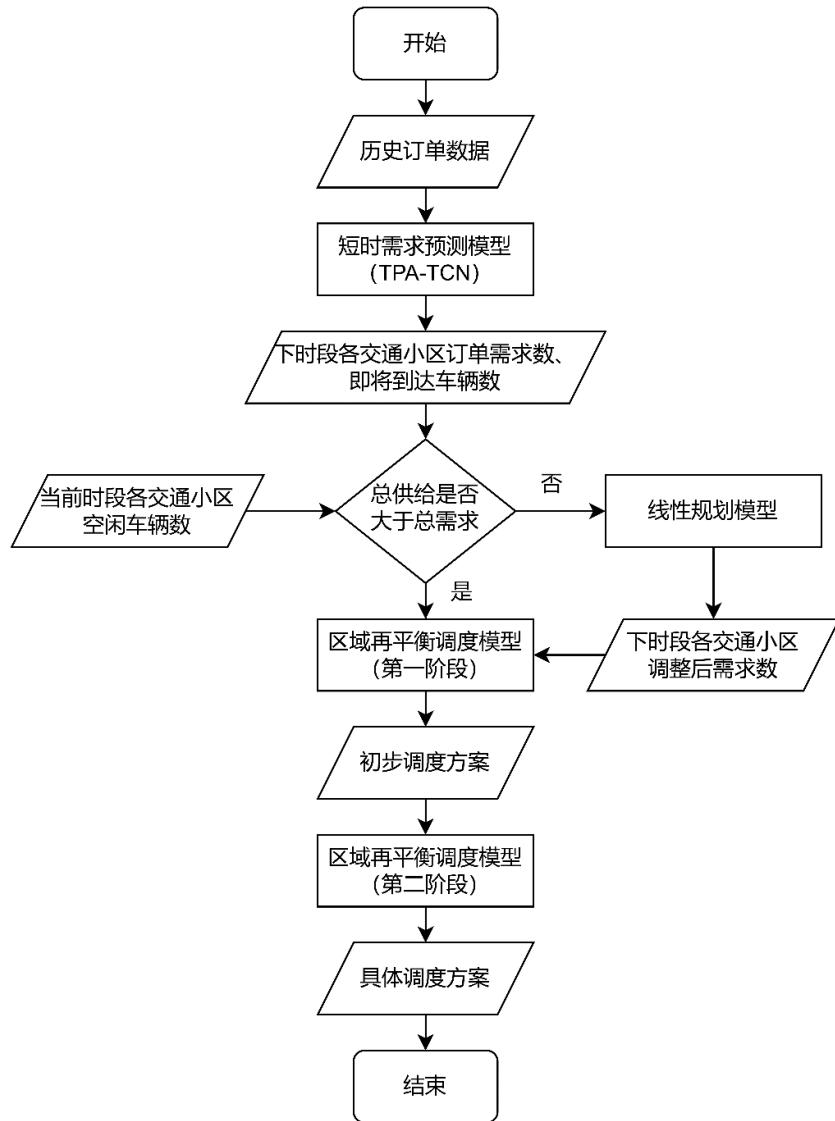


图 4.1 完整调度框架

4.4 本章小结

本章介绍了混合车队的区域调度模型的构建过程。首先对研究问题进行描述,明确车队区域调度的目标为在满足区域车辆需求的基础上,使系统整体调度费用最低。随后建立区域供需再平衡调度模型,明确了调度模型的相关参数、决策变量、目标函数以及约束条件,将车辆调度问题抽象为组合优化问题。最后确定了模型的求解方法,引入运筹优化工具 OR-Tools 中的 CP-SAT 开源求解器对模型进行求解并输出结果。

第五章 实例分析

在第三章和第四章中，建立了区域出租车订单短时需求预测模型和区域供需再平衡调度模型。为了检验建立的模型是否能够有效地实现区域出租车供给与需求的匹配，以及输出的调度方案是否合理，需要进行实例分析。本文以纽约曼哈顿和芝加哥为例，对模型进行仿真，输出实验结果，对模型效果进行评估。

5.1 调度模型实例应用参数设置

5.1.1 交通小区划分

在进行订单预测和区域调度之前需要将城市划分为若干个大小适宜的交通小区。在实际应用中，通常采用正六边形网格化(Hexagonal Grids)的方式，将整体划分为若干个大小相等的正六边形网格。相较于矩形网格，正六边形网格可以更紧密地填充区域，同时具有更接近圆形的外形以及更平均的邻近关系，在正六边形网格中，每个单元格与周围六个单元格相邻，且质心间距离相等，这种规则的邻近关系使得处理和分析图网络问题时更为简单一致。

然而，纽约曼哈顿出租车订单数据中缺少上下客点的经纬度信息；芝加哥出租车订单数据集中虽有经纬度信息，但经过了聚类处理，与实际情况不完全一致。为了尽可能保留真实性，在本研究中，在两地行政区划的基础上模拟调度方案的生成。

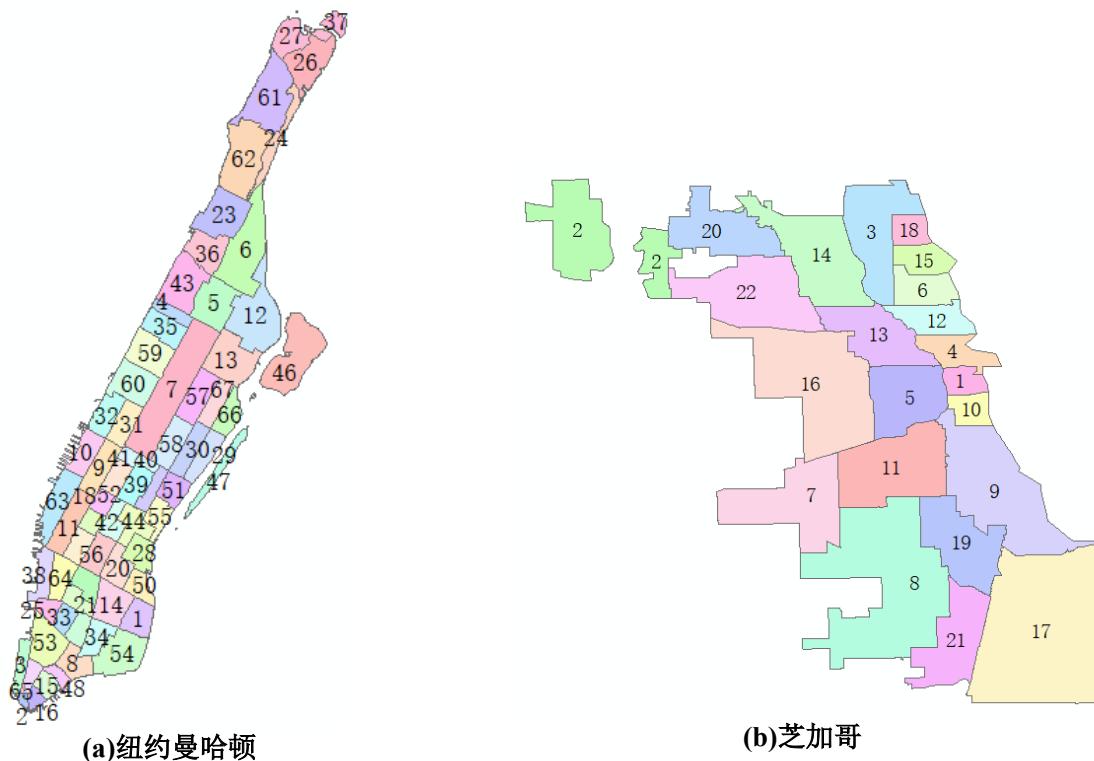


图 5.1 交通小区划分

通过 2.2.2 节的出租车行程空间分析，发现在芝加哥现有人口普查区分区下，各区域的订单分布极为不均衡，并且在 3.4.2 节中，以现有的分区进行订单需求预测难以保证预测精度，进而影响车队调度的有效性。因此，在 3.4.2 节中引入 K-means 聚类算法，最终将交通小区数量调整为 22 个。在本章中，分别将图 2.2(b)和图 3.13(b)所示的分区作为纽约曼哈顿和芝加哥调度模拟的交通小区。为了便于编程求解，将纽约曼哈顿的交通小区重新编号，如图 5.1(a)所示；芝加哥交通小区划分如图 5.1(b)所示。

为了了解两地的整体尺寸和各交通小区间的距离，计算交通小区两两之间的质心间直线距离，依据质心间距离绘制箱型图，如图 5.2 和图 5.3 所示。

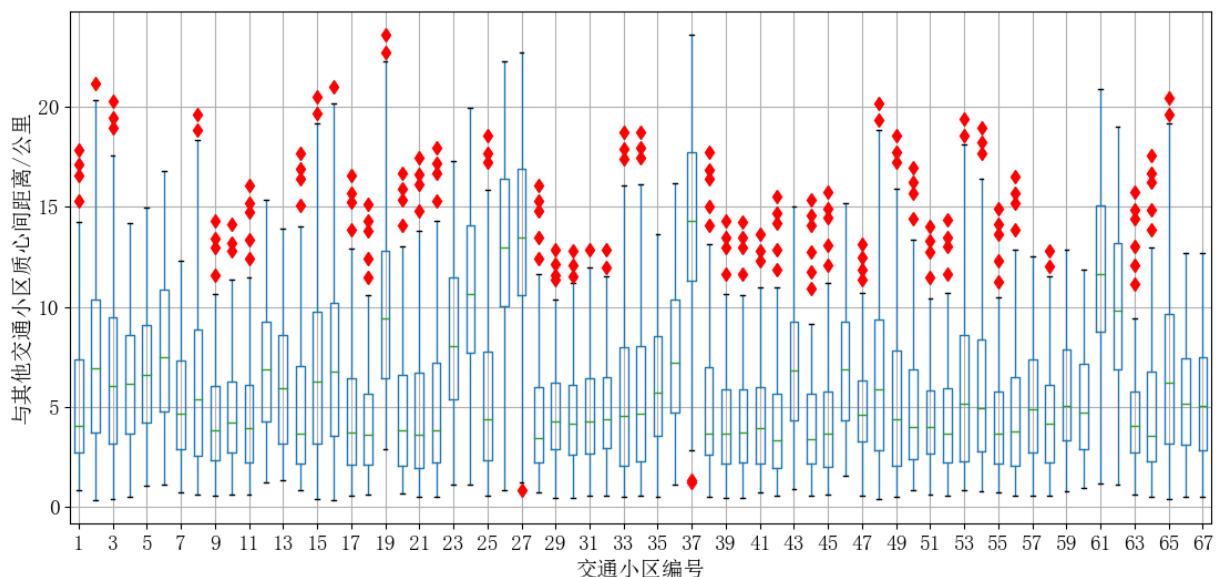


图 5.2 纽约曼哈顿交通小区质心间直线距离

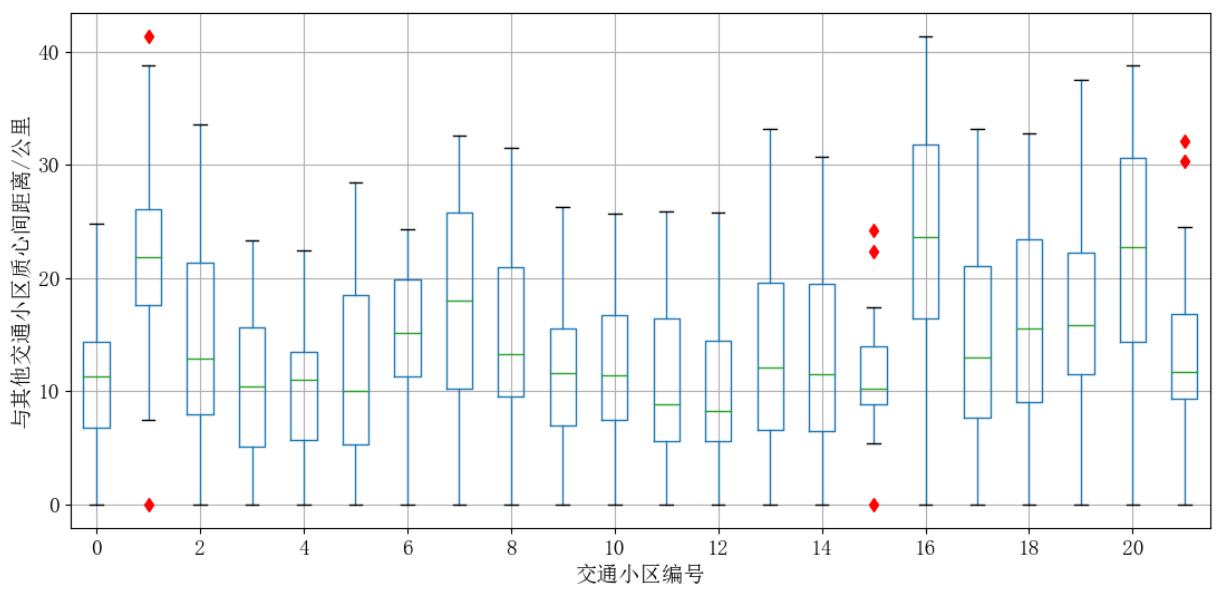


图 5.3 芝加哥交通小区质心间直线距离

对比图 5.2 和图 5.3 可以发现，相对于芝加哥城区，纽约曼哈顿的整体尺寸更小。

在纽约曼哈顿中，编号为 19 的交通小区与编号为 37 的交通小区质心间直线距离最长，约为 23.607 公里，多数小区质心间直线距离在 5 公里左右；对于芝加哥，交通小区 2 与交通小区 17 质心间的直线距离最长，约为 41.337 公里，多数小区质心间直线距离在 10-20 公里之间。

在 4.2 节构建调度模型时，公式(4.8)对调度区域的质心间直线距离进行了约束，以使调度时间尽可能短的同时能够输出可行的调度方案。调度的距离限制应与调度范围大小、交通小区的划分尺寸等因素有关。结合纽约曼哈顿以及芝加哥城区的交通小区质心间距离的分布情况，对于纽约曼哈顿，将 ρ 值确定为 5 公里，即如果交通小区 p 和交通小区 q 质心间直线距离超过 5 公里，则两个交通小区之间不能直接调度车辆。对于芝加哥，则将 ρ 值确定为 15 公里。

5.1.2 路网结构与路径计算

由于纽约曼哈顿数据中缺少上下客点的经纬度信息，因此在对纽约曼哈顿的车辆进行调度时，不考虑城市真实路网，因此不定义路网结构、计算路径。

对于芝加哥，本文使用的芝加哥路网数据来源于交通网络委员会整理的开源路网数据^[66]，包括路网节点、道路起止点、道路通行能力、道路长度、平均瞬时速度等字段，部分字段信息如表 5.1 所示。

表 5.1 芝加哥路网数据部分字段信息

字段	类型	示例
init_node	Int	1842
term_node	Int	1843
capacity	Int	1540
length	Float	0.31
free_flow	Float	0.744
speed	Int	24

在芝加哥路网数据中，共有路网节点 12981 个，连接 39017 条。路网覆盖范围较大，包含了芝加哥中心城区与周边地区的路网。而芝加哥出租车行程数据集中，上下车点位主要集中在中心城区，因此在人口普查区图形的基础上对原始路网进行删减，以更好地与出租车订单行程数据匹配。在原本的路网结构数据中，包含每条连接的通行速度，因此按照路段上不同的通行速度对路网结构进行展示。最终得到的路网如图 5.4 所示。

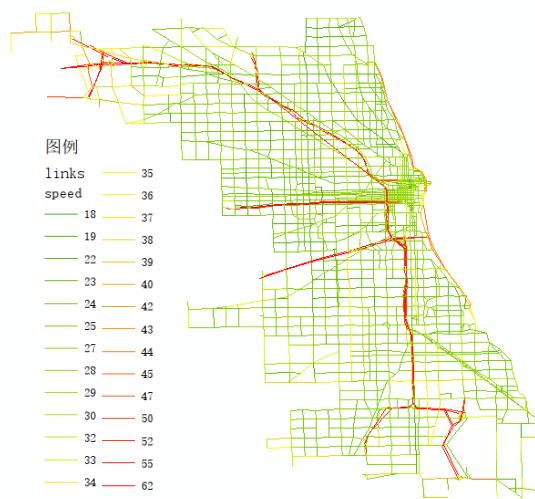


图 5.4 删减后的路网结构示意图

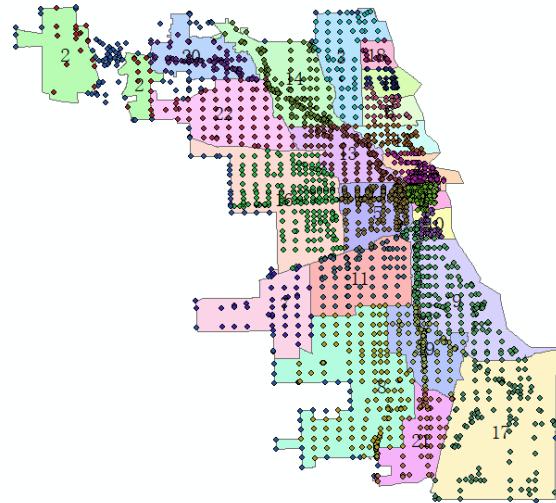


图 5.5 节点编号分类

最终处理完成的路网结构图中，共包含路网节点 2265 个，节点连接 7023 条。将节点与所在区域进行匹配，即为每个节点添加所在区域的编号。添加所在区域编号后的节点分类如图 5.5 所示。可以看出，在 GIS 中通过空间连接能较好地实现路网节点与所在区域的匹配。但由于部分路网节点位于芝加哥城区的外边界处，用 GIS 进行空间连接处理时无法识别，因此赋值为空。基于本研究的目的，引入路网结构图是为了更真实地模拟各区域间的车辆调度，因此边界处节点未匹配这一情况对车辆调度任务几乎没有影响。

根据字段中的路段开始节点、路段结束节点和路段长度建立距离邻接矩阵；根据路段开始节点、路段结束节点和通行速度建立速度邻接矩阵。

在 4.2.1 节建立的基于抽象网络的车辆调度模型中，车辆在构造的虚拟站点之间直线移动，无需对车辆移动的路径进行搜索。而在 4.2.2 节构建的基于真实网络的车辆调度模型中，引入了真实的城市路网结构，因此节点间的通行距离需要通过路径进行计算。本研究在路段长度矩阵的基础上，采用 Dijkstra 算法^[67]计算路网节点之间的最短路径。

Dijkstra 算法是一种用于在加权图中找到单源最短路径的经典算法，以一个节点作为起始点，逐步探索离起始点越来越近的节点，并更新到达每个节点的最短路径长度。其基本思路为：

- (1) 指定起点 s；
- (2) 引进两个集合 S 和 U，S 的作用是记录已求出最短路径的顶点(以及相应的最短路径长度)，而 U 则是记录还未求出最短路径的顶点(以及该顶点到起点 s 的距离)；
- (3) 设置初始状态，初始时 S 中只有起点 s，U 中则包含除 s 之外的顶点。将起始节点的最短路径长度设为 0，其他节点的最短路径长度设为无穷大；
- (4) 从 U 中找出路径最短的顶点，并将其加入到 S 中；接着，更新 U 中的顶点和

顶点对应的路径。重复该操作，直到遍历完所有顶点，最终在加权图中找到从起始节点到其他所有节点的最短路径长度。

用 Dijkstra 算法寻找最短路径后，用路径中经过的路段长度/通行速度，即可得到节点之间的通行时间。

5.1.3 车辆总数与车辆状态

在 4.2 节的模型构建中，需要确定车队中的车辆总数，以及各个车辆的以下几方面状态：①是否有司机驾驶；②是否空驶；③当前所在站点编号。

由于纽约曼哈顿数据集中缺少出租车 ID 信息，因此难以从数据集中对车辆状态进行识别，因此直接进行手动标定。在纽约曼哈顿的调度模拟中，确定当前空闲车辆总数为 1000，按照 1-1000 为每辆车编号。空闲车辆均匀地分布在全部 67 个交通小区中，1-62 号交通小区每个小区内各有 15 辆空闲车辆，63-67 号小区内各有 14 辆空闲车辆。

由于本研究使用的数据集为出租车订单数据，未涉及共享无人车，因此混合车队中共享无人车占比需要进行假设。在纽约曼哈顿的实例分析中，空闲的共享无人车占车队中所有空闲车辆的比例为 50%。同时增设两组实验，其他参数相同，但共享无人车占比分别为 25%、75% 作为对照。共享无人车标记在 Python 中调用 random 函数随即生成。

对于芝加哥的车辆总数与车辆状态，采用如下方式进行标定：

1. 车队中车辆总数

芝加哥出租车行程数据中包含出租车 ID、出租车行程的上下车时间、上下车点所在分区等信息。因此，在确定车队总数时，遍历调度目标时刻当天的所有出租车行程数据，获取当天共有多少车辆参与接取订单，作为参与调度的车队总数。选择 2023 年 10 月 11 日 17:00 作为调度目标时刻，对 2023 年 10 月 11 日当天的订单数据进行搜索之后，得到有接取订单记录的车辆共 2411 辆。

2. 车辆是否有司机驾驶

设置共享无人车占车队中车辆总数的比例为 50%，在 Python 中调用 random 函数随即生成。

3. 车辆是否空驶

在车队调度中，只有当前时间为空驶状态的车辆才能被分配调度任务。在本研究中，基于出租车历史行程，选择调度开始时刻，为该时刻的每辆车添加一个状态标记。首先确定调度开始的时刻，由于调度时间为 30 分钟，调度目标时刻为 2023 年 10 月 11 日

17:00，因此调度开始时刻确定为2023年10月11日16:30。然后按照出租车的ID遍历每辆出租车历史订单中出发时刻和结束时刻的时间戳，若调度开始时刻时间戳“2023-10-11 16:30”处于单条订单的出发时刻和结束时刻之间，则将该车辆标记为“载客”，否则，则将该车辆标记为“空闲”。

经过标记，在全部的2411辆出租车中，共有1396辆出租车处于“空闲”状态，其他车辆为“载客”状态。

4. 当前所在站点编号

结合上一步所确定的“车辆是否空驶”状态，进一步确认车辆所在站点编号。若车辆标记为“载客”，则将车辆所在站点编号赋为0（在5.1.1节交通小区划分中，最终确定的小区编号为1到22的整数，因此将车辆所在站点编号赋为0即为车辆未停放在任何站点）；若车辆标记为“空闲”，车辆所在站点编号即为在调度开始时刻前，该车辆的最后一次订单的下客点所在区域编号。经过统计，各交通小区空闲车辆数如图5.6所示。

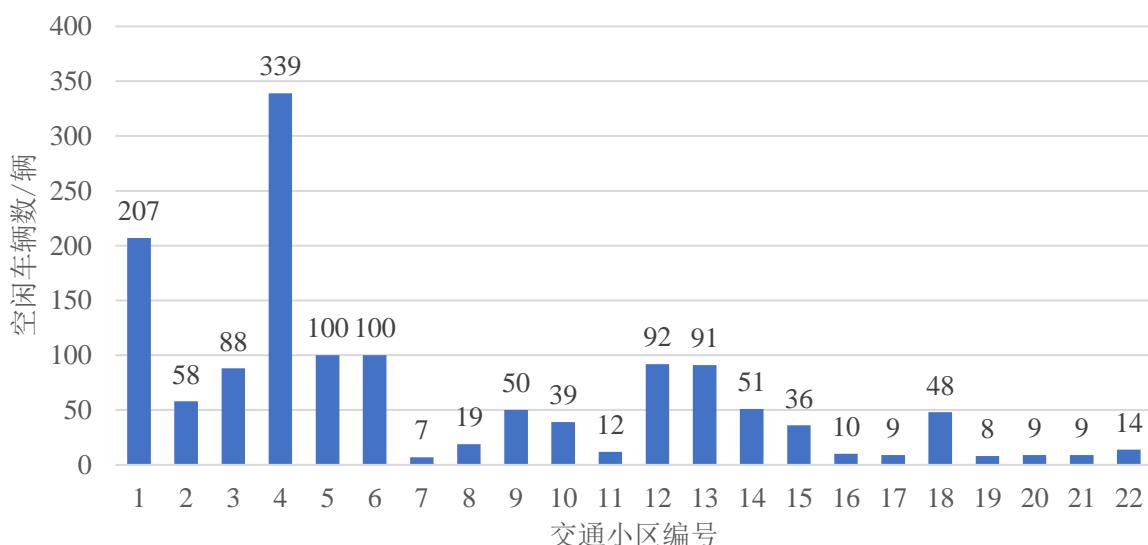


图5.6 各交通小区空闲车辆数

5. 当前所在路网节点编号

对于真实路网上的仿真，需要为车辆匹配路网节点。在本研究中，运用KD树算法^[68]将车辆当前所在位置匹配到最近的路网节点。

KD树是一种二叉树，通过空间分割和递归查找的方式，有效地处理多维空间中的近邻搜索问题，特别适用于多维空间中的关键数据检索，如范围搜索和最近邻搜索。KD树中的空间分割过程，为先用分割面把整个空间按照左右和上下进行一定次数的等分，再将分割后的小空间以二叉树形式进行记录；KD树中的递归查找过程，则是从树的根节点开始，沿着分割面递归地查找最近邻，一旦找到一个最近邻，就停止搜索并返回该

最近邻。这样的处理方式使得 KD 树在处理大型图网络时能够保持较高的搜索效率，避免了全局空间搜索，从而大大减少计算量，提高匹配速度。

匹配到空闲车辆的路网节点如图 5.7 所示。可以看出，在 1 号、4 号、6 号、10 号交通小区中，匹配到空闲车辆的路网节点数量较多。在 2.1.1 节介绍数据集时，绘制了芝加哥数据集中上下客点的分布图(图 2.1)。由于原始数据中上下客点的分布较为集中，因此在匹配完成后，只有 121 个路网节点匹配到了空闲车辆。

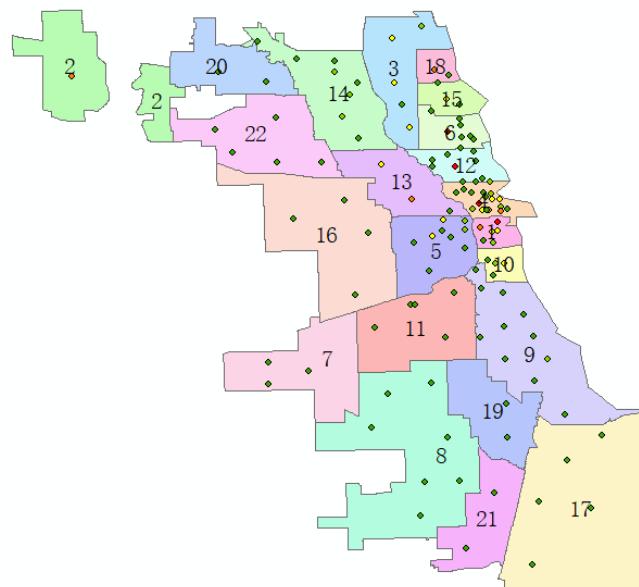


图 5.7 匹配到空闲车辆的路网节点

在全部 121 个匹配到空闲车辆的路网节点中，有 90 个节点停放的空闲车辆数在 1 至 11 辆之间。而部分节点匹配到的车辆数在 70 辆以上。由于原始数据中部分区域只存在 5 个以内的上下客点位，但对应的空闲车辆数达到 100 辆以上。在就近匹配的时候，算法只能将大量的空闲车辆匹配至最近的一个路网节点。

5.1.4 调度模型实例分析其他相关参数

1. 成本标定

对于平台来说，调度产生的成本主要包括车辆的能源消耗成本、车辆折损成本和平台对司机的激励成本；对于司机来说，调度产生的成本主要包括车辆的能源消耗成本、车辆折损成本、调度过程中产生的时间感知成本。

(1) 车辆成本。车辆的能源消耗成本、车辆折损成本为所有参与调度的车辆都会产生的成本。一般情况下，车辆的折损成本与折旧率有关。折旧率可以按照每行驶一定里程后车辆的价值减少的百分比来计算。车辆的能源消耗成本则一般用每公里燃油费用乘以行驶里程。

(2) 司机成本。在实际调度中,对司机的激励成本应当与司机对时间的感知成本挂钩。在运营期间,出租车司机会因长时间未接单产生负面情绪。在运营状态下,司机需要付出出租车成本和人工成本,因此司机需要服务乘客以兑换经济价值。当司机认为时间能够带来价值时,就会产生一种不能浪费时间的认知。而在调度期间司机处于空驶状态,没有得到相应的经济收入,会产生时间焦虑感,而对时间焦虑感起重要作用的是时间价值预期。对于司机来说,时间价值预期则与载客状态下该时间内产生的净收益相关。如果能以奖励的方式对完成调度任务的司机予以激励,能提高司机的调度意愿。

由于成本受车辆型号和行驶状态等因素的影响,难以准确标定,而本研究主要针对的是共享无人车和有人驾驶的出租车的调度问题,因此在进行成本标定时能将共享无人车和有人驾驶的出租车区分开即可。其他学者关于网约车订单定价^[69]以及出租车司机调度^[70]的相关研究中均提及出租车辆的运营成本及计算方式,根据前面对成本的分析以及相关研究,最终确定每公里调度成本为 2,对于有司机驾驶的出租车辆,增加每分钟调度成本 0.5,不对成本的单位另做定义。

2. 时间标定

在分配调度车辆并生成路径时,调度的时间限制应与城市大小、交通小区的划分尺寸、城市当前各路段的通行时间以及订单预测任务中所选择的目标时间点等因素有关。在第三章对纽约曼哈顿的出租车订单预测中,将时间间隔为 15 分钟以及 30 分钟的预测结果进行对比,发现以 15 分钟为间隔进行时间切片并预测时,出租车订单量的预测精度有所降低,但总体来看测试集损失值仍处于较低水平。而芝加哥订单预测即使在时间间隔为 30 分钟的情况下也远高于纽约曼哈顿数据集上的预测损失值。

基于订单预测工作以及预调度对出租车供需匹配的作用,结合订单预测精度以及 5.1.1 节分析的交通小区质心间直线距离的分布情况及形成初步方案时的距离限制,对于在纽约曼哈顿数据上进行的调度实验,将 15 分钟作为调度任务最大用时;芝加哥调度的最大用时则确定为 30 分钟。

随着调度时间的增加,司机随之产生的时间焦虑也会上升,因此除了确定整体调度时间以外,还应对司机的调度时间进一步限制。在纽约曼哈顿调度方案的生成中,司机参与单次调度的时间限制在 10 分钟以内;在芝加哥调度方案的生成中,司机参与单次调度的时间限制在 15 分钟以内。

5.2 出租车区域调度结果分析

5.2.1 纽约曼哈顿出租车区域调度方案分析

在 5.1 节参数标定部分便提到, 由于纽约曼哈顿出租车订单数据集中缺少上下车经纬度, 无法匹配路网节点, 因此在计算调度方案时对模型进行了简化。

简化后, 同样分两步求得调度方案, 但是在输出具体调度方案 Y 时, Y 中的每个元组 (i, j, k) 的含义有所不同, 其中 i 表示调出区域标号, j 表示调入区域编号, k 表示调度车辆编号。由于未引入真实路网, 因此车辆从交通小区 i 调度至交通小区 j 的行程距离即为两区域质心间的直线距离。在 5.1.1 节中定义质心间移动速度为 20 公里/小时, 因此, 从交通小区 i 调度至交通小区 j 的行程时间为直线距离/ 20×60 分钟。

对模型求解后, 得到的初步调度方案如表 5.2 所示。

表 5.2 初步调度方案 (部分)

调出区域编号	调入区域编号	调度车辆数	调度距离
1	44	4	2.671058
9	39	7	1.125751
9	52	8	0.547372
10	39	11	1.729202
10	52	4	1.16549
11	39	10	2.154663
13	58	8	2.909152
14	39	7	3.451716
17	39	5	2.388051
17	45	3	0.789948
17	56	1	0.589686
18	39	7	1.060394
20	39	7	2.417774
20	56	8	0.66625
21	39	9	3.156738
21	56	6	0.92157
22	39	5	3.694165
28	39	2	1.958439
28	40	1	1.840101
28	44	5	0.829713

经过求解, 参与调度的车辆共有 332 辆, 其中最远调度距离为从 22 号交通小区调度至 39 号交通小区, 调度距离约为 3.69 公里。在此初步调度方案的基础上求解具体调度方案, 部分调度任务如表 5.3 所示 (全部调度方案见附录 B 表 B1)。由于没有引入真实路网, 因此相对于初步方案, 具体方案只是进一步确定了调度车辆编号和车辆是否有司机驾驶。

表 5.3 具体调度方案（部分）

PUL	DOL	Taxi ID	K	dis
1	44	0	0	2.671058
1	44	201	0	2.671058
1	44	670	0	2.671058
1	44	737	0	2.671058
9	39	142	0	1.125751
9	39	276	0	1.125751
9	39	410	0	1.125751
9	39	477	0	1.125751
9	39	544	0	1.125751
9	39	611	0	1.125751
9	39	678	0	1.125751
10	39	143	0	1.729202
10	39	210	0	1.729202
10	39	277	0	1.729202
10	39	612	0	1.729202

在最终形成的调度方案中，调度总成本为 1261.58，参与调度车辆共有 332 辆，其中 98 辆是有司机驾驶的出租车，其余 234 辆均为共享无人车。在最初对车辆是否有司机驾驶进行标定时，有司机驾驶的出租车和共享无人车各占 50%，但在生成的调度方案中参与调度的共享无人车辆数为有人驾驶的出租车辆数的 2.39 倍。从调度距离看，司机参与的调度任务中最大调度距离约为 1.78 公里，平均调度距离约为 0.90 公里；而共享无人车调度的最大距离为约为 3.69 公里，平均调度距离约为 1.99 公里。在用本文构建的区域再平衡调度模型生成调度方案时，能够有效地减少司机参与调度的数量以及单个司机调度的行程距离。

按照初步调度方案完成调度后，各个交通小区的出租车辆供给数和实际需求数对比如图 5.8 所示。

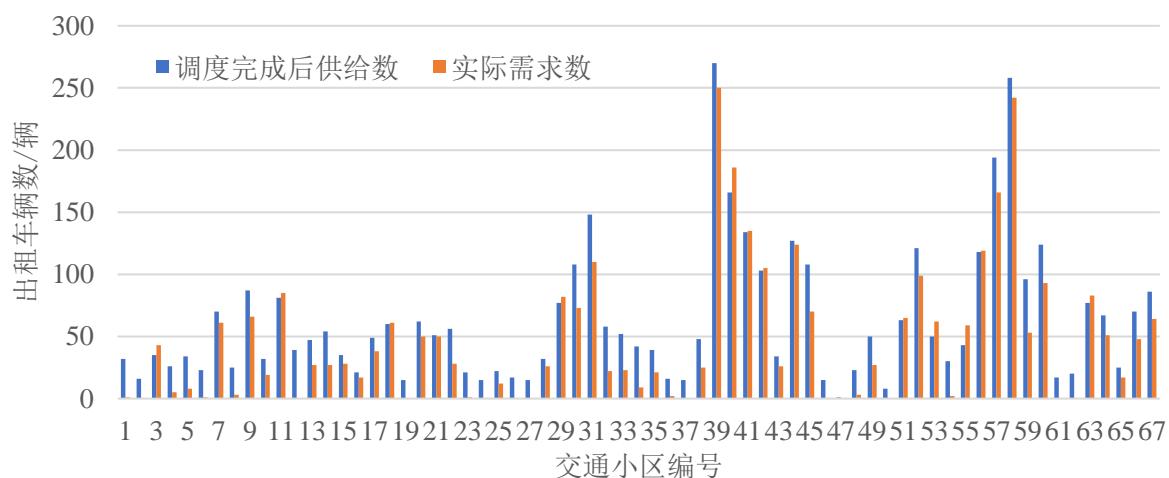


图 5.8 调度后各交通小区出租车辆供给数和实际需求数

按照计算出的调度方案完成调度后，对于全部 67 个交通小区，有 54 个交通小区的出租车辆供给能够满足该小区内的需求，而存在 13 个交通小区无法满足需求，产生的总缺口为 78 辆。其中供给缺口最大的小区为 40 号小区，按照调度方案调度后，区域出租车供给数仍比实际需求数少 20 辆。

绘制各交通小区实际订单量与预测订单量比较图，如图 5.9 所示。

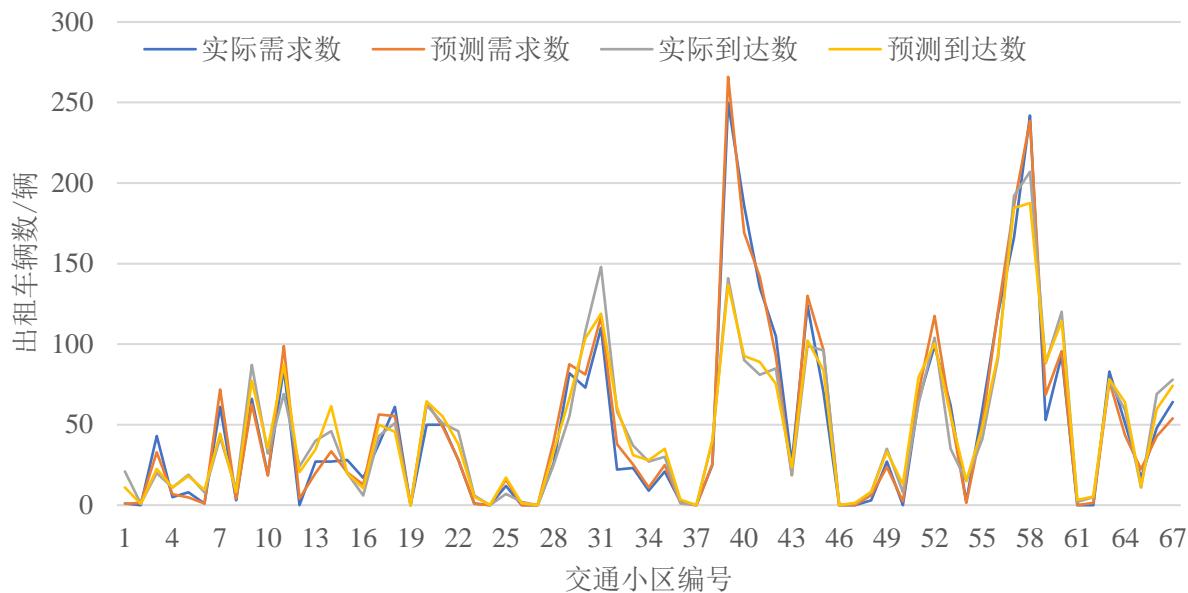


图 5.9 各交通小区实际订单量与预测订单量比较

可以看出，在订单预测阶段，部分区域需求量预测值偏低而即将到达量预测值偏高，因此将预测结果输入下阶段的车辆调度模型进行计算后，得到的调度方案没有完全满足区域需求。但总体来看，在该实例中通过区域预调度，全部区域产生的车辆总缺口由 302 辆降低为 78 辆，且供给缺口的分布在区域间更加均匀。在此基础上能够更高效地进行派单调度，提高服务水平。

设置共享无人车占比为 25% 和 75% 时，能够成功生成新的调度方案以满足预测需求，但具体的调度方案与调度总成本有所不同，对比如表 5.4 所示。

表 5.4 不同共享无人车占比下调度方案对比

车队中的共享无人车占比	调度车辆中的共享无人车占比	调度总成本
25%	48.58%	1418.87
50%	70.48%	1261.58
75%	88.18%	1123.10

对比共享无人车占比 50% 时调度花费的总成本，车队中的共享无人车占比为 25% 时，调度总成本增加了 12.47%；车队中的共享无人车占比为 25% 时，调度总成本减少了 10.98%。随着车队中的共享无人车占比提高，计算的调度方案中共享无人车的占比会相

应提高，同时调度所花费的总成本会有所降低。

5.2.2 芝加哥区域调度方案分析

通过模型第一步求解，得到初步调度方案如表 5.5 所示，在调度过程中，参与调度的车辆共有 35 辆。调度完成后区域出租车供给量与实际需求量对比如图 5.10 所示。

表 5.5 初步调度方案

调度起点	调度终点	调度车辆数
1	10	18
14	2	6
20	2	7
22	2	4

调度完成后区域出租车供给量与实际需求量对比如图 5.10 所示。从图中可以看出，除 2 号小区和 7 号小区外，所有交通小区的出租车供给量均大于出租车需求量，在下个时段能够满足区域内的出租车出行需求。

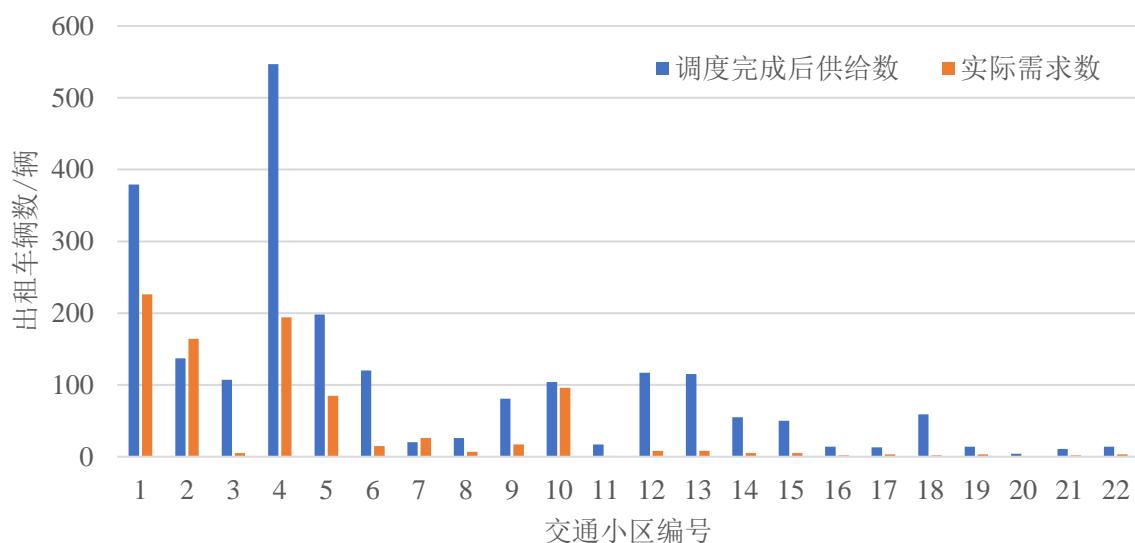


图 5.10 调度后各交通小区出租车供给数和实际需求数

与纽约曼哈顿类似，对 2 号交通小区和 7 号交通小区下时段的订单需求数以及即将到达量进行预测时，需求数偏低而到达数预测偏高，导致调度方案没有满足需求。

求解得到的调度方案如表 5.6 所示(全部调度方案见附录 B 表 B2)。在调度过程中，产生的调度总成本为 181.46。参与调度的 35 各车辆中，2 辆为有人驾驶的出租车，33 辆为共享无人车。在共享无人车在车队中同样占比 50%的前提下，相对于在纽约曼哈顿数据集基础上开展的实验，由于芝加哥整体空闲的出租车辆数偏多，因此在为车辆分配调度任务时可供选择的选项更多，对芝加哥的出租车辆进行调度时更有可能选择到共享无人车。

表 5.6 具体调度方案（部分）

调出 区域	调入 区域	出租车 编号	是否有 司机	起点 节点	终点 节点	路径	距离/ 公里	时间/ 分钟
1	10	742	0	116	1751	[116, 1752, 1870, 536, 1751]	0.77	2
1	10	1321	0	503	1751	[503, 535, 536, 1751] [1531, 756, 1631, 1630,	0.74	2
14	2	526	0	1531	2074	1388, 755, 754, 752, 751, 749, 489, 1591, 747, 2152, 529, 2074] [1953, 2137, 2135, 2134,	6.02	14
14	2	918	0	1953	69	2133, 1578, 2057, 308, 2129, 2128, 2130, 69] [1359, 1362, 1365, 1598,	5	11
20	2	1100	0	1359	1084	1530, 1599, 615, 613, 1701, 2066, 1084] [1368, 1603, 1602, 620,	4.06	9
20	2	703	0	1368	1084	1600, 615, 613, 1701, 2066, 1084] [1368, 1603, 1602, 620,	3.42	9
20	2	1371	1	1368	1084	1600, 615, 613, 1701, 2066, 1084]	3.42	9
22	2	228	0	748	2158	[748, 2075, 2153, 529, 2158]	1.61	4
22	2	828	0	752	2074	[752, 751, 749, 489, 1591, 747, 2152, 529, 2074]	3.74	9
22	2	114	0	1980	2158	[1980, 783, 782, 2159, 2157, 2158]	2.74	6

附注：行程时间不是按照邻接矩阵计算的真实时间，而是向上取整为整分钟。

根据仿真实验结果，可以认为在第三章中建立的出租车短时订单需求模型，以及第四章建立的车队调度模型的基础上，能够在时间限制内，通过车辆区域间预调度的方式，进一步满足区域间差异化的订单需求，使系统调度成本最小化，并尽可能地减少有司机驾驶的出租车参与调度任务所花费的时间和行程距离。

在本文的实例分析中，因为数据集中的信息限制，无法完全贴合实际的出租车辆调度场景。需要结合实际情况对模型涉及的参数的标定方法进行完善，例如对每辆出租车、无人车的成本单独计算，得到差异化的价值函数，能够更加有效地指导调度。

5.3 本章小结

本章以纽约曼哈顿和芝加哥两地为例，对本文建立的出租车区域再平衡调度模型开展仿真试验。在对纽约曼哈顿进行实例分析时，由于缺少经纬度、车辆 ID 等信息，对模型进行了简化。在设置芝加哥相关参数时，引入真实路网，利用路网中的字段信息建

立距离邻接矩阵和速度邻接矩阵。根据订单数据集中的出租车 ID 确定车队的整体规模以及各车辆的运营状态，然后运用 KD 树算法将车辆匹配至最近的路网节点。采用 Dijkstra 算法对真实路网上的最短路径进行搜索，最后用求解器对模型进行求解并输出结果。结果表明，基于本文搭建的订单预测模型以及区域调度模型实施调度方案，能够使车辆在时间限制内完成调度任务，有效满足各区域的用车需求，并减少司机的调度距离和调度时间。

总结与展望

总结

随着自动驾驶技术的逐步推进，结合出租车服务和自动驾驶技术的共享自动驾驶服务应运而生。本文在此背景下，基于出租车短时订单需求预测，针对有人驾驶的出租车和共享无人车组成的车队构建区域再平衡调度模型，以更好地适应不同时间段尤其是高峰时段的出租车订单需求。

首先对研究选取的两个出租车订单数据集——芝加哥出租车订单数据集和纽约黄色出租车订单数据集进行介绍、词条筛选和异常数据处理，并对订单数据的时空特征进行分析。随后对出租车订单的短时需求预测任务进行研究，将城市划分为多个区域，以一定的时间间隔将时间划分为多个时间切片，针对每个区域内出租车出发量和到达量，用已经观测到的时间切片内的出租车历史行程，预测下一个时间切片该区域的出租车出发数量和到达数量。引入 LSTM 模型、TCN 模型，并尝试在模型中添加时序模式注意力机制，采用网格搜索的方式确定最优超参数。通过分析预测结果，认为在本研究中涉及的出租车订单量预测问题中，使用 Huber 损失训练时，模型的收敛速度最快；引入时序模式注意力机制的 TCN 模型在预测精度、模型稳定性和模型泛化等方面表现较优。

在出租车订单需求短时预测的基础上，构建出租车区域供需再平衡调度模型。考虑有司机驾驶的出租车和共享无人车产生的不同调度成本，在满足区域车辆需求的基础上，使系统整体调度费用最低。在建立模型时，考虑到大规模车队调度问题的求解难度，在建立区域调度模型时将车辆调度方案的求解分为两阶段，第一阶段求解调出区域、调入区域和调度车辆数，第二阶段根据第一阶段的求解结果，进一步解出调度车辆标号、调入节点、调出节点和调度路径。

最后以纽约曼哈顿和芝加哥为例对本文建立的出租车区域供需再平衡调度模型进行实例分析。在模型求解过程中，采用 K-means 聚类算法重新划分交通小区，采用 KD 树算法将车辆匹配至最近的路网节点，采用 Dijkstra 算法对真实路网上车辆的行程路径进行搜索，最后引入 OR-Tools 开源求解器对模型进行求解并输出结果。结果表明，采用本文搭建的调度模型，在短时需求预测的基础上对车队进行区域预调度，能够使车辆在时间限制内完成调度任务，有效满足各区域的用车需求，并有效区分共享无人车和有司机驾驶的出租车，使系统调度成本尽可能小的同时限制了司机的调度时间和距离。

展望

对于本文所提出的出租车订单需求预测方法以及混合车队的调度方法，未来可以从以下几方面进行改进：

1. 本文主要基于出租车订单的时间序列对区域出租车订单量进行预测，在处理空间特征时，只是将各分区的订单数据共同输入模型进行训练，这种预测方式虽然能够学习到不同分区之间订单量存在的关联，但未结合实际的路网结构等信息。在下一步研究中，可以通过引入图结构以及城市 POI 等信息，进一步挖掘空间特征，提高出租车订单的预测精度。
2. 本文基于实际的城市出租车运营数据确定车队规模，未考虑不同车队规模下出租车与共享无人车服务的供给需求匹配效率。因此在未来的研究中可以基于高峰时期的区域出租车需求，对混合车队的规模进行测算，在满足出行需求的基础上提高运营商的收益率。

参考文献

- [1] 张华,马苗僖.平台模式下的类出租车服务治理挑战与新思路[J/OL].综合运输,1-11,
<http://kns.cnki.net/kcms/detail/11.1197.U.20231116.1419.002.html>,2023-11-17.
- [2] 王睿.基于车联网环境的共享自动驾驶汽车调度研究[D].大连理工大学,2019.
- [3] Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting[A]. IJCAI International Joint Conference on Artificial Intelligence[C]. Stockholm, Sweden: International Joint Conferences on Artificial Intelligence, 2018: 3634-3640.
- [4] Bai M, Lin Y, Ma M, et al. Travel-time prediction methods: A review[A]. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)[C]. Tokyo, Japan: Springer Verlag, 2018: 67-77.
- [5] 崔建勋,要甲,赵泊媛.基于深度学习的短期交通流预测方法综述[J].交通运输工程学报,2024,24(02):50-64.
- [6] Fan W, Machemehl R B, Lownes N E. Carsharing. Carsharing Dynamic Decision-Making Problem for Vehicle Allocation[J]. Transportation Research Record: Journal of the Transportation Research Board, 2008, 2063(1): 97-104.
- [7] Chan K Y, Dillon T S, Singh J, et al. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm[J]. IEEE Transactions on Intelligent Transportation Systems, 2011, 13(2): 644-654.
- [8] GE S Y, ZHENG C J, HOU M M. Forecast of Bus Passenger Traffic Based on Exponential Smoothing and Trend Moving Average Method[J]. Applied Mechanics and Materials, 2013, 433-435: 1374-1378.
- [9] 杨兆升,朱中.基于卡尔曼滤波理论的交通流量实时预测模型[J].中国公路学报,1999,12(3):63-67.
- [10] Cai L, Zhang Z, Yang J, et al. A noise-immune Kalman filter for short-term traffic flow forecasting[J]. Physica A: Statistical Mechanics and its Applications, 2019, 536: 122601.
- [11] Ahmed M S, Cook A R. Analysis of freeway traffic time-series data by using Box-Jenkins techniques[M]. 1979, 722: 1-9.
- [12] Li X L, Pan G, Wu Z H, et al. Prediction of urban human mobility using large-scale taxi traces and its applications[J]. Frontiers of Computer Science, 2012, 6(1): 111-121.
- [13] 韦凌翔,陈红,王永岗,等.基于 RVM 和 ARIMA 的短时交通流量预测方法研究[J].武汉

- 理工大学学报(交通科学与工程版),2017,41(2):349-354.
- [14] 叶锋,欧阳智超,陈威彪,等.基于机器学习的智能出租车预测系统[J].计算机系统应用,2018,27(9):61-67.
- [15] Liu Z, Chen H, Sun X, Chen H. Data-Driven Real-Time Online Taxi-Hailing Demand Forecasting Based on Machine Learning Method[J]. Applied Sciences, 2020, 10(19): 6681.
- [16] Chen Z, Zhao B, Wang Y, et al. Multitask Learning and GCN-Based Taxi Demand Prediction for a Traffic Road Network[J]. Sensors, 2020, 20(13): 3776.
- [17] Liu L, Qiu Z, Li G, et al. Contextualized Spatial-Temporal Network for Taxi Origin-Destination Demand Prediction[J]. IEEE Transactions on Intelligent Transportation Systems. 2019, 20(10): 3875-3887.
- [18] Wang Y, Wo T, Yin H, et al. Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling[A]. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. Anchorage, AK, United states: Association for Computing Machinery, 2019: 1227-1235.
- [19] Kim T, Sharda S, Zhou X, et al. A stepwise interpretable machine learning framework using linear regression (LR) and long short-term memory (LSTM): City-wide demand-side prediction of yellow taxi and for-hire vehicle (FHV) service[J]. Transportation Research Part C: Emerging Technologies, 2020, 120: 102786.
- [20] 吴海龙.基于强化学习的不同合乘意愿下出租车调度方法[D].哈尔滨工业大学, 2021.
- [21] 张宗垚,孟康,董龙琦.机场出租车司机选择策略与出租车调度设计[J].信息技术与信息化, 2020(07):66-68.
- [22] Sadrani M, Tirachini A, Antoniou C. Vehicle dispatching plan for minimizing passenger waiting time in a corridor with buses of different sizes: Model formulation and solution approaches[J]. European Journal of Operational Research, 2022, 299(1): 263-282.
- [23] Kim B, Kim J, Huh S, et al. Multi-objective predictive taxi dispatch via network flow optimization[J]. IEEE Access, 2020, 8: 21437-2145.
- [24] Long J, Szeto W Y, Du J, et al. A dynamic taxi traffic assignment model: A two-level continuum transportation system approach[J]. Transportation Research Part B Methodological, 2017, 100(6): 222-254.
- [25] Yang Y, Wang X, Xu Y, et al. Multiagent Reinforcement Learning-Based Taxi Pre-dispatching Model to Balance Taxi Supply and Demand[J]. Journal of advanced transportation, 2020, 2020.
- [26] 王鑫涛.基于多智能体强化学习的出租车调度方法研究[D].吉林大学, 2021.

- [27] Li M, Zhi W, Qin, et al. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning[A]. The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019[C]. San Francisco, CA, United states: Association for Computing Machinery, 2019: 983-994.
- [28] Jin J, Zhou M, Zhang W, et al. CoriDE: Joint order dispatching and fleet management for multi-scale ride-hailing platforms[A]. International Conference on Information and Knowledge Management, Proceedings[C]. Beijing, China: Association for Computing Machinery, 2019: 1983-1992.
- [29] Xu Z, Li Z, Guan Q, et al. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach[A]. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. London, United kingdom: Association for Computing Machinery, 2018: 905-913.
- [30] Tang X, Qin Z, Zhang F, et al. A deep value-network based approach for multi-driver order dispatching[A]. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. Anchorage, AK, United states: Association for Computing Machinery, 2019: 1780-1790.
- [31] Lin K, Zhao R, Xu Z, et al. Efficient large-scale fleet management via multi-agent deep reinforcement learning[A]. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. London, United kingdom: Association for Computing Machinery, 2018: 1774-1783.
- [32] Yu Z, Hu M. Deep Reinforcement Learning with Graph Representation for Vehicle Repositioning[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(8): 13094-13107.
- [33] Chen H, Guo B, Yu Z, et al. The Framework of Increasing Drivers' Income on the Online Taxi Platforms[J]. IEEE Transactions on Network Science and Engineering, 2020, 7(4): 2182-2191.
- [34] Mitchell W J, Borroni-Bird C E, Burns L D. Reinventing the automobile: personal urban mobility for the 21st century[M]. MIT Press, Cambridge, 2010.
- [35] Iglesias R, Rossi F, Zhang R, et al. A BCMP network approach to modeling and controlling autonomous mobility-on-demand systems[J]. Journal of Robotics Research, 2019, 38(2): 357–374.
- [36] Rossi F, Zhang R, Hindy Y, et al. Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms[J]. Autonomous Robots, 2018, 42(7): 1427–1442.

- [37] Alonso-Mora J, Samaranayake S, Wallar A, et al. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment[J]. National Academy of Sciences, 2017, 114(3): 462–467.
- [38] Levin M W, Kockelman K M, Boyles S D, et al. A general framework for modeling shared autonomous vehicles with dynamic network-loading and dynamic ride-sharing application[J]. Computers, Environment and Urban Systems, 2017, 64: 373–383.
- [39] Fiedler D, Erticky M, Alonso-Mora J, et al. The Impact of Ridesharing in Mobility-on-Demand Systems: Simulation Case Study in Prague[A]. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC[C]. Maui, HI, United states: Institute of Electrical and Electronics Engineers Inc., 2018: 1173-1178.
- [40] Zhang R, Pavone M. Control of robotic mobility-on-demand systems: a queueing - theoretical perspective[J]. International Journal of Robotics Research, 2014, 35(1): 186-203.
- [41] Chuah S P, Xiang S, Wu H. Optimal rebalancing with waiting time constraints for a fleet of connected autonomous taxi[A]. IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings[C]. Singapore, Singapore: Institute of Electrical and Electronics Engineers Inc., 2018: 629-634.
- [42] Rossi F, Iglesias R, Alizadeh M, et al. On the Interaction between Autonomous Mobility-on-Demand Systems and the Power Network: Models and Coordination Algorithms[J]. IEEE Transactions on Control of Network Systems, 2020, 7(1): 384-397.
- [43] Shi J, Gao Y, Wang W, Yu N, et al. Operating Electric Vehicle Fleet for Ride-Hailing Services with Reinforcement Learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 21(11): 4822-4834.
- [44] Kim S, Lee U, Lee I, et al. Idle vehicle relocation strategy through deep learning for shared autonomous electric vehicle system optimization[J]. Journal of Cleaner Production, 2022, 333: 130055.
- [45] Qian T, Shao C, Wang X, et al. Shadow-Price DRL: A Framework for Online Scheduling of Shared Autonomous EVs Fleets[J]. IEEE Transactions on Smart Grid, 2022, 13(4): 3106 -3117.
- [46] Bischoff J, Maciejewski M. Simulation of City-wide Replacement of Private Cars with Autonomous Taxis in Berlin[J]. Procedia Computer Science, 2016, 83: 237-244.
- [47] Spieser K, Treleaven K, Zhang R, et al. Toward a Systematic Approach to the Design and Evaluation of Automated Mobility-on-Demand Systems: A Case Study in Singapore[J]. In Road Vehicle Automation, 2014: 229-245.

- [48] Li X, Ma J, Cui J, et al. Design framework of large-scale one-way electric vehicle sharing systems: A continuum approximation model[J]. Transportation Research Part B: Methodological, 2016, 88: 21-45.
- [49] Levin M W, Kockelman K M, Boyles S D, et al. A general framework for modeling shared autonomous vehicles with dynamic network-loading and dynamic ride-sharing application [J]. Computers, Environment and Urban Systems, 2017, 64: 373-383.
- [50] Taxi trips reported to the City of Chicago [DB/OL], <https://data.cityofchicago.org/Transportation/Taxi-Trips>.
- [51] New York City Taxi and Limousine Commission (TLC) Trip Record Data[DB/OL], <https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>.
- [52] Census tract boundaries in Chicago [DB/OL], <https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Census-Tracts-2010/5jrd-6zik>.
- [53] Liu L, Qiu Z, Li G, et al. Ouyang and L. Lin, Contextualized Spatial–Temporal Network for Taxi Origin-Destination Demand Prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(10): 3875-3887.
- [54] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
- [55] Understanding LSTM Networks[DB/OL], <https://colah.github.io/posts/2015-08-Understanding-LSTMs>.
- [56] Bai S, Kolter J Z, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[J]. arXiv: 1803.01271, 2018.
- [57] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30: 6000–6010.
- [58] Li C, Hou Y, Wang P et al. Joint distance maps based action recognition with convolutional neural networks[J]. IEEE Signal Processing Letters, 2017, 24(5): 624-628.
- [59] Hu D. An introductory survey on attention mechanisms in NLP problems[A]. Advances in Intelligent Systems and Computing[C]. London, United kingdom: Springer Verlag, 2020: 432-448.
- [60] Shih S Y, Sun F K, Lee H. Temporal pattern attention for multivariate time series forecasting[J]. Mach Learn, 2019, 108: 1421–1441.
- [61] 余佩.基于出行需求的无人驾驶出租车优化调度与部署研究[D].浙江工商大学,2020.
- [62] MAC Q j. Some methods for classification and analysis of multivariate observations[M]. Berkeley: Berkeley Symposium on Mathematical Statistics and Probability, 1967.

- [63] 李雪峰.混合需求条件下城市网约车需求预测及派单方法研究[D].东南大学,2021.
- [64] 李凯文,张涛,王锐,等.基于深度强化学习的组合优化研究进展[J].自动化学报,2021,47(11):2521-2537.
- [65] OR-Tools Reference [DB/OL], <https://developers.google.cn/optimization>.
- [66] Transportation Networks in Chicago [DB/OL], <https://github.com/bstabler/Transportation-Networks/tree/master/chicago-regional>.
- [67] Dijkstra E.W. A note on two problems in connexion with graphs[J]. Numerische Mathematik, 1959, 1: 269–271.
- [68] Bentley J. L. Multidimensional binary search trees used for associative searching[J]. Communications of the ACM, 1975, 18 (9): 509–517.
- [69] 陈豪.重庆市网约车定价方法研究[D].重庆交通大学,2022.
- [70] 刘梦寒.基于司机决策行为的机场出租车供需均衡和调度优化研究[D].北京交通大学,2023.

附录 A

```
#区域调度模型求解代码
from ortools.sat.python import cp_model

# 求解初步方案
def solve_initial_dispatch(l, U, D, O, dis, max_dis):
    model = cp_model.CpModel()

    # 定义变量 X, 表示从交通小区 p 向交通小区 q 调度的车辆数
    X = {}
    for p in range(l):
        for q in range(l):
            X[(p, q)] = model.NewIntVar(0, U[p], f'X_{p}_{q}'')

    # 约束 1: 每个交通小区调出的车辆数不超过其空闲车辆数
    for p in range(l):
        model.Add(sum(X[(p, q)] for q in range(l)) <= U[p])

    # 约束 2: 每个交通小区的调度后车辆数不小于调度前的空闲车辆数
    for p in range(l):
        model.Add(U[p] + D[p] - O[p] -
                  sum(X[(p, q)] for q in range(l)) +
                  sum(X[(q, p)] for q in range(l)) >= 0)

    # 约束 3: 交通小区之间的调度距离不能超过最大距离 max_dis
    for p in range(l):
        for q in range(l):
            if dis[p][q] > max_dis:
                model.Add(X[(p, q)] == 0)

    # 定义目标函数: 全部车辆在质心间移动距离最短
    total_dis = sum(X[(p, q)] * dis[p][q] for p in range(l) for q in range(l))
```

```

model.Minimize(total_dis)

solver = cp_model.CpSolver()
status = solver.Solve(model)

if status == cp_model.OPTIMAL or status == cp_model.FEASIBLE:
    result = [[0 for _ in range(l)] for _ in range(l)]
    for p in range(l):
        for q in range(l):
            result[p][q] = solver.Value(X[(p, q)])
    return result

# 求解最终方案

def solve_vehicle_dispatch(X, l, n, m, s, v, b, T, h, max_time):
    model = cp_model.CpModel()

    # 构建调度方案 Y
    Y = []
    for p in range(l):
        for q in range(l):
            if X[p][q] > 0:
                for i, j in range(n):
                    if s[i] == p and s[j] == q:
                        for k in range(m):
                            if v[k] == i: # 约束 2: 车辆 k 能否从节点 i 开始调度
                                Y.append((i, j, k))

    # 定义变量 Y_vars[(i, j, k)]表示调度方案中的调度元组是否被选中
    Y_vars = {}
    for i, j, k in Y:
        Y_vars[(i, j, k)] = model.NewBoolVar(f'Y_{i}_{j}_{k}')

    # 约束 1: 相同的 k 值最多出现 b[k]次

```

```
E = {}  
for i, j, k in Y:  
    if k not in E:  
        E[k] = model.NewIntVar(0, 1, fD_{k}')  
        model.Add(E[k] <= b[k])  
  
# 约束 3：全部车辆完成调度时的用时不超过 max_dispatch_time  
for i, j, k in Y:  
    model.Add(T[i][j] <= max_dispatch_time)  
  
# 约束 4：通过调度成本约束司机调度时间  
cost = sum(Y_vars[(i, j, k)]*h[i][j][k] for i, j, k in Y)  
# 最小化调度成本  
model.Minimize(cost)  
  
solver = cp_model.CpSolver()  
status = solver.Solve(model)  
  
if status == cp_model.OPTIMAL or status == cp_model.FEASIBLE:  
    final_solution = {}  
    for i, j, k in Y:  
        final_solution[(i, j, k)] = solver.Value(Y_vars[(i, j, k)])  
    return final_solution
```


附录 B

表 B1 纽约曼哈顿具体调度方案

调出 区域	调入 区域	车辆 编号	有无 司机	调度距离	调出 区域	调入 区域	车辆 编号	有无 司机	调度距离
1	44	0	0	2.671058	11	39	747	0	2.154663
1	44	201	0	2.671058	11	39	814	0	2.154663
1	44	670	0	2.671058	11	39	881	0	2.154663
1	44	737	0	2.671058	11	39	948	0	2.154663
9	39	142	0	1.125751	13	58	280	0	2.909152
9	39	276	0	1.125751	13	58	548	0	2.909152
9	39	410	0	1.125751	13	58	615	0	2.909152
9	39	477	0	1.125751	13	58	682	0	2.909152
9	39	544	0	1.125751	13	58	749	0	2.909152
9	39	611	0	1.125751	13	58	816	0	2.909152
9	39	678	0	1.125751	13	58	883	0	2.909152
9	52	8	1	0.547372	13	58	950	0	2.909152
9	52	75	1	0.547372	14	39	13	0	3.451716
9	52	209	1	0.547372	14	39	80	0	3.451716
9	52	343	1	0.547372	14	39	147	0	3.451716
9	52	745	1	0.547372	14	39	415	0	3.451716
9	52	812	1	0.547372	14	39	616	0	3.451716
9	52	879	1	0.547372	14	39	817	0	3.451716
9	52	946	1	0.547372	14	39	884	0	3.451716
10	39	143	0	1.729202	17	39	351	0	2.388051
10	39	210	0	1.729202	17	39	619	0	2.388051
10	39	277	0	1.729202	17	39	686	0	2.388051
10	39	612	0	1.729202	17	39	753	0	2.388051
10	39	679	0	1.729202	17	39	820	0	2.388051
10	39	746	0	1.729202	17	45	83	1	0.789948
10	39	813	0	1.729202	17	45	150	1	0.789948
10	39	880	0	1.729202	17	45	418	1	0.789948
10	39	9	1	1.729202	17	56	284	1	0.589686
10	39	545	1	1.729202	18	39	17	0	1.060394
10	39	947	1	1.729202	18	39	352	0	1.060394
10	52	76	1	1.16549	18	39	486	0	1.060394
10	52	344	1	1.16549	18	39	553	0	1.060394
10	52	411	1	1.16549	18	39	821	0	1.060394
10	52	478	1	1.16549	18	39	888	0	1.060394
11	39	10	0	2.154663	18	39	955	0	1.060394
11	39	144	0	2.154663	20	39	19	0	2.417774
11	39	278	0	2.154663	20	39	86	0	2.417774
11	39	345	0	2.154663	20	39	153	0	2.417774
11	39	613	0	2.154663	20	39	622	0	2.417774
11	39	680	0	2.154663	20	39	756	0	2.417774

调出 区域	调入 区域	车辆 编号	有无 司机	调度距离	调出 区域	调入 区域	车辆 编号	有无 司机	调度距离
20	39	823	0	2.417774	29	40	631	0	1.777163
20	39	890	0	2.417774	29	40	698	0	1.777163
20	56	220	1	0.66625	29	40	765	0	1.777163
20	56	287	1	0.66625	29	40	966	0	1.777163
20	56	354	1	0.66625	30	40	96	0	1.565128
20	56	421	1	0.66625	30	40	230	0	1.565128
20	56	488	1	0.66625	30	40	364	0	1.565128
20	56	555	1	0.66625	30	40	498	0	1.565128
20	56	689	1	0.66625	30	40	699	0	1.565128
20	56	957	1	0.66625	30	40	967	0	1.565128
21	39	20	0	3.156738	30	58	29	1	0.538198
21	39	154	0	3.156738	30	58	163	1	0.538198
21	39	221	0	3.156738	30	58	297	1	0.538198
21	39	288	0	3.156738	30	58	431	1	0.538198
21	39	489	0	3.156738	30	58	565	1	0.538198
21	39	556	0	3.156738	30	58	632	1	0.538198
21	39	690	0	3.156738	30	58	766	1	0.538198
21	39	757	0	3.156738	30	58	833	1	0.538198
21	39	891	0	3.156738	30	58	900	1	0.538198
21	56	87	1	0.92157	31	39	97	0	1.765015
21	56	355	1	0.92157	31	41	30	0	1.077325
21	56	422	1	0.92157	31	41	365	0	1.077325
21	56	623	1	0.92157	31	41	432	0	1.077325
21	56	824	1	0.92157	31	41	633	0	1.077325
21	56	958	1	0.92157	31	41	767	0	1.077325
22	39	21	0	3.694165	31	41	834	0	1.077325
22	39	222	0	3.694165	31	41	968	0	1.077325
22	39	557	0	3.694165	31	41	164	1	1.077325
22	39	825	0	3.694165	31	41	231	1	1.077325
22	39	959	0	3.694165	31	41	298	1	1.077325
28	39	295	0	1.958439	31	41	499	1	1.077325
28	39	831	0	1.958439	31	41	566	1	1.077325
28	40	965	0	1.840101	31	41	700	1	1.077325
28	44	27	0	0.829713	31	41	901	1	1.077325
28	44	94	0	0.829713	32	39	165	0	2.163319
28	44	161	0	0.829713	32	39	299	0	2.163319
28	44	228	0	0.829713	32	39	366	0	2.163319
28	44	697	0	0.829713	32	39	433	0	2.163319
29	40	28	0	1.777163	32	39	634	0	2.163319
29	40	162	0	1.777163	32	39	768	0	2.163319
29	40	296	0	1.777163	32	39	835	0	2.163319
29	40	430	0	1.777163	32	39	969	0	2.163319
29	40	497	0	1.777163	32	41	31	1	1.538757

附录

调出 区域	调入 区域	车辆 编号	有无 司机	调度距离	调出 区域	调入 区域	车辆 编号	有无 司机	调度距离
32	41	98	1	1.538757	50	40	384	0	2.78808
32	41	232	1	1.538757	50	40	451	0	2.78808
32	41	500	1	1.538757	50	40	518	0	2.78808
32	41	567	1	1.538757	50	40	652	0	2.78808
32	41	701	1	1.538757	50	40	719	0	2.78808
32	41	902	1	1.538757	50	40	853	0	2.78808
35	7	34	0	1.735745	50	40	920	0	2.78808
35	7	302	0	1.735745	50	40	987	0	2.78808
35	7	503	0	1.735745	50	44	49	1	1.77797
35	7	637	0	1.735745	50	44	317	1	1.77797
35	7	905	0	1.735745	50	44	585	1	1.77797
35	7	972	0	1.735745	50	44	786	1	1.77797
38	39	37	0	3.671912	51	40	50	0	0.607319
38	39	104	0	3.671912	51	40	117	0	0.607319
38	39	238	0	3.671912	51	40	251	0	0.607319
38	39	573	0	3.671912	51	40	318	0	0.607319
38	39	640	0	3.671912	51	40	385	0	0.607319
38	39	707	0	3.671912	51	40	519	0	0.607319
38	39	774	0	3.671912	51	40	653	0	0.607319
45	39	111	0	1.632368	51	40	720	0	0.607319
45	39	245	0	1.632368	51	40	787	0	0.607319
45	39	312	0	1.632368	51	40	854	0	0.607319
45	39	513	0	1.632368	51	40	184	1	0.607319
45	39	647	0	1.632368	51	40	452	1	0.607319
45	39	915	0	1.632368	51	40	586	1	0.607319
47	40	113	0	1.974005	51	40	921	1	0.607319
47	40	180	0	1.974005	51	40	988	1	0.607319
47	40	247	0	1.974005	52	39	51	0	0.582398
47	40	448	0	1.974005	52	39	118	0	0.582398
47	40	582	0	1.974005	52	39	185	0	0.582398
47	40	850	0	1.974005	52	39	319	0	0.582398
47	29	46	1	0.566916	52	39	453	0	0.582398
47	29	314	1	0.566916	52	39	520	0	0.582398
47	29	381	1	0.566916	52	39	587	0	0.582398
47	29	515	1	0.566916	52	39	788	0	0.582398
47	29	649	1	0.566916	52	39	922	0	0.582398
47	29	716	1	0.566916	52	39	989	0	0.582398
47	29	783	1	0.566916	55	40	54	0	0.770237
47	29	917	1	0.566916	55	40	188	0	0.770237
47	29	984	1	0.566916	55	40	255	0	0.770237
50	40	116	0	2.78808	55	40	322	0	0.770237
50	40	183	0	2.78808	55	40	523	0	0.770237
50	40	250	0	2.78808	55	40	590	0	0.770237

调出 区域	调入 区域	车辆 编号	有无 司机	调度距离	调出 区域	调入 区域	车辆 编号	有无 司机	调度距离
55	40	657	0	0.770237	60	41	997	0	2.174595
55	40	791	0	0.770237	63	39	62	0	2.277962
55	40	858	0	0.770237	63	39	330	0	2.277962
55	40	925	0	0.770237	63	39	531	0	2.277962
55	40	456	1	0.770237	63	39	665	0	2.277962
55	40	724	1	0.770237	63	39	732	0	2.277962
55	40	992	1	0.770237	63	39	933	0	2.277962
56	42	256	0	1.019019	63	11	129	1	0.643255
56	42	390	0	1.019019	63	11	263	1	0.643255
56	42	792	0	1.019019	63	11	397	1	0.643255
57	58	56	0	1.501677	63	11	464	1	0.643255
57	58	190	0	1.501677	63	11	598	1	0.643255
57	58	391	0	1.501677	63	11	799	1	0.643255
57	58	592	0	1.501677	63	11	866	1	0.643255
57	58	659	0	1.501677	63	18	196	1	1.282946
57	58	726	0	1.501677	64	39	63	0	3.361374
57	58	793	0	1.501677	64	39	398	0	3.361374
57	7	123	1	0.754184	64	39	532	0	3.361374
57	7	257	1	0.754184	64	39	599	0	3.361374
57	7	458	1	0.754184	64	39	666	0	3.361374
57	7	860	1	0.754184	64	39	733	0	3.361374
57	7	927	1	0.754184	66	58	534	0	1.804299
57	7	994	1	0.754184	66	58	668	0	1.804299
59	7	862	0	1.204505	66	58	735	0	1.804299
59	41	259	0	3.057566	66	58	802	0	1.804299
59	41	527	0	3.057566	66	58	869	0	1.804299
59	41	594	0	3.057566	66	58	936	0	1.804299
59	41	728	0	3.057566	66	29	132	1	1.352746
59	41	795	0	3.057566	66	29	199	1	1.352746
59	41	929	0	3.057566	66	29	266	1	1.352746
60	41	59	0	2.174595	66	29	333	1	1.352746
60	41	126	0	2.174595	66	29	400	1	1.352746
60	41	193	0	2.174595	66	29	467	1	1.352746
60	41	260	0	2.174595	66	29	601	1	1.352746
60	41	327	0	2.174595	67	58	66	0	1.670502
60	41	394	0	2.174595	67	58	133	0	1.670502
60	41	528	0	2.174595	67	58	200	0	1.670502
60	41	662	0	2.174595	67	58	401	0	1.670502
60	41	729	0	2.174595	67	58	870	0	1.670502
60	41	930	0	2.174595	67	58	937	0	1.670502

附录

表 B2 芝加哥具体调度方案

调出 区域	调入 区域	车辆 编号	有无 司机	起点 节点	终点 节点	路径	距离/ 公里	时间/ 分钟	成本
1	10	44	0	116	1751	[116, 1752, 1870, 536, 1751]	0.77	2	1.54
1	10	288	0	116	1751	[116, 1752, 1870, 536, 1751]	0.77	2	1.54
1	10	323	0	116	1751	[116, 1752, 1870, 536, 1751]	0.77	2	1.54
1	10	742	0	116	1751	[116, 1752, 1870, 536, 1751]	0.77	2	1.54
1	10	181	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
1	10	267	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
1	10	298	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
1	10	528	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
1	10	569	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
1	10	577	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
1	10	672	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
1	10	963	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
1	10	987	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
1	10	1110	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
1	10	1285	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
1	10	1298	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
1	10	1301	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
1	10	1321	0	503	1751	[503, 535, 536, 1751]	0.74	2	1.48
14	2	526	0	1531	2074	[1531, 756, 1631, 1630, 1388, 755, 754, 752, 751, 749, 489, 1591, 747, 2152, 529, 2074]	6.02	14	12.04
14	2	838	0	1531	2074	[1531, 756, 1631, 1630, 1388, 755, 754, 752, 751, 749, 489, 1591, 747, 2152, 529, 2074]	6.02	14	12.04
14	2	1040	0	1531	2074	[1531, 756, 1631, 1630, 1388, 755, 754, 752, 751, 749, 489, 1591, 747, 2152, 529, 2074]	6.02	14	12.04
14	2	1207	0	1531	2074	[1531, 756, 1631, 1630, 1388, 755, 754, 752, 751, 749, 489, 1591, 747, 2152, 529, 2074]	6.02	14	12.04
14	2	1339	0	1531	2074	[1531, 756, 1631, 1630, 1388, 755, 754, 752, 751, 749, 489, 1591, 747, 2152, 529, 2074]	6.02	14	12.04
14	2	918	0	1953	69	[1953, 2137, 2135, 2134, 2133, 1578, 2057, 308, 2129, 2128, 2130, 69]	5	11	10
20	2	629	0	1359	1084	[1359, 1362, 1365, 1598,	4.06	9	8.12

调出 区域	调入 区域	车辆 编号	有无 司机	起点 节点	终点 节点	路径	距离/ 公里	时间/ 分钟	成本
20	2	799	0	1359	1084	1530, 1599, 615, 613, 1701, 2066, 1084]	4.06	9	8.12
20	2	1100	0	1359	1084	[1359, 1362, 1365, 1598, 1530, 1599, 615, 613, 1701, 2066, 1084]	4.06	9	8.12
20	2	703	0	1368	1084	[1368, 1603, 1602, 620, 1600, 615, 613, 1701, 2066, 1084]	3.42	9	6.84
20	2	707	0	1368	1084	[1368, 1603, 1602, 620, 1600, 615, 613, 1701, 2066, 1084]	3.42	9	6.84
20	2	1337	1	1368	1084	[1368, 1603, 1602, 620, 1600, 615, 613, 1701, 2066, 1084]	3.42	9	11.34
20	2	1371	1	1368	1084	[1368, 1603, 1602, 620, 1600, 615, 613, 1701, 2066, 1084]	3.42	9	11.34
22	2	228	0	748	2158	[748, 2075, 2153, 529, 2158]	1.61	4	3.22
22	2	400	0	752	2074	[752, 751, 749, 489, 1591, 747, 2152, 529, 2074]	3.74	9	7.48
22	2	828	0	752	2074	[752, 751, 749, 489, 1591, 747, 2152, 529, 2074]	3.74	9	7.48
22	2	114	0	1980	2158	[1980, 783, 782, 2159, 2157, 2158]	2.74	6	5.48