

**TRIP SEQUENCING ALGORITHM DEVELOPMENT
FOR CENTRALIZED, PRESCHEDULED TAXI SYSTEMS**

by

Yun Tang

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Civil Engineering

Spring 2025

© 2025 Yun Tang
All Rights Reserved

**TRIP SEQUENCING ALGORITHM DEVELOPMENT
FOR CENTRALIZED, PRESCHEDULED TAXI SYSTEMS**

by

Yun Tang

Approved: _____
Jack A. Puleo, Ph.D.
Chair of the Department of Civil, Construction and Environmental
Engineering

Approved: _____
Pamela M. Norris, Ph.D.
Dean of the College of Engineering

Approved: _____
Louis F. Rossi, Ph.D.
Vice Provost for Graduate and Professional Education and
Dean of the Graduate College

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed:

Earl Lee, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed:

Philip J. Barnes, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed:

Abdulaziz Banawi, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed:

Dominique Guillot, Ph.D.
Member of dissertation committee

ACKNOWLEDGMENTS

I want to express my gratitude to my advisor, Professor Rusty Lee, for his guidance and support throughout my research. His advice has been invaluable in helping me navigate this project. I am also grateful to my committee members for their helpful feedback, which pushed me to improve my work. I also want to thank my colleagues in the Civil Infrastructure Systems program for their support and collaboration. Finally, a big thank you to my family and friends for always believing in me and cheering me on. Your encouragement made all the difference.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
ABSTRACT	ix

Chapter

1	INTRODUCTION.....	1
1.1	Background and Motivation	1
1.2	Challenge and Contribution.....	2
1.3	Regulatory Framework.....	3
1.3.1	Current TLC Regulatory Baseline.....	4
1.3.2	Proposed Regulatory Adaptations for a Prescheduled Model.....	5
1.4	Assumptions	6
2	LITERATURE REVIEW.....	8
2.1	Overview	8
2.2	Key Area of Investigation	9
2.2.1	Parallel Machine Sequencing	9
2.2.1.1	Definition.....	9
2.2.1.2	History	10
2.2.1.3	Application	12
2.2.2	Taxi Dispatching Problem.....	13
2.2.3	Ride-Hailing Platform in Urban Mobility	15
3	BACKGROUND AND RESEARCH DESIGN.....	19
3.1	Problem statement	19
3.1.1	Research Problem Definition	19
3.1.2	Research Objectives	19

3.2	Research Design	20
3.2.1	Illustrative Example.....	22
4	THEORETICAL BASIS AND METHODOLOGY	27
4.1	Introduction	27
4.2	Notation	27
4.3	Mathematical Formulation	28
4.4	Computational Implementation	29
4.4.1	Data preprocessing	29
4.4.2	Instrumentation and Procedure.....	30
4.5	Experimental Design	36
4.5.1	Time Based Split	36
4.5.2	Spatial Split	39
4.5.3	Parameter Selection	41
4.6	Results and Discussion	42
4.6.1	Dataset Selection and Analysis	42
4.6.2	Performance Analysis.....	43
4.6.2.1	Typical Weekday Scenario (January 9, 2013).....	43
4.6.2.2	Maximum Demand Scenario (February 23, 2013).....	44
4.6.2.3	Demand Volatility Scenario (August 11-12, 2013).....	45
4.6.2.4	Weekly Pattern Analysis (September 8-14, 2013)	46
4.6.3	Sensitive analysis.....	49
4.6.4	Discussion.....	50
5	CONCLUSION	52
5.1	Summary.....	52
5.2	Future Research	54
5.2.1	Integration of Additional Optimization Factors	54
5.2.2	Real-Time Adjustments and Dynamic Optimization	55
5.2.3	Policy and Socioeconomic Implications	56
	REFERENCES	57

LIST OF TABLES

Table 3-1 Manhattan Hive Trip Data	22
Table 3-2 Sample Data Set	23
Table 3-3 Sample Data Results	25
Table 4-1 k value comparison.....	38
Table 4-2 Spatial Split Result, $k=8$	40
Table 4-3 Experiment comparison, $k=8$	40
Table 4-4 Comparison of α_1 value by number of taxis in use	41
Table 4-5 Typical Weekday Scenario	44
Table 4-6 Maximum Demand Scenario	44
Table 4-7 Demand Volatility Scenario, Aug 11	45
Table 4-8 Demand Volatility Scenario, Aug 12	45
Table 4-9 Weekly Pattern Analysis, Time based split	46
Table 4-10 Weekly Pattern Analysis, Spatial based split.....	47

LIST OF FIGURES

Figure 1-1 Three-Day Traffic Pattern.....	7
Figure 3-1 Manhattan Hive: New Taxi Zone	21
Figure 3-2 Sample Data Map	24
Figure 3-3 Sample Data Result with Time Window	26
Figure 4-1 Hive with Taxi Zone Number	30
Figure 4-2 Runtime of single-time computation	37
Figure 4-3 Runtime Optimization for k Split	37
Figure 4-4 Vehicle Count in k Splits	38
Figure 4-5 Spatial Split for $k=8$	39
Figure 4-6 Wait time comparison of Jan 9 data	50

ABSTRACT

This dissertation presents a reservation-based model designed to improve the taxi system performance in New York City. By using offline parallel machine sequencing, the research aims to enhance the allocation of ride requests, reducing the number of taxis required while maintaining a high level of service. Using data from the New York City Taxi and Limousine Commission, this research addresses operational challenges such as deadheading and balancing the supply of available taxis to match demand.

The key contributions include the development of a novel approach to the taxi assignment problem, using predictive offline models rather than traditional real-time algorithms, significantly lowering computational demands. Two allocation strategies—time-based and spatial-based splits—were evaluated experimentally to assess their impact on fleet management. The time-based split consistently showed better results in terms of minimizing the fleet size compared to spatial-based allocation and existing conditions.

The research also employs a parallel processing strategy, which further enhances the taxi assignment process by minimizing unnecessary cross-zone travel and increasing fleet utilization. The results indicate that the proposed model is a scalable and effective solution for urban taxi dispatch, providing practical insights for improving fleet operations in high-density areas like New York City.

Chapter 1

INTRODUCTION

1.1 Background and Motivation

The accelerating pace of urbanization has highlighted how traditional transportation systems in large cities are struggling to keep up with modern demands. With its dense network of bustling streets, diverse neighborhoods spanning five boroughs, and 24-hour transit demand, New York City exemplifies both complexities and opportunities for urban transportation planning.

Taxis have long been an essential part of the city's transit system, symbolizing both the vitality and complexity of urban mobility. However, over the past decade, this traditional mode of transportation has faced mounting pressure from ride-sharing platforms like Uber and Lyft (Schaller, 2021). While these companies have varied urban transit by offering more flexible and on-demand services, the resulting in growth of vehicle numbers has added more challenges to the existing system (Tarduno, 2021).

The New York City Taxi and Limousine Commission (TLC) has made available an extensive dataset that includes trip records for taxis (Taxi & Limousine Commission, 2022). This dataset has been a valuable resource for traffic analysis and understanding taxi operations. Numerous studies have utilized the TLC dataset to explore urban mobility trends in New York City. For example, Chen (2018) analyzed traffic flow and density, highlighting the spatial clustering of taxi movements and identifying key areas with high pickup and drop-off activity. Similarly, Schneider

(2018) conducted an extensive analysis of 1.1 billion taxi and Uber trips, detailing the shifting dynamics between traditional taxis and ride-hailing platforms and their impact on transportation patterns. These studies underscore the wealth of information that can be extracted from the TLC dataset, supporting data-driven approaches to understanding urban transportation issues.

Inspired by the potential of this dataset and the analytical methods employed by past researchers, this dissertation introduces a fully reservation-based taxi model. While previous studies have mapped demand and analyzed traffic patterns, they have not focused on planned dispatch systems that are capable of allocating resources ahead. By using the TLC dataset, this research proposes to develop a model to reduce the taxi fleet size and redundant trips in New York City.

In summary, this dissertation aims to develop a forward-thinking solution that responds to the current real-world challenges in New York City's taxi system. Grounded in comprehensive data analysis and inspired by previous works, it will explore how a reservation-based model can help inspire urban mobility for the future.

1.2 Challenge and Contribution

The optimization of New York City's taxi system reveals challenges in urban transportation that have not been fully addressed by current research. In my assessment, existing taxi dispatch models rely more heavily on real-time machine learning algorithms, but often overlook the potential advantages of offline computational strategies. This dissertation proposes an offline dispatch model that maintains service quality while reducing computational demands.

Another key challenge is that traditional optimization approaches, such as the Traveling Salesman Problem (TSP), are not well-suited to this research problem. TSP

seeks the shortest possible route to visit each of a set of locations exactly once and then return to the starting point. However, the nature of taxi dispatch in an urban environment differs significantly: it involves trips with a distinct origin and destination, forming directed line segments rather than individual points to be visited. This means that each trip has a direction, and the focus is on effectively managing multiple directed trips rather than finding a closed-loop path. TSP does not inherently handle this directional complexity or the continuously changing nature of taxi requests, which makes it unsuitable for addressing the dynamic and directional nature of urban taxi operations. This research, therefore, focuses on a model that better handles dynamic scheduling and resource allocation rather than relying on a path optimization problem like TSP.

In general, the contribution of this dissertation is to introduce a travel model utilizing a reservation system enhanced by offline parallel machine sequencing. This model is designed to allocate ride requests effectively, reducing the number of taxis in use while maintaining stable service.

1.3 Regulatory Framework

New York City's taxicab and for-hire vehicle (FHV) system operates under a comprehensive legal and administrative structure established by the Taxi and Limousine Commission (TLC), which regulates medallion taxicabs, street-hail liveries, for-hire vehicles, and various other operators (*TLC Rules and Local Laws*, 2025). This regulatory environment defines critical elements such as service requirements, licensing conditions, insurance coverage, driver conduct, and permissible technology systems. Below is an overview of the current TLC regulatory

baseline, followed by a summary of the regulatory refinements potentially required by a prescheduled, reservation-based taxi model.

1.3.1 Current TLC Regulatory Baseline

Under the TLC’s Medallion Taxicab Service regulations and For-Hire Vehicle Owners regulations, vehicles must be licensed in accordance with rules that specify operating standards, inspection regimes, and insurance coverage. Medallion taxicabs (yellow cabs) are limited in number, providing street-hail services in all five boroughs. For-hire vehicles (such as black cars, livery cars, and certain high-volume services) typically operate on a prearranged basis but must affiliate with a TLC-licensed base. These provisions ensure that only authorized entities—each holding valid insurance, driver credentials, and vehicle inspections—can offer passenger rides within New York City.

Chapters on Drivers of Taxicabs and FHV’s and Street-Hail Livery address the qualifications, conduct, and service obligations of drivers. For standard taxi operations, the TLC mandates adherence to fare structures (meter-based in yellow cabs, zone/time rates in other segments), pickup/drop-off boundaries, passenger refusal prohibitions, and trip-record reporting. E-Hail and ride-hail applications must be licensed to legally connect passengers to drivers, ensuring compliance with TLC data-collection and rider-protection rules.

TLC regulations also govern trip-record collection (meter-based or dispatch logs) and require robust insurance coverage. These measures protect passengers and other road users by ensuring liability coverage in the event of an incident.

In summary, the existing TLC framework establishes rigorous standards for licensing, driver conduct, vehicle condition, and dispatch operations, primarily oriented toward real-time or near-real-time trip matching.

1.3.2 Proposed Regulatory Adaptations for a Prescheduled Model

The dissertation’s offline, reservation-based taxi system modifies the traditional “hail-and-ride” paradigm by scheduling trips and allocating taxis ahead of time. Implementing such a system under current TLC rules may require:

1. Time-Window and Wait-Time Provisions

While TLC rules address on-demand service expectations (e.g., passenger refusal restrictions, wait times for e-hail trips), no explicit provision governs pre-booked windows allowing small flexibility in pickup times. Dedicated guidelines specifying maximum allowable “tolerance windows” for prescheduled trips would clarify service standards for drivers and passengers.

2. Fleet-Size and Medallion Utilization Adjustments

With medallion numbers capped, a system that reduces the real-time fleet needed might prompt the TLC to revisit how many vehicles must be on duty simultaneously or how many “reservation slots” are made available per shift, especially under congestion-management goals or minimum-service requirements.

3. Data Reporting for Advance Bookings

TLC rules emphasize metered fares and immediate trip-record reporting. A prescheduled system may need additional fields (scheduled pickup time vs. actual) and metrics for lateness or acceptance or rejection rates.

4. Integration with E-Hail Licensing

Current E-Hail Licensing focuses on app-based dispatch for immediate trips. Incorporating fully offline prescheduling under that same license may require clarifying that advanced bookings are permissible or requiring an addendum to E-Hail provider licenses.

The essential pillars of vehicle licensing, driver training, and data oversight remain critical in a reservation-based context. Still, the offline nature of this approach suggests that TLC might consider formalizing wait-time tolerance, amending real-time dispatch obligations, and creating reporting templates that capture prescheduled performance. These refinements would align the proposed model with the city's passenger-safety and service-quality objectives while preserving accountability and consumer protection under existing TLC rules.

1.4 Assumptions

This section outlines the key assumptions that used in the proposed model.

These assumptions are categorized into two parts:

1. Operational Assumptions

- a. **Demand Predictability:** Passenger demand, particularly during peak hours, is observed to generally follow predictable patterns based on historical data. Figure 1-1 shows traffic trips over three consecutive days, showing recurring demand cycles, especially during morning and evening peak hours, which reinforces the model's reliance on demand predictability. Additionally, the model accounts for professional judgment to manage minor deviations that could occur due to external factors.

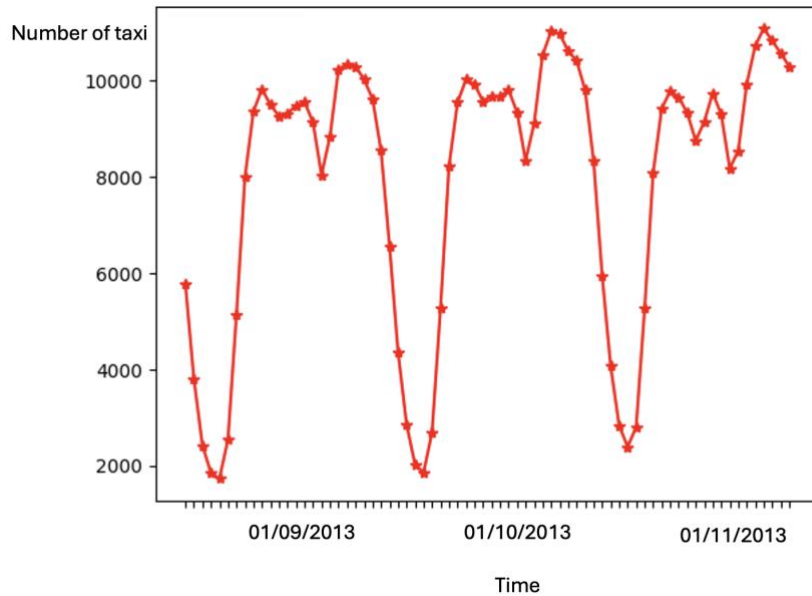


Figure 1-1 Three-Day Traffic Pattern

- b. Stable Traffic Conditions: Traffic conditions are assumed to remain consistent during the study period, without significant disruptions.
- c. Compliance with Dispatch Recommendations: Taxi drivers and the dispatch system are assumed to fully comply with the routes and schedules provided by the algorithms.
- d. Regulatory Adaptations: The model functions within existing TLC regulations, though minor adjustments would enhance implementation.

2. Modeling Assumptions

- a. Homogeneity of the Taxi Fleet: All taxis are assumed to have identical performance characteristics, making them interchangeable in the optimization process.
- b. Accuracy of Data: It is assumed that all historical and real-time data used in the study are accurate and free of significant errors.

Chapter 2

LITERATURE REVIEW

2.1 Overview

This chapter critically reviews the existing research on optimizing urban taxi systems, identifying key limitations, such as the in real-time resource allocation and challenges in fleet utilization. These gaps form the basis of the innovations proposed in this research.

The optimization of taxi fleet size has been studied through three primary approaches: real-time demand response, fixed-route planning, and market-based pricing strategies. Real-time demand response models, exemplified by Yang et al. (2019), utilize GPS data to dynamically adjust fleet size based on temporal demand patterns, offering insights into service coverage optimization while maintaining driver income sustainability.

Fixed-route planning approaches are demonstrated by Babaei et al. (2016), who developed a mixed-integer linear programming model that minimized both travel time and route numbers through capacity-based departure constraints. This methodology achieved efficiency gains through demand bundling and improved taxi turnover rates.

Market-based pricing strategies have been explored through bi-level programming models. Zhang and Ukkusuri (2016) simulated multi-stakeholder interactions in New York City, identifying market inefficiencies addressable through regulatory optimization. Similarly, Yao et al. (2016) employed Stackelberg game

theory to determine optimal fleet size and pricing under stochastic demand conditions, focusing on reducing oversupply through equilibrium-based adjustments.

This research diverges from existing approaches by implementing a reservation-based model enhanced by offline parallel machine sequencing. Unlike the real-time adjustment focus of Yang et al. (2019) or the fixed-route emphasis of Babaei et al. (2016), this study employs an offline, centralized model for pre-booked trip scheduling. This approach addresses identified inefficiencies while avoiding the computational complexity associated with real-time optimization.

2.2 Key Area of Investigation

2.2.1 Parallel Machine Sequencing

2.2.1.1 Definition

The parallel machine scheduling problem involves assigning a set of tasks or jobs to a set of parallel machines, with the goal of optimizing one or multiple objectives (Monma & Sidney, 1979).

Given $n \geq 1$ and jobs J_1, J_2, \dots, J_n , which are assigned to $1 \leq m \leq n$ parallel machines, M_1, M_2, \dots, M_m , any job can be processed on any of the machines. Depend on the difference of the processing speed p_j , there are parallel identical or related machines with the same processing speed, parallel uniform machines refer to a set of machines in a scheduling context where each machine operates at a speed that is a multiple of the speed of the slowest machine in the set, and if the machines vary in speed and this variation is dependent on the specific job being executed, the scenario is classified as involving parallel unrelated machines (Gawiejnowicz, 2020). This

paper will focus exclusively on the case of parallel identical machines, where the processing speeds are consistent across all machines.

2.2.1.2 History

Scheduling theory first appeared in the mid-fifties, Johnson (1954) brought up a two- and three-stage production schedule as one of the classic algorithms. In this early stage, scheduling theory mainly focused on simple models like single-machine and flow-shop scheduling to solve complex industrial resource allocation problems (Muth et al., 1963). Later, in the 1970s, scheduling theory expanded into more complex models, and more research focused on the complexity hierarchy of scheduling problems (Karp, 1975). The work of Conway, Maxwell, and Miller (1967) systematized scheduling theory by introducing the foundational concepts, classifications, and analytical techniques for various scheduling problems. The complexity of machine scheduling problems was investigated and extended, and NP-completeness for many other machine scheduling problems was established (Lenstra et al., 1977).

The field of scheduling experienced remarkable computational advancements and expanded applications. In the 1990s, Lawler (1993) focused on deterministic machine scheduling, developing complexity results, and exploring both optimization and approximation algorithms, particularly for problems involving single machines, parallel machines, and job shops, as well as extensions to resource-constrained project scheduling and stochastic machine scheduling. Peter Brucker (1994, 1999) authored key texts and developed sophisticated scheduling algorithms. The understanding and efficiency of scheduling problem solutions were more advanced, offering insights into approximation limits for scheduling problems with precedence constraints (Afrati et

al., 1999; Munier et al., 1998). In the 2000s, with the growth of research in machine learning, there was a marked integration of advanced technologies in these settings. For example, genetic algorithms and simulated annealing were adopted, for solving complex parallel machine scheduling problems (Vallada & Ruiz, 2011; Kim et al., 2002). Applications of stochastic and dynamic scheduling also have been influential in transportation and logistics (Z.-L. Chen, 2010; Lee & Chen, 2001; M. Pinedo, 2005).

The evolution of manufacturing industries into the Industry 4.0, or the fourth industrial revolution, has propelled scheduling theory into a new era (Diez-Olivan et al., 2019), integrating sophisticated computational techniques and emerging industrial demands. This includes advancements like GraphLab, which improves the efficiency of parallel algorithms for machine learning tasks by enabling asynchronous computation with high data consistency (Low et al., 2014). Key algorithms include metaheuristics like ant colony optimization (Akpınar et al., 2013; Arnaout et al., 2010), as well as hybrid algorithms combining metaheuristics with optimization techniques (Pan et al., 2017; Talbi, 2016).

New topics include dynamic and real-time scheduling (M. L. Pinedo, 2016; Goli & Keshavarz, 2022; Fang & Lin, 2013); energy-efficient scheduling with a focus on sustainability (Wu & Che, 2019); scheduling in cloud computing and data centers (Huang et al., 2012); and robust and stochastic scheduling to handle uncertainty (Verderame et al., 2010; Cohen et al., 2023). Additionally, the integration of the Internet of Things (IoT) and smart technologies in scheduling (Hwang et al., 2012) reflects a trend towards technologically integrated, adaptive, and resource conscious scheduling solutions.

2.2.1.3 Application

Parallel machine scheduling has found widespread use across diverse sectors. In industrial manufacturing, for example, Xhafa and Abraham (2008) highlighted how metaheuristics such as genetic algorithms and tabu search have been used to improve assembly-line scheduling, thus reducing idle time and production bottlenecks. In computing and IT services, Hwang et al. (2012) explored the use of parallel scheduling for large-scale data processing, especially in cloud infrastructures where multiple virtual machines can handle tasks in parallel. A further extension of these ideas appears in service industries, such as airline scheduling: Hancerliogullari et al. (2013) discussed assigning multiple runways (treated as parallel machines) to incoming and outgoing flights to minimize total delay.

Outside these production-oriented contexts, parallel machine scheduling has also been examined in healthcare logistics (Tucker et al., 2009), focusing on how to distribute operating-room tasks to various surgical teams in parallel. Similarly, transportation and supply-chain systems may involve parallel scheduling of delivery trucks, as shown in Ivanov et al. (2016), where coordinating short-term supply routes in a “smart factory” environment can reduce fleet size and operational costs. In each scenario, the core challenge is to adapt the fundamental principles of parallel machine scheduling—namely, allocating multiple independent jobs among parallel, homogeneous or heterogeneous resources—to meet domain-specific constraints such as time windows, setup times, and permissible idle periods.

2.2.2 Taxi Dispatching Problem

The taxi dispatching problem as a subset of vehicle dispatching emerged in the early 1990s. For comprehensive understanding, this section briefly covers the history and evolution of the vehicle dispatching problem.

In the 1960s, the rise of urbanization created a need for efficient transportation systems. The vehicle scheduling problem, first formulated by Dantzig and Ramser (1959), is considered to be a generalization of the Traveling Salesman Problem. This paper uses linear programming to optimize the routing used by a fleet of gasoline delivery trucks between terminals and service stations. Inspired by Dantzig and Ramser, Christofieds and Eilon (1969) gave three methods of solution considering vehicle capacity and distance constraints: branch and bound, “saving” approach (Qarke, 1962) and the 3-optimal tour method. And when used to solve ten vehicle scheduling problems, the 3-optimal approach was the best among them.

Later, Gillett and Miller (1974) brought up a new method for solving the single-depot vehicle dispatching problem called the sweep algorithm. Compared to the exact approach Christofieds and Eilon (1969) developed, this new heuristic algorithm can handle hundreds of vehicles and locations. Gillett and Johnson (1976) modified the sweep algorithm into the multi-terminal sweep algorithm which partitioned the multi-terminal problem into a collection of single-terminal problems that emphasize the spatial distribution criterion.

In the 90s, Bozer and Srinivassan (1991) brought up a promising concept for automated guided vehicle (AGV) systems called tandem configuration which divides systems into single-vehicle loops operating in tandem. In the late 90s, with the upgrading of computer and internet technologies, this phase of research introduced more sophisticated models with real-time data, known as dynamic routing and

scheduling problems (Psaraftis, 1995). Gendreau and Guertin (1999) implemented tabu search heuristics on a parallel platform for real-time vehicle routing and dispatching problem, which allowed the vehicle to serve more customers with less travel distance and lateness compare to other heuristic approaches.

At the beginning of the 21st century, enhanced GPS and communication technologies have advanced dynamic vehicle routing and scheduling. Horn (2002) introduced the “L2sched” system to manage demand-responsive passenger vehicles, optimizing travel time and fleet ridership with classical insertion procedures. Neighborhood search heuristics were further employed by Gendreau and Larsen (2002) to refine real-time vehicle dispatching. Subsequently, the scope of research expanded to encompass vehicle routing problems with time windows, addressing more complex real-world scenarios like perishable food delivery, using algorithms such as parallel tabu search (Ichoua et al., 2003), an insertion heuristic (Potvin et al., 2006), and the Time-Oriented Nearest Neighbor Heuristic (Hsu et al., 2007). By adopting a vehicle-waiting heuristic, Ichoua (2006) introduced forecasted requests to better manage a fleet of vehicles.

In the past decade, the applications of vehicle dispatch problems have diversified. For instance, Schmid (2012) utilized approximate dynamic programming for efficient ambulance dispatching, while Chang et al. (2014) developed a multi-objective genetic algorithm for equitable relief resource distribution. The study of the vehicle routing problem with drones (VRPD) represented another innovative direction (Wang et al., 2017).

Contemporary research in urban mobility, inspired by intelligent transportation systems (ITS), AVs, and ride-hailing services, is leading in a new approach of

solutions. Predictive models using streaming data have been proposed to assist taxi drivers in making informed decisions (Moreira-Matias et al., 2013). Geng et al. (2019) proposes a deep learning model for ride-hailing demand forecasting, the spatiotemporal multi-graph convolution network (ST-MGCN), which could improve vehicle utilization, reduce the wait-time, and mitigate traffic congestion. Moreover, the feasibility of urban transit systems leveraging autonomous vehicles has been explored through various simulation studies (Bagloee et al., 2016; Bischoff & Maciejewski, 2016; Shen et al., 2018), while the concept of shared vehicle networks has been examined for its efficiency and potential to reduce fleet sizes (Jung et al., 2016; Vazifeh et al., 2018).

2.2.3 Ride-Hailing Platform in Urban Mobility

The rise of ride-hailing platforms such as Uber, Lyft, and DiDi has greatly changed urban transportation systems. By offering on-demand services, these platforms have both complemented and competed with traditional modes of transport, such as taxis and public transit. This section reviews studies that apply advanced algorithms, data analysis, and fairness mechanisms to address the challenges in modern mobility systems.

One of the central challenges in ride-hailing is optimizing the matching of drivers to passengers in real-time. Tosoni et al. (2020) explores how scalability issues continue to challenge ride-sharing algorithms, especially in dense urban environments. Their study introduced a locality filtering approach that enhanced the computational efficiency of matching algorithms by limiting the number of trip combinations evaluated in real-time. This method improved the scalability of existing ride-sharing systems, significantly reducing the number of vehicles required and the total distance

traveled. Such improvements are crucial for the continued expansion of ride-hailing platforms, especially as cities become more congested and demand for these services increases. While Tosoni's work (2020) emphasizes real-time efficiency and scalability, this proposed research takes a proactive approach by introducing a reservation-based taxi model. Instead of optimizing matches as requests arise, this work focuses on predicting demand in advance and sequencing trips accordingly

Similarly, Sundt et al. (2021) examine heuristic methods for ride-pooling assignment, contrasting them with computationally intensive optimization-based strategies. They argued that simpler heuristics can strike a balance between maximizing system throughput and maintaining high customer satisfaction by reducing wait times and detours. These findings suggested that while optimization-based models have their merits, heuristic approaches offered a more practical solution in real-time ride-pooling applications where customer experience was a priority.

Advances in algorithmic design have significantly improved the efficiency of ride-pooling platforms. Acharya (2024) proposed a preference-aware task assignment model that incorporated both driver and rider preferences into the matching algorithm. This model, based on the Gale-Shapley deferred acceptance algorithm, balanced system-wide revenue maximization with individual driver satisfaction. The introduction of such preference-aware models reflected a growing recognition that optimizing for overall efficiency must also account for the well-being of drivers.

Despite the advancements in operational efficiency, ride-hailing platforms often face criticism regarding fairness, particularly in income distribution among drivers and access to services for marginalized communities. Raman et al. (2021) investigates this issue by proposing fairness constraints within the objective functions

of ride-pooling platforms, aiming to reduce income inequality and improve service access in underserved areas. Their study, using New York City taxi data, demonstrated that optimizing for fairness can lead to better outcomes both in terms of driver satisfaction and the number of riders serviced in disadvantaged neighborhoods.

In addition to income redistribution, Zhou et al. (2023) highlight the importance of incorporating user preferences into pricing and matching decisions. Their research proposed a fairness-aware pricing model that accounted for users' willingness to share rides, detour times, and overall travel experience. By considering these factors, their model sought to create a more equitable system without sacrificing operational efficiency. This aligned with broader efforts to ensure that ride-hailing platforms not only maximize profits but also provided fair and accessible services to all users.

The impact of ride-hailing platforms on urban transportation extends beyond operational challenges. Jin et al. (2019) explores how Uber and similar platforms interacted with public transit systems, particularly in terms of equity and access. Their study in New York City found that Uber both complemented and competed with public transit, depending on the time of day and location. However, they also found that Uber services were disproportionately used in wealthier areas, raising concerns about transportation equity. This aligned with other studies that suggest ride-hailing platforms may exacerbate social inequities, particularly in cities with existing disparities in transportation access.

Further complicating this issue is the competition between ride-hailing and public transit services. Research has shown that ride-hailing platforms often draw riders away from public transit, particularly during peak hours, contributing to

increased congestion and reduced public transit ridership (Jin et al., 2019). These findings underscore the need for policymakers to carefully manage the integration of ride-hailing platforms within existing transportation networks to mitigate negative externalities, such as congestion and decreased transit equity.

The literature on ride-hailing and ride-pooling platforms underscores the complexity of balancing operational efficiency, fairness, and societal impacts. While advancements in algorithms have greatly improved the scalability and efficiency of these platforms, challenges remain in ensuring equitable access and fair compensation for drivers. As this dissertation explores the development of a fully reservation-based taxi model, insights from these studies will inform the design of a system that addresses both the operational inefficiencies and equity concerns inherent in current ride-hailing and ride-pooling platforms.

Chapter 3

BACKGROUND AND RESEARCH DESIGN

3.1 Problem statement

This chapter outlines the research methodology designed to address the challenge of optimizing taxi fleet size in high-density urban environments. Urban transportation systems face increasing pressures from congestion, environmental concerns, and economic constraints, necessitating more efficient resource allocation strategies. This research employs a systematic approach to evaluate whether taxi fleet size can be reduced while maintaining or improving service quality through the application of parallel machine scheduling techniques.

3.1.1 Research Problem Definition

The core problem addressed in this dissertation is the inefficient allocation of taxi resources in dense urban environments, which results in excessive numbers of idle vehicles contributing to traffic congestion, suboptimal matching of available taxis to passenger demand, increased operational costs for fleet operators, environmental impacts from unnecessary vehicle miles traveled, and variable passenger wait times affecting service quality.

3.1.2 Research Objectives

This dissertation pursues three specific, measurable research objectives.

The first objective focuses on adapting parallel machine scheduling algorithms from manufacturing contexts to taxi dispatching operations. This involves developing a mathematical model that represents taxi vehicles as parallel machines and passenger trips as jobs to be scheduled. The research formulates appropriate objective functions that balance fleet size minimization with service quality while establishing constraints that accurately reflect real-world operational limitations of urban taxi systems.

The second objective establishes and analyzes quantifiable performance metrics for evaluating system efficiency. This includes primary metrics such as required taxi fleet size and passenger wait time. The research conducts comparative analysis between the proposed model and baseline dispatch methods to quantify improvements.

The third objective determines optimal parameter configurations through systematic sensitivity analysis. This involves investigating the impact of varying time windows (α values) on fleet size requirements. The research aims to identify the most influential parameters for fleet size reduction while maintaining service quality.

3.2 Research Design

The theoretical foundation of this research is based on a custom-designed approach to improve the New York City taxi system. Drawing upon operations research, the strategy employed involves a two-step process to regionalize the city into computationally manageable 'hives', a term adopted in this research to represent strategically defined urban zones.

In the first step, an adjacency matrix was created to denote connected taxi zones (Taxi and Limousine Commission (TLC), 2023) with binary indicators. Alongside this, a trip matrix was established, enumerating trips between zones. From

this data, high-demand zones were selected as initial hives to begin the clustering process.

The second step involved an iterative selection process. Beginning with the zone selected in the first step, the algorithm proceeds to calculate trip volumes for all adjacent zones. Sequentially, the zone with the highest proportion of same-hive trips was incorporated into the hive. This process was continued until the proportion of same-hive trips arrives at least the 30% threshold, signaling the completion of one hive before initiating the formation of the next. The results are shown in Table 3-1.

An example application of this methodology is shown for the Manhattan area in Figure 3-1. Here, the empirical data shaped the creation of eight hives, aiming to significantly foster same-hive trips and curtail the operational inefficiencies associated with cross-hive travel.

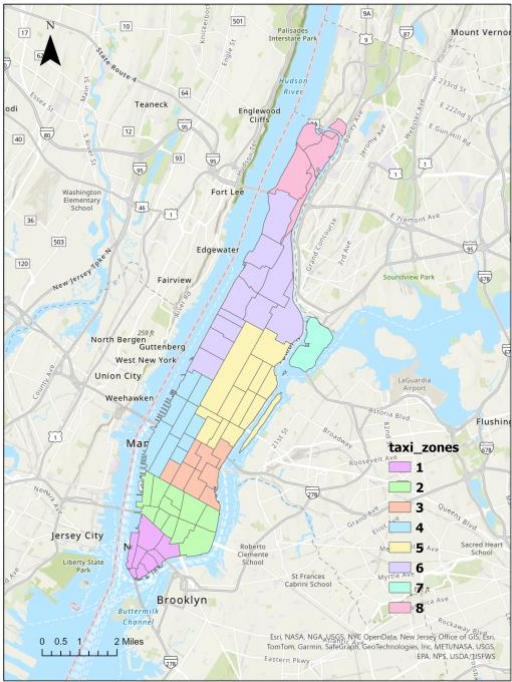


Figure 3-1 Manhattan Hive: New Taxi Zone

Table 3-1 Manhattan Hive Trip Data

Hive	Total Trip	Same Hive Trip	Same Hive Trip Percentage
1	4288437	1314509	30.7%
2	13484299	5334769	39.6%
3	34685153	12900616	37.2%
4	36516333	15945670	43.7%
5	92359996	66838210	72.4%
6	4128116	1701420	41.2%
7	938	900	95.9%
8	16448	76	0.5%

This unique approach enabled the use of more tractable computational methods for solving the overarching problem, as it reduced the dataset to a manageable size. Within these hives, the objective is to efficiently allocate taxis based on pre-known demand patterns, essentially treating each hive as an independent entity for the purpose of optimization.

To explain the employed algorithm within the hive, the next section will present a small-scale sample that explains its operation.

3.2.1 Illustrative Example

This example will illustrate the decision-making process in selecting zones based on the trip data and will showcase the practical implementation of the theoretical model. Two algorithms will be employed for comparison: a baseline algorithm utilizing the First-Come-First-Serve (FCFS) approach, and the proposed algorithm introduced in Chapter 4. The results will highlight the effectiveness and advantages of the developed methodology.

Table 3-2 presents a sample dataset of 8 taxi trips occurring between 7 am and 8 am. To visualize these trip patterns spatially, Figure 3-2 illustrates the corresponding origin-destination pairs as directional vectors on a coordinate grid, with each line representing the route of a single trip. This visualization enables the identification of spatial movement patterns.

Table 3-2 Sample Data Set

Trip ID	Pick up Location	Drop off Location	Pick up Time	Drop off Time	Trip Duration(min)
1	(1,1)	(2,0)	7:00	7:01	1
2	(2,7)	(5,12)	7:13	7:21	8
3	(4,4)	(5,0)	7:20	7:25	5
4	(15,18)	(17,20)	7:48	7:51	3
5	(9,13)	(15,5)	7:01	7:14	13
6	(7,18)	(9,16)	7:40	7:44	4
7	(17,1)	(18,2)	7:30	7:32	2
8	(15,12)	(17,10)	7:00	7:03	3

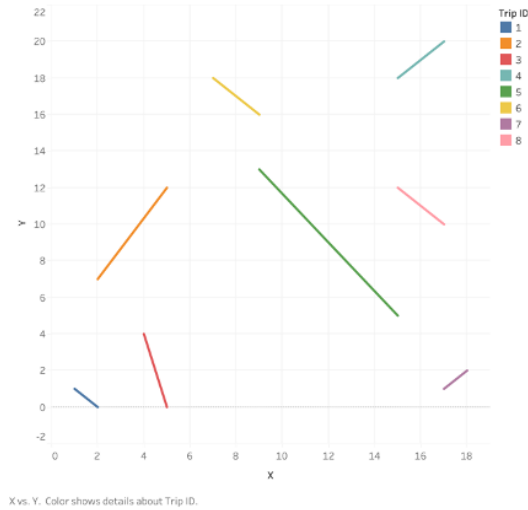


Figure 3-2 Sample Data Map

For the baseline algorithm, the initial step involves the systematic processing of trip data. This process commences with organizing trips in ascending order of pick-up times, enabling sequential allocation to the corresponding number of operational taxis based on First-Come-First-Served (FCFS) principles.

In contrast, the proposed algorithm first sorts of trips by ascending drop-off time, then assigns each trip to either an existing taxi or a new taxi by selecting the option with the lowest cost.

For example, we assign the first trip (trip 1) to an initial taxi (taxi A). When evaluating the second trip (trip 8), we calculate two potential costs: the cost of assigning it to the existing Taxi A ($Cost_A$) or the cost of introducing a new taxi, Taxi B ($Cost_B$). The algorithm selects the option with the lower cost. The mathematical expression that we seek to optimize is given below. Detailed explanations are provided in Chapter 4.

$$Cost = \alpha_1 * \sum_{i=1}^n \text{Max} \left(d'_{i-1} + \frac{\|L_{d_{i-1}}, L_{p_i}\|_1}{speed} - p_i - \epsilon, 0 \right) + \alpha_2 * \left(n - \sum_{i=1}^n \sum_{j=1}^m x_{ij} \right) + \alpha_3 * \sum_{j=1}^m y_j,$$

where $\alpha_1 = 2000, \alpha_2 = 10^8, \alpha_3 = 10^4, \epsilon = 15 \text{ min}$.

$$Cost_A = 2000 \times \text{Max}\{(7:01 + 25\text{min} - 7:00 - 15 \text{ min}), 0\}$$

$$+ M \times [2 - (x_{11} + x_{21})] + 10^4 \times 1 = 2000 \times 11 + 10^4 = 3.2 \times 10^4$$

$$Cost_B = 2000 \times 0 + M \times [[2 - (x_{11} + x_{21} + x_{21} + x_{22})] + 10^4 \times 2 = 2 \times 10^4$$

In this example, since taxi B has the lower cost, we assign trip 8 to taxi B. This process repeats iteratively until all trips have been assigned.

Table 3-3 Sample Data Results

	Baseline Algorithm	Proposed Algorithm
Taxi A	1, 2, 7	1, 2, 3, 4
Taxi B	8, 3, 4	8, 5, 7, 6
Taxi C	5, 6	

The comparative performance of both algorithms is illustrated in Table 3-3 and visualized in Figure 3-3, demonstrating the efficiency gains achieved through the proposed method.

Operational constraints are embedded within the algorithm to maintain defined service standards. An initial tolerance window of fifteen minutes is set to ensure taxis arrive within an acceptable timeframe; other wait time is also tested in Chapter 4. This

time window serves as a constraint, guiding the algorithm to allocate the minimum number of taxis required to handle the scheduled trips.

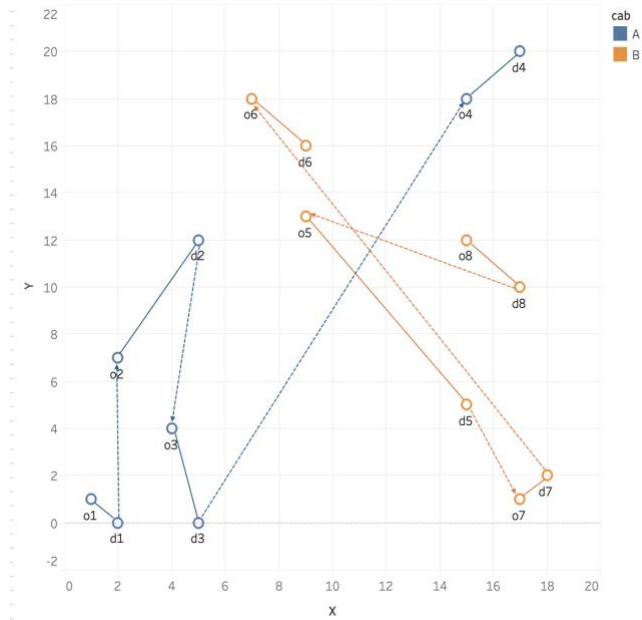


Figure 3-3 Sample Data Result with Time Window

Chapter 4

THEORETICAL BASIS AND METHODOLOGY

4.1 Introduction

The chapter details the underlying mathematical models, and algorithmic approaches used in this research to develop and evaluate the proposed reservation-based taxi system.

4.2 Notation

This section introduces the key concepts and notations that will be employed in the subsequent mathematical formulation.

J: The set of jobs (trip).

M: The set of machines (taxi).

D_i : duration time of trip i , $D_i = d_i - p_i$.

p_i : pick up time of trip i .

d_i : drop off time of trip i .

p'_i : actual pick up time of trip i .

d'_i : actual drop off time of trip i .

L_{p_i} : pick up location of trip i .

L_{d_i} : drop off location of trip i .

$MD_{(i,i+1)}$: manhattan distance bewteen trip i and trip $i + 1$.

4.3 Mathematical Formulation

1. Sets and parameters

$$J = \{J_1, J_2, \dots, J_n\}, \quad n \in \mathbb{R}$$

$$M = \{M_1, M_2, \dots, M_m\}, \quad m \in \mathbb{R}$$

$$L_{p_i} = (x_{L_{p_i}}, y_{L_{p_i}}), \quad L_{p_i} \in \mathbb{R}^2$$

$$L_{d_i} = (x_{L_{d_i}}, y_{L_{d_i}}), \quad L_{d_i} \in \mathbb{R}^2$$

$$\text{For } i \in [n], \quad MD_{(i,i+1)} = \|L_{d_i} - L_{p_{i+1}}\|_1$$

2. Decision variables

$$i \in [n], j \in [m]$$

$$x_{ij} = \begin{cases} 1, & \text{assign trip } i \text{ to taxi } j, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_j = \begin{cases} 1, & \text{taxi } j \text{ is used,} \\ 0, & \text{otherwise.} \end{cases}$$

3. Objective function: Minimize the number of machines used.

$$\min \alpha_1 * \sum_{i=1}^n Max + \alpha_2 * \left(n - \sum_{i=1}^n \sum_{j=1}^m x_{ij} \right) + \alpha_3 * \sum_{j=1}^m y_j$$

$\alpha_1 \alpha_2 \alpha_3$ are positive weighting, n is the total number of trips, ϵ is the number of minutes that a taxi is allowed to be late before incurring a penalty.

In our model, $\epsilon = 15$ min, speed = 11.2 mph. Speed is modeled as a constant average value for simplicity in this formulation. In practice, speed varies by route, time of day, and traffic conditions, which can be incorporated through dynamic speed parameters in extended versions of the model.

4. Constraints

- Assignment Constraint: Each job must be assigned to exactly one machine.

$$\sum_{j=1}^m x_{ij} = 1, \quad i \in T$$

- Idle Time Constraint: To ensure the idle time between consecutive jobs on the same machine.

$$p_{i+1} + \epsilon \geq d_i + \frac{\|L_{d_i^-}, L_{p_{i+1}}\|_1}{speed}, \quad i \in J, j \in M$$

4.4 Computational Implementation

4.4.1 Data preprocessing

The empirical analysis conducted in this study utilizes data sourced from the NYC Open Data portal, provided by the Taxi and Limousine Commission (TLC) (*TLC Trip Record Data - TLC, 2022*). In pursuit of the study's objectives to improve same-hive travel efficiency, the conventional NYC taxi zones were redefined into eight novel, strategically delineated hives. The dataset chosen is the entire Manhattan area, illustrated in Figure 4-1.

This research utilizes 2013 New York City taxi data for two key reasons. First, this period represents a baseline of traditional taxi operations before major shifts in the transportation market, providing clear insights into system dynamics. Second, the 2013 dataset contains detailed geographic coordinates that were later removed from TLC data releases, enabling precise spatial analysis crucial for model development. While more recent data exists, the COVID-19 pandemic's disruption of urban mobility patterns makes historical data more suitable for understanding typical taxi operations.

The dataset utilized in this analysis comprises eight attributes, representing key dimensions of taxi trips: taxiID (the unique identifier for each taxi), pick up time and drop off time (timestamp), pick up longitude and pick up latitude (geographical coordinates of the pickup location), drop off longitude and drop off latitude (geographical coordinates of the drop-off location), and trip time (duration of the trip).

These attributes can be represented as an 8-tuple: $J_i = \{ID, p_i, d_i, x_{L_{p_i}}, y_{L_{p_i}}, x_{L_{d_i}}, y_{L_{d_i}}, D_i\}$. Each variable corresponds to the attributes, providing a structured framework for subsequent analysis.

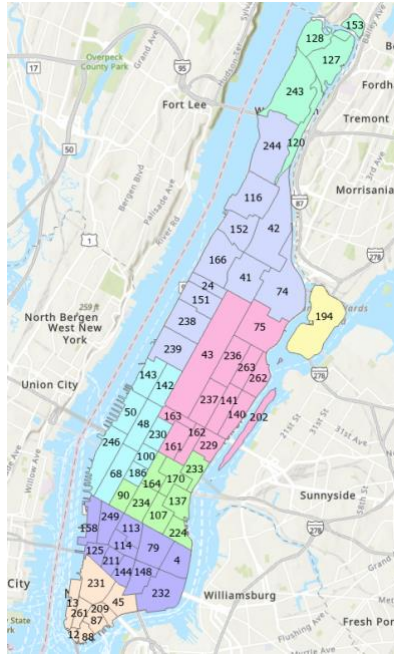


Figure 4-1 Hive with Taxi Zone Number

Prior to the analysis, erroneous geospatial coordinates were removed using ArcGIS, filtering out anomalies that fell outside the plausible range for New York City. Additionally, trip records manifesting a zero-travel time, or unusually protracted durations were removed. Eliminating these anomalies ensures the accuracy and reliability of the further analysis.

4.4.2 Instrumentation and Procedure

The experiments were conducted on a Macintosh HD, utilizing Python 3.0 for all computational tasks. This research focused on optimizing taxi dispatch by testing two allocation approaches: spatial-based and time-based trip allocation. The dataset

used for analysis is based on a full day of taxi data across Manhattan, divided into time intervals. Both spatial and time-based splits were compared to evaluate which method yields more efficient taxi fleet utilization.

In addition to the two experimental algorithms, the baseline algorithm serves as a reference point. The baseline approach initializes taxis and trips, sorted by expected pick-up times in ascending order. The algorithm iteratively assigns the earliest available trip to an available taxi until all trips are scheduled or no taxis can accommodate further trips.

The detailed pseudo code for the baseline algorithm is below:

Baseline Algorithm

$M = \{M_j: j = 1, \dots, numTaxi\}$

$J = \{J_i: i = 1, \dots, numTrips\}$:

List of Trips sorted by expected pick up time in ascending order

$j = 0$

While J is not empty:

While any J_i in J is available for M_j

assign the earliest available trips J_i to M_j

remove J_i from J

continue from J_{i+1} until no J_i in J is available for M_j

$j += 1$

Building on this, the spatial-based and time-based algorithms introduce more dynamic, context-specific methodologies. For the spatial split algorithm, Manhattan is divided into predefined zones (hives), and taxis are assigned based on trip locations

within those zones. For the time-based split algorithm, trips are grouped into hourly intervals, and taxi assignments are optimized based on the time of the request.

The general setting of taxi assignment can be described as follows. At the beginning, a set of taxi and trips are given. Then an algorithm selects its action in an action space for every trip. The action space includes rejecting the trip, assigning the trip to a new taxi, and assigning the trip to a taxi with trip assigned. In this work, the performance of the algorithm is evaluated by the total number of taxis involved, the sum of delay time, and the number of trips rejected. However, the method to solve the problem is not straightforward due to two main reasons. First, metrics interfere with each other, so it is hard to find one solution with best performance in every metric simultaneously. Second, the computation cost is considerable when the number of trips and number of taxis are in real life scale.

This work contributes in two aspects to solving the taxi assignment problem. First, a cost function is designed to quantify the effect of the metrics mentioned above, and thus taxi assignment can be interpreted as an optimization problem for smallest cost. Second, a time-efficient algorithm is proposed to solve the optimization problem with acceptable time cost. The detail of these two points will be described in detail in the following part of this section.

The cost item for the total sum of delay time, f_{delay} , is defined as

$$f_{delay} = \sum_{i=1}^n \text{Max}\{\text{Delay}_i, 0\},$$

where n is the total number of trips, Delay_i is the time delay for the i^{th} trip, which is computed as

$$\text{Delay}_i = \left(d_{i-1} + \frac{\|L_{d_{i-1}^-}, L_{p_i}\|_1}{\text{speed}} - p_i - \epsilon \right)$$

where d_{i-1} is the drop-off time of the $(i-1)^{th}$ trip, and p_i is the pick-up time of the i^{th} trip. $\|L_{d_{i-1}} - L_{p_i}\|_1$ is the Manhattan distance between the $L_{d_{i-1}}$ drop-off location of the $(i-1)^{th}$ trip, and L_{p_i} the pick-up location of the i^{th} trip. Moreover, ϵ is the time delay tolerance, initially set as 15 minutes.

The cost for the trip rejection is

$$f_{rej} = \left(n - \sum_{i=1}^n \sum_{j=1}^m x_{ij} \right),$$

where $x_{ij} = \begin{cases} 1, & \text{assign trip } i \text{ to taxi } j, \\ 0, & \text{otherwise.} \end{cases}$

The cost for the total number of taxis used is

$$f_{num} = \sum_{j=1}^m y_j,$$

where $y_j = \begin{cases} 1, & \text{taxi } j \text{ is used,} \\ 0, & \text{otherwise.} \end{cases}$

Finally, the overall cost function f is a weighted sum of the different costs to balance competing performance metrics:

$$f = \alpha_1 * f_{delay} + \alpha_2 * f_{rej} + \alpha_3 * f_{num},$$

where $\alpha_1, \alpha_2, \alpha_3$ are the weight for each cost item, and their value can be adjusted to establish a trade-off between different cost items.

In our setting, $\alpha_1 = 100$, $\alpha_2 = M$, M is a large number, and $\alpha_3 = 10^4$. The rationale for these values will be elaborated later in this chapter.

With the cost function defined above, the optimal solution of the taxi assignment problem is the solution with minimal cost f . However, it is practically impossible to find the global optimal solution for f , because it is a NP-hard problem (Kravchenko & Werner, 1997). Therefore, this work proposes solving \tilde{f} , an approximation of f , using a greedy algorithm-based approach.

We first define f_i , the cost function of solving the i^{th} trip as:

$$f_i = \min (\alpha_1 * f_{used}, \alpha_2 * f_{rej}, \alpha_3 * f_{new}),$$

where f_{used} is the cost of assigning a trip to a taxi used before, f_{rej} is for cost of rejecting the current trip, and f_{new} is the cost to assign the current trip to a new taxi. $\alpha_1, \alpha_2, \alpha_3$ have same meanings and values as in f . When solving the i^{th} trip, the action with smallest f_i is selected.

The cost of assigning one trip to a used car, f_{used} , is defined as:

$$f_{used} = \min (\{\max \left(d_j + \frac{\|L_{d_j}, L_{p_i}\|}{speed} - p_i - \epsilon, 0 \right) : j \in M\}),$$

where M is the set of all used taxis, and d_j is the last drop-off location of j^{th} taxi.

With f_i defined above, \tilde{f} is defined as:

$$\tilde{f} = \sum_{i=1}^n f_i$$

We proposed a method based on the greedy algorithm to find a local optimal solution of \tilde{f} . First, all the trips are started in ascending order sorted by their expected drop-off time. Then trips are assigned in sequence by selecting the action with smallest cost value for this single step. This step repeats until all trips are assigned. The detail of this algorithm is shown below.

Proposed Algorithm

$M = \{M_j : j = 1, \dots, numTaxi\}$: Taxis assigned at least one trip

$J = \{J_i : i = 1, \dots, numTrips\}$:

Trips sorting by expected drop off time in ascending order

For i in $[1, 2, \dots, \text{len}(J)]$:
 For j in $[1, 2, \dots, \text{len}(M)]$:
 Calculate C_{ij} , cost of assigning J_i to taxi M_j
 $C_1 = \min\{C_{ij}: j = 1, \dots, \text{len}(M)\}$
 $k = \text{argmin}\{C_{ij}: j = 1, \dots, \text{len}(M)\}$
 Calculate C_2 , cost of assigning J_i to a new taxi
 Calculate C_3 , cost of not assigning J_i to any taxi
 If $C_1 < C_2$ and $C_1 < C_3$:
 assign J_i to M_k
 else if $C_2 < C_1$ and $C_2 < C_3$:
 assign J_i to M_{j+1}
 add M_{j+1} to M
 else:
 not assign J_i to any taxi

As a method with a time complexity of $O(mn)$, where m represents the number of taxis used and n represents the number of trips, the time cost to solve a single trip increases significantly as the number of taxis involved becomes larger. In computational complexity terms, $O(mn)$ indicates that the time required to compute a solution grows proportionally with the product of m and n , making the problem computationally expensive for large-scale instances.

To further accelerate computation, the entire trip set is divided into k subsets, and the method is applied independently to each subset. While this operation introduces additional involvement of taxis, it reduces the computational time

significantly, achieving a practical trade-off between computation speed and resource allocation.

4.5 Experimental Design

4.5.1 Time Based Split

The initial experiments encountered significant runtime issues when dealing with larger dataset. As the dataset size increased, the running time of the model began to slow considerably. Figure 4-2 illustrates the relationship between runtime for a single trip and the increasing number of vehicles during the experiments conducted on the dataset comprising 22,961 trips from January 9th, between 7 am and 8 am. The relationship is predominantly linear for a single iteration, while the overall computational cost, including all iterations, demonstrates a quadratic trend due to the summation over multiple trips.

To address the scalability issue, the dataset was divided into k splits to achieve better runtime efficiency. The data splitting methodology began with all trips being sorted chronologically by their start times. Following this, each trip was assigned to a specific split using a modulus operation, where the index of each trip (i) was taken modulo the number of splits (k). This approach distributed the trips among different splits systematically. For $k = 1$, the algorithm compared all potential trips, leading to minimized vehicle usage since every trip was considered comprehensively. However, for $k > 1$, not all trips could be compared simultaneously, which led to an increase in vehicle usage due to the reduced ability to optimize across the entire dataset. This trade-off between runtime efficiency and vehicle usage is an important consideration when determining the number of splits.

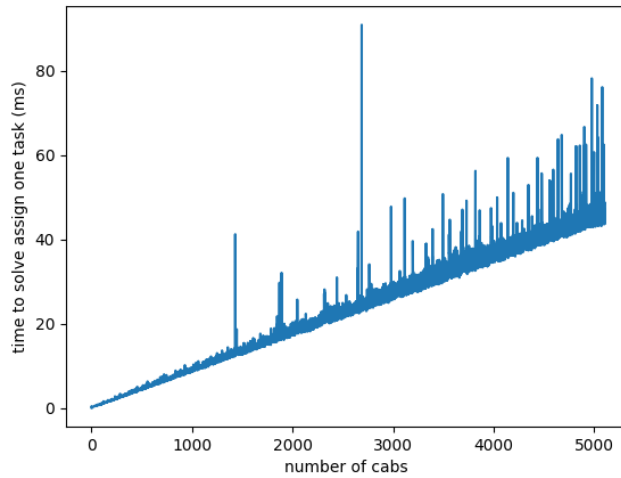


Figure 4-2 Runtime of single-time computation

Figure 4-3 presents the runtime optimization for the same dataset. Here, the dataset was split into k splits for parallel processing. The introduction of multiprocessing enabled the splits to be processed simultaneously, significantly reducing the total runtime. Figure 4-4 shows the vehicle count required for different values of k . As k increased, the running time decreased, while the number of vehicles increased. This indicates that finding a balance for k is critical to minimizing both vehicle usage and computational costs.

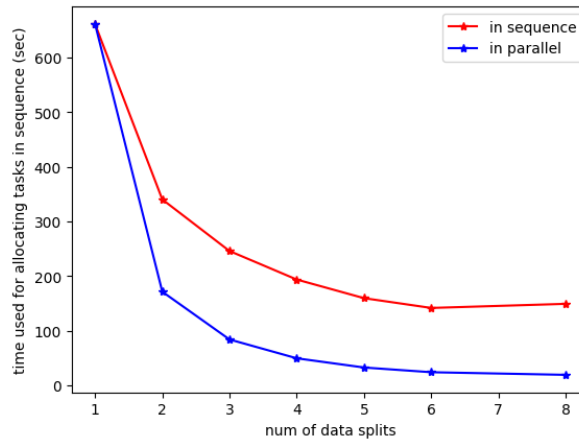


Figure 4-3 Runtime Optimization for k Split

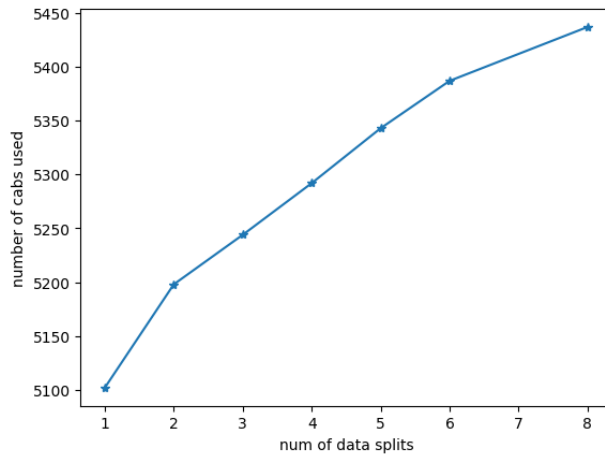


Figure 4-4 Vehicle Count in k Splits

We conducted experimentation with different k values to identify the balance between computational efficiency and solution quality using Jan 9, 2013, data. As shown in Table 4-1, increasing k values resulted in faster computation times but required more taxis. The value eight was selected as the optimal value for our experimental implementation because it offered the best trade-off between computational efficiency and fleet size optimization. With $k = 8$, the calculation time was reduced to 5 minutes while the taxi fleet size increased only marginally from 7,888 to 8,048. Moreover, the passenger wait time remains steady around 13 min. Further increases in k values provided diminishing returns in computational efficiency while continuing to increase fleet requirements more substantially.

Table 4-1 k value comparison

	$k=4$	$k=8$	$k=16$	$k=32$
Number of taxis	7888	8048	8291	8619
Run time (min)	15.7	5.0	2.5	1.3
Passenger wait time (min)	13.55	13.28	12.91	12.45

4.5.2 Spatial Split

Following the promising results of the time-based split, this research also explored a spatial-based split to determine its viability for optimizing runtime and vehicle utilization. The spatial split involved clustering the Manhattan area into eight predefined zones, consistent with the existing division into hives.

For consistent comparison with the time-based approach, we selected $k = 8$ for the spatial split. This choice was motivated by both practical considerations, matching existing operational boundaries, and our findings from the time-based split experiments where $k = 8$ provided an optimal balance between computational efficiency and solution quality. Figure 4-5 illustrates the spatial clustering of the dataset from January 9th, 6 am to 10 am, with results that closely matching the original hive map.

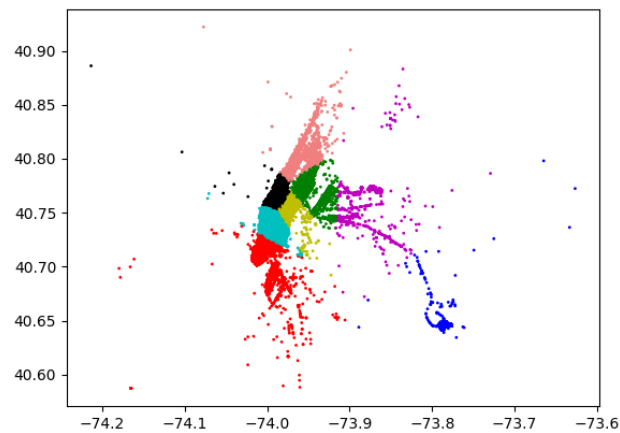


Figure 4-5 Spatial Split for $k=8$

The detailed results of the spatial split are presented in Table 4-2. The spatial split exhibited considerable cross-hive trips, leading to inefficiencies in fleet management. Specifically, zones experienced a substantial percentage of trips moving across different areas, resulting in increased operational complexity and higher vehicle

usage. For example, Cluster 3 (cyan) had 18,263 trips across different areas, significantly contributing to increased vehicle requirements.

Table 4-2 Spatial Split Result, $k=8$

Cluster	Color code	Number of taxis	Trip in same area	Trip in different areas	Total Trips	Inner area percentage (%)
0	red	1090	3279	6213	9492	34.5
1	green	2057	5714	15012	20729	27.5
2	blue	503	175	1014	1189	14.7
3	cyan	2080	9992	18263	28255	35.4
4	purple	677	337	1856	2193	15.4
5	olive	1519	7142	12083	19225	37.2
6	black	1472	5159	13150	18309	28.2
7	pink	8484	2555	5737	8292	30.8

Both time based split and spatial based split experimental results are summarized in Tables 4-3. It is evident that both the time-based and spatial splits outperformed the existing scenario in terms of vehicle usage. However, the time-based split performed significantly better in terms of both the number of taxis and running time. Therefore, the time-based split was chosen for the final set of experiments.

Table 4-3 Experiment comparison, $k=8$

Algorithm	Number of taxis in use	Run time (min)
Existing condition	11098	N/A
Time based split	7574	1.0
Spatial based split	10246	3.7

4.5.3 Parameter Selection

Our objective function incorporates three principal weight parameters ($\alpha_1, \alpha_2, \alpha_3$) that control the relative importance of competing goals: minimizing passenger wait time (trip delay), maximizing service coverage, and optimizing fleet utilization.

The trip delay weight parameter (α_1) was set to a baseline value of 100, which our experiments showed provided an appropriate balance between passenger convenience and operational efficiency. As shown in Table 4-4, our week-long analysis from January 7 through 13, 2013, increasing α_1 from 100 to 2000 resulted in only marginal increases in fleet size—ranging from 0.3% (January 9) to 1.9% (January 12)—while maintaining similar passenger wait times. This minimal difference suggests that our model is relatively robust to changes in this parameter once a certain threshold is reached.

Table 4-4 Comparison of α_1 value by number of taxis in use

	$\alpha_1 = 100$	$\alpha_1 = 2000$	Existing condition
Jan 7 Mon	7632	7657	22236
Jan 8 Tue	7926	7958	23036
Jan 9 Wed	8069	8085	23439
Jan 10 Thu	8953	9060	23793
Jan 11 Fri	9845	9834	23750
Jan 12 Sat	8617	8781	22540
Jan 13 Sun	8645	8663	21769
Avg. Passenger wait time (min)	13.2	13.13	N/A

The service coverage parameter (α_2) was set to M , where M represents a large value (10^8) to ensure that all trip requests are serviced. This hard constraint guarantees 100% service coverage, which is essential for comparing our algorithm's performance against the existing taxi system where all trips are assumed to be served.

The fleet utilization parameter (α_3) was calibrated to 10^4 , striking a balance between minimizing the number of vehicles while ensuring that individual taxis were not overutilized to the point of operational impracticality.

4.6 Results and Discussion

4.6.1 Dataset Selection and Analysis

This section presents a detailed analysis of the New York City taxi trip data from 2013, which informed our experimental design decisions and parameter calibration. We conducted an extensive analysis of 172,731,922 taxi trips across all 365 days of 2013. The annual data exhibited an average daily trip volume of 473,238.14 trips, with substantial variability ranging from a minimum of 195,405 trips to a maximum of 584,812 trips on February 23, 2013.

For initial algorithm development and parameter tuning, we selected January 9, 2013, as our primary test date. This date was chosen based on several important criteria. January 9 represents a typical weekday with trip volumes, approximately 473,000 trips, closely matching the annual average. As a Wednesday, it falls in the middle of the workweek pattern, where our analysis showed a gradual increase in trip volumes from Monday, 12.94% of weekly trips, through Friday, 15.24% of weekly trips. The date does not coincide with any major holidays, extreme weather events, or city-wide special occasions that might skew demand patterns.

To assess our algorithm's robustness under more challenging conditions, we also identified and analyzed several outlier days. The busiest day, February 23, with over 570,000 trips, served as stress tests for our algorithm under maximum demand conditions. We also examined the August 11 to August 12, 2013, which showed the most dramatic day-to-day change in our dataset, with a 56.09% decrease followed by a 104.32% increase, providing an excellent test case for algorithm adaptability during rapid demand fluctuations.

Finally, for a typical week analysis, we selected the second full week of September, September 8 to September 14. This span avoids holiday or major-event anomalies and excludes the monthly peak on September 21, thereby offering a stable baseline period that aligns closely with overall average trip volumes.

4.6.2 Performance Analysis

4.6.2.1 Typical Weekday Scenario (January 9, 2013)

Table 4-5 presents a comparative analysis of taxi utilization efficiency under three distinct scenarios: existing conditions, the baseline algorithm, and our proposed approaches.

Our time-based split algorithm achieved a 65.6% reduction in fleet size compared to existing conditions, while maintaining acceptable passenger wait times. It also demonstrated superior efficiency to the baseline algorithm, requiring 25.3% fewer vehicles with only a 4.38-minute increase in average wait time, while dramatically reducing computational requirements. The spatial split approach, while still improving upon existing conditions with a 44.8% fleet reduction, proved less efficient than the time-based approach.

Table 4-5 Typical Weekday Scenario

		Number of taxis in use	Passenger wait time (min)	Run time (min)
Existing condition		23439	N/A	N/A
Baseline algorithm		10809	8.90	40.2
Proposed algorithm	Time based split	8069	13.28	5.0
$k=8$	Spatial based split	12936	13.68	21.0

4.6.2.2 Maximum Demand Scenario (February 23, 2013)

To test scalability under peak conditions, we applied our algorithms to February 23, the busiest day of 2013, shown below on Table 4-6.

Despite the 23.7% higher trip volume compared to January 9, our time-based split algorithm required only 10.5% more taxis, demonstrating excellent scalability under peak demand. The algorithm maintained consistent service quality with a nearly identical average wait time, while keeping computational requirements manageable. This non-linear scaling suggests that our approach becomes increasingly efficient at higher demand levels.

Table 4-6 Maximum Demand Scenario

		Number of taxis in use	Passenger wait time (min)	Run time (min)
Existing condition		23216	N/A	N/A

Baseline algorithm		13293	8.96	63.45
Proposed algorithm	Time based	8914	13.31	7.97
	split			
k =8	Spatial based	14324	13.68	31.40
	split			

4.6.2.3 Demand Volatility Scenario (August 11-12, 2013)

To evaluate robustness during rapid demand changes, we analyzed performance during the most volatile 48-hour period in our dataset, shown below on Table 4-7 and Table 4-8.

Table 4-7 Demand Volatility Scenario, Aug 11

		Number of taxis in use	Passenger wait time (min)	Run time (min)
Existing condition		11160	N/A	N/A
Proposed algorithm	Time based	3988	12.57	2.5
	split			
k=8	Spatial based	6516	13.20	12.8
	split			

Table 4-8 Demand Volatility Scenario, Aug 12

		Number of taxis in use	Passenger wait time (min)	Run time (min)
Existing condition		22647	N/A	N/A
	Time based split	7385	13.15	5.7

Proposed algorithm	Spatial based split	11875	13.64	31.0
---------------------------	---------------------	-------	-------	------

***k* =8**

During this period of extreme volatility, the time-based algorithm demonstrated remarkable adaptability, adjusting from 3,988 taxis on August 11 to 7,385 taxis on August 12 in response to the demand surge. This response closely tracked the proportional increase in demand without compromising service quality, as evidenced by the consistent wait times across both days. This adaptability highlights the algorithm's effectiveness in dynamic real-world conditions where demand can fluctuate substantially.

4.6.2.4 Weekly Pattern Analysis (September 8-14, 2013)

Tables 4-9 and 4-10 show the evaluated performance over a representative week in September using both time-based and spatial-based split algorithms.

The time-based method consistently reduced fleet requirements by 58 to 63% while maintaining passenger wait times between 13.10 to 13.30 minutes. Fleet needs followed a predictable pattern, increasing from Sunday through Thursday before declining on weekends, aligning with typical urban mobility cycles.

While the spatial-based approach achieved 35 to 42% fleet reductions, it consistently underperformed compared to the time-based method and required 3 to 4 times longer computational run time.

The consistent service quality across varying demand volumes validates the algorithm's ability to balance efficiency with service reliability.

Table 4-9 Weekly Pattern Analysis, Time based split

	Number of taxis in use		Passenger wait time (min)	Run time (min)
	Existing condition	Time based split		
Sep 8	22655	8719	13.10	7.8
Sep 9	23545	8722	13.24	7.7
Sep 10	24249	9400	13.30	12.9
Sep 11	24616	9108	13.29	11.7
Sep 12	24740	10307	13.27	18.7
Sep 13	24774	9268	13.20	18.7
Sep 14	23578	9004	13.21	17.8

Table 4-10 Weekly Pattern Analysis, Spatial based split

Number of taxis in use	Passenger wait time (min)	R
		u
		n
		t
		i
		n
		e
		(
		n
		i
		n
)

Existing condition	Spatial based		
	split		
S 22655	14677	13.57	3
e			1
p			.
8			6
S 23545	13797	13.66	3
e			0
p			.
9			5
S 24249	14349	13.70	3
e			5
p			.
1			2
0			
S 24616	14181	13.68	3
e			5
p			.
1			9
1			
S 24740	15434	13.60	3
e			4
p			.
			4

			4.6.3 Sensitive analysis
1			The results
2			
S 24774	15243	13.62	3 also demonstrate a
e			4 clear correlation
p			· between wait time
1			8 parameters and
3			fleet size
S 23578	13977	13.64	5 requirements, show
e			6 in Figure 4-6
p			· below. With a 5-
1			3 minute wait time
4			constraint, the

model calculated a required fleet of 8,289 taxis, which represents approximately 35.4% of the baseline fleet size of 23,439 taxis observed on January 9. When the wait time constraint was relaxed to 10 minutes, the required fleet size decreased to 8,155 taxis, and further relaxation to 15 minutes yielded a fleet requirement of 8,069 taxis. This suggests that increasing passenger wait time tolerance beyond 10 minutes provides diminishing returns in fleet size reduction, with only a 1.1% decrease in required vehicles between the 10-minute and 15-minute scenarios, compared to a 1.6% reduction between the 5-minute and 10-minute scenarios.

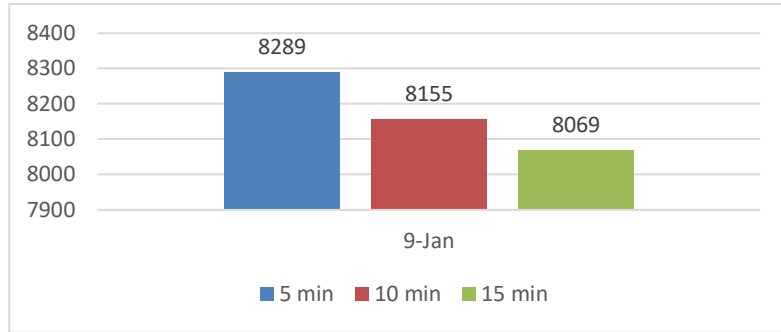


Figure 4-6 Wait time comparison of Jan 9 data

Overall, the findings of this study highlight the trade-offs between computational efficiency and vehicle usage in urban taxi dispatch systems. The time-based split emerged as the preferred solution for optimizing taxi utilization, balancing between computational cost and fleet efficiency effectively.

4.6.4 Discussion

Our comprehensive analysis across diverse operating scenarios demonstrates the robustness and efficiency of the proposed time-based split algorithm. Several key findings emerge:

1. Superior efficiency of time-based approach:

The time-based split consistently outperformed both existing conditions and the spatial split approach across all tested scenarios, achieving fleet reductions of 58-65% while maintaining acceptable service levels.

2. Scalability under peak demand:

When tested on the busiest day of the year, our algorithm required only a modest 10.5% increase in fleet size despite a 23.7% increase in trip volume, suggesting increasing returns to scale at higher demand levels.

3. Adaptability to demand volatility:

During periods of extreme demand fluctuation, the algorithm demonstrated excellent responsiveness, adjusting fleet size proportionally while maintaining consistent service quality.

4. Consistency across weekly patterns:

The algorithm effectively captured day-of-week variations in demand while providing stable service levels throughout the week, confirming its generalizability to typical operational cycles.

5. Computational efficiency:

The time-based split approach with $k=8$ significantly reduced computational requirements compared to baseline approaches, making real-time or near-real-time implementation feasible even for large-scale systems.

These findings suggest that the time-based split approach represents a promising direction for large-scale taxi fleet optimization, offering substantial efficiency gains across diverse operating conditions while maintaining practical computational requirements.

Chapter 5

CONCLUSION

5.1 Summary

This research investigated the optimization of taxi fleet operations through a novel time-based split algorithm aimed at reducing the number of vehicles required while maintaining adequate service levels. The study utilized comprehensive New York City taxi trip data from 2013, to evaluate algorithm performance under diverse operating conditions.

Our approach addressed two fundamental challenges: (1) the computational complexity of large-scale fleet optimization and (2) the need for robust performance across varying demand patterns. The proposed time-based split methodology partitions the trip data chronologically and processes these partitions in parallel, achieving substantial computational efficiency while preserving solution quality. We systematically compared this approach against both existing conditions and a spatial split alternative across multiple scenarios.

The experimental results demonstrate that the time-based split algorithm consistently outperforms existing operational practices, reducing fleet requirements by 58-65% across different scenarios while maintaining acceptable passenger wait times averaging 13.2 minutes. When applied to January 9th, 2013, our algorithm reduced the required fleet from 23,439 to 8,069 vehicles—a 65.6% reduction—while keeping computational time to just 5 minutes. This significantly outperformed the spatial split

approach, which required 12,936 vehicles and 21 minutes of computation time for the same dataset.

The algorithm's robustness was confirmed through testing under challenging conditions. On February 23rd, the busiest day of the year with 584,812 trips, our approach required only 8,914 taxis, demonstrating excellent scalability under peak demand. During the most volatile period (August 11-12, 2013), the algorithm adaptively adjusted fleet size from 3,988 to 7,385 vehicles in response to a dramatic demand surge, while maintaining consistent service quality. Weekly pattern analysis (September 8-14, 2013) further validated the approach's generalizability across typical operational cycles.

Several key insights emerged from our analysis. First, the time-based split consistently outperformed the spatial approach, highlighting the limitations of geographic constraints in urban taxi operations. Second, the algorithm demonstrated non-linear scaling properties, with efficiency gains becoming more pronounced at higher demand levels. Third, the parameter sensitivity analysis confirmed robust performance across various parameter configurations, with the selected values ($\alpha_1=100$, $\alpha_2=M$, $\alpha_3=10^4$, $\epsilon=15$ min, $k=8$) providing an optimal balance between computational efficiency, fleet size requirements, and service quality.

These findings suggest that the proposed time-based split approach offers a promising direction for large-scale taxi fleet optimization, providing substantial efficiency gains while remaining computationally tractable for real-world implementation.

5.2 Future Research

The findings from this research open several avenues for future research. One potential area is the exploration of hybrid models that combine both time-based and spatial-based allocation approaches. Such hybrid models could aim to leverage the advantages of both methods, potentially minimizing the inefficiencies observed in cross-zone travel while still optimizing the overall fleet utilization.

5.2.1 Integration of Additional Optimization Factors

Future research should expand the optimization model to incorporate several critical economic and operational factors that were beyond the scope of the current study. The current model optimizes primarily for fleet size and passenger wait time but does not account for variable trip pricing. Future work could integrate dynamic pricing models that adjust based on demand patterns, distance, time of day, and service level agreements. This would allow for analyzing how pricing strategies affect both system efficiency and economic viability while potentially creating more equitable access to transportation services across different neighborhoods.

A more comprehensive model should include granular fleet cost components. This would encompass fuel consumption and efficiency across different vehicle types, maintenance schedules and associated costs, vehicle depreciation based on usage patterns, insurance premiums that vary with service type, and parking costs that differ across urban zones. By incorporating these elements, the model could provide more realistic assessments of the true operational costs of taxi services in urban environments.

The current model does not explicitly maximize driver earnings or account for driver experience variability, representing significant research opportunities. Future

research could extend the model to balance system efficiency with driver income stability by incorporating various compensation models (hourly, per-trip, or hybrid approaches) while differentiating between novice and experienced drivers. This enhanced model could analyze how driver experience impacts route selection, travel times, and willingness to accept advanced scheduling, while simultaneously examining the relationship between deadhead time and earnings. Optimization could then address both driver-specific parameters reflecting route knowledge or professional tenure and shift patterns that maximize earning potential across driver segments, creating incentive structures that align driver behavior with system objectives while ensuring fair compensation based on experience level.

Environmental impact metrics present another compelling direction for future research. Building on the preliminary environmental considerations in this study, future models could incorporate CO₂ emissions calculations based on vehicle type, distance, and congestion levels. The optimization could also account for incentive structures that prioritize lower-emission vehicles within the dispatch algorithm and explicitly optimize for reduced total vehicle miles traveled across the system, supporting broader urban sustainability goals.

5.2.2 Real-Time Adjustments and Dynamic Optimization

Another important direction would be to incorporate real-time adjustments into the otherwise offline optimization model. A fixed taxi pool will be impacted by high volume days or high volatility days, making dynamic allocation capabilities essential. By integrating real-time data, such as traffic conditions or sudden demand surges, the system could be made even more responsive, thus improving its practical applicability in dynamic urban environments. This could involve developing hybrid offline-online

algorithms that start with prescheduled assignments but allow for real-time modifications as conditions change. Machine learning models could be developed to predict and respond to traffic anomalies, while dynamic rebalancing protocols could adjust vehicle distribution in response to emerging patterns throughout the day.

5.2.3 Policy and Socioeconomic Implications

Finally, the implications for policy and infrastructure development should be explored. Future research could investigate the broader social and economic impacts of adopting a fully reservation-based taxi system, including how it affects driver income stability, user accessibility, and city-wide traffic patterns. These analyses would provide valuable insights for policymakers and urban planners considering the adoption of such systems in other cities.

A particularly promising research direction would be to develop multi-objective optimization frameworks that simultaneously address service quality, environmental impact, economic viability, and social equity concerns. Such holistic models would better reflect the complex trade-offs faced by transportation planners and regulatory bodies in major urban centers. This integrated approach could help bridge the gap between theoretical optimization models and practical implementation challenges in real-world urban environments.

REFERENCES

- Acharya, R., Chen, J., & Xiao, H. (2024). *Uber Stable: Formulating the Rideshare System as a Stable Matching Problem* (arXiv:2403.13083). arXiv.
<http://arxiv.org/abs/2403.13083>
- Afrati, F., Bampis, E., Chekuri, C., Karger, D., Kenyon, C., Khanna, S., Milis, I., Queyranne, M., Skutella, M., Stein, C., & Sviridenko, M. (1999). Approximation schemes for minimizing average weighted completion time with release dates. *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, 32–43.
<https://doi.org/10.1109/SFFCS.1999.814574>
- Akpınar, S., Mirac Bayhan, G., & Baykasoglu, A. (2013). Hybridizing ant colony optimization via genetic algorithm for mixed-model assembly line balancing problem with sequence dependent setup times between tasks. *Applied Soft Computing*, *13*(1), 574–589. <https://doi.org/10.1016/j.asoc.2012.07.024>
- Arnaut, J.-P., Rabadi, G., & Musa, R. (2010). A two-stage Ant Colony Optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times. *Journal of Intelligent Manufacturing*, *21*(6), 693–701. <https://doi.org/10.1007/s10845-009-0246-1>

- Babaei, M., Schmöcker, J., Khademi, N., Ghaffari, A., & Naderan, A. (2016). Fixed-route taxi system: Route network design and fleet size minimization problems. *Journal of Advanced Transportation*, 50(6), 1252–1271.
<https://doi.org/10.1002/atr.1400>
- Bagloee, S. A., Tavana, M., Asadi, M., & Oliver, T. (2016). Autonomous vehicles: Challenges, opportunities, and future implications for transportation policies. *Journal of Modern Transportation*, 24(4), 284–303.
<https://doi.org/10.1007/s40534-016-0117-3>
- Bischoff, J., & Maciejewski, M. (2016). Simulation of City-wide Replacement of Private Cars with Autonomous Taxis in Berlin. *Procedia Computer Science*, 83, 237–244. <https://doi.org/10.1016/j.procs.2016.04.121>
- Bozer, Y. A., & Srinivasan, M. M. (1991). Tandem Configurations for Automated Guided Vehicle Systems and the Analysis of Single Vehicle Loops. *IIE Transactions*, 23(1), 72–82. <https://doi.org/10.1080/07408179108963842>
- Brucker, P., Drexl, A., Mo, R., & Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*.
- Brucker, P., Jurisch, B., & Sievers, B. (1994). A branch and bound algorithm for the job-shop scheduling problem. *Discrete Applied Mathematics*.
- Chang, F.-S., Wu, J.-S., Lee, C.-N., & Shen, H.-C. (2014). Greedy-search-based multi-objective genetic algorithm for emergency logistics scheduling. *Expert Systems*

with Applications, 41(6), 2947–2956.

<https://doi.org/10.1016/j.eswa.2013.10.026>

Chen, Z. (2018). *Traffic Flow and Density Analysis of NYC TLC Taxi Data*.

Chen, Z.-L. (2010). Integrated Production and Outbound Distribution Scheduling:

Review and Extensions. *Operations Research*, 58(1), 130–148.

<https://doi.org/10.1287/opre.1080.0688>

Christofides, N., & Eilon, S. (1969). *An Algorithm for the Vehicle-Dispatching Problem*. 20(3).

Cohen, I., Postek, K., & Shtern, S. (2023). An adaptive robust optimization model for parallel machine scheduling. *European Journal of Operational Research*,

306(1), 83–104. <https://doi.org/10.1016/j.ejor.2022.07.018>

Conway, R. W., Maxwell, W. L., & Miller, L. W. (1967). *Theory of Scheduling*.

Dover. https://books.google.com/books?id=Yr5_kQDa_ssC

Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. *Management*

Science, 6(1), 80–91. <https://doi.org/10.1287/mnsc.6.1.80>

Diez-Olivan, A., Del Ser, J., Galar, D., & Sierra, B. (2019). Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0.

Information Fusion, 50, 92–111. <https://doi.org/10.1016/j.inffus.2018.10.005>

Fang, K.-T., & Lin, B. M. T. (2013). Parallel-machine scheduling to minimize

tardiness penalty and power cost. *Computers & Industrial Engineering*, 64(1),

224–234. <https://doi.org/10.1016/j.cie.2012.10.002>

- Gawiejnowicz, S. (2020). A review of four decades of time-dependent scheduling: Main results, new topics, and open problems. *Journal of Scheduling*, 23(1), 3–47. <https://doi.org/10.1007/s10951-019-00630-w>
- Gendreau, M., Guertin, F., Potvin, J.-Y., & Séguin, R. (2006). Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. *Transportation Research Part C: Emerging Technologies*, 14(3), 157–174. <https://doi.org/10.1016/j.trc.2006.03.002>
- Gendreau, M., Guertin, F., Potvin, J.-Y., & Taillard, É. (1999). Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching. *Transportation Science*, 33(4), 381–390. <https://doi.org/10.1287/trsc.33.4.381>
- Geng, X., Li, Y., Wang, L., Zhang, L., Yang, Q., Ye, J., & Liu, Y. (2019). Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 3656–3663. <https://doi.org/10.1609/aaai.v33i01.33013656>
- Gillett, B. E., & Johnson, J. G. (1976). Multi-terminal vehicle-dispatch algorithm. *Omega*, 4(6), 711–718. [https://doi.org/10.1016/0305-0483\(76\)90097-9](https://doi.org/10.1016/0305-0483(76)90097-9)
- Gillett, B. E., & Miller, L. R. (1974). A Heuristic Algorithm for the Vehicle-Dispatch Problem. *Operations Research*, 22(2), 340–349. <https://doi.org/10.1287/opre.22.2.340>
- Goli, A., & Keshavarz, T. (2022). Just-in-time scheduling in identical parallel machine sequence-dependent group scheduling problem. *Journal of Industrial and Management Optimization*, 18(6), 3807. <https://doi.org/10.3934/jimo.2021124>

- Hancerliogullari, G., Rabadi, G., Al-Salem, A. H., & Kharbeche, M. (2013). Greedy algorithms and metaheuristics for a multiple runway combined arrival-departure aircraft sequencing problem. *Journal of Air Transport Management*, 32, 39–48. <https://doi.org/10.1016/j.jairtraman.2013.06.001>
- Horn, M. E. T. (2002). Fleet scheduling and dispatching for demand-responsive passenger services. *Transportation Research Part C: Emerging Technologies*, 10(1), 35–63. [https://doi.org/10.1016/S0968-090X\(01\)00003-1](https://doi.org/10.1016/S0968-090X(01)00003-1)
- Hsu, C.-I., Hung, S.-F., & Li, H.-C. (2007). Vehicle routing problem with time-windows for perishable food delivery. *Journal of Food Engineering*, 80(2), 465–475. <https://doi.org/10.1016/j.jfoodeng.2006.05.029>
- Huang, Q., Su, S., Li, J., Xu, P., Shuang, K., & Huang, X. (2012). Enhanced Energy-Efficient Scheduling for Parallel Applications in Cloud. *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgriid 2012)*, 781–786. <https://doi.org/10.1109/CCGrid.2012.49>
- Hwang K., Fox G. C., & Dongarra J. J. (2012). *Distributed and cloud computing: From parallel processing to the Internet of things*. Morgan Kaufmann.
- Ichoua, S., Gendreau, M., & Potvin, J.-Y. (2003). Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research*, 144(2), 379–396. [https://doi.org/10.1016/S0377-2217\(02\)00147-9](https://doi.org/10.1016/S0377-2217(02)00147-9)
- Ichoua, S., Gendreau, M., & Potvin, J.-Y. (2006). Exploiting Knowledge About Future Demands for Real-Time Vehicle Dispatching. *Transportation Science*, 40(2), 211–225. <https://doi.org/10.1287/trsc.1050.0114>

- Ivanov, D., Dolgui, A., Sokolov, B., Werner, F., & Ivanova, M. (2016). A dynamic model and an algorithm for short-term supply chain scheduling in the smart factory industry 4.0. *International Journal of Production Research*, *54*(2), 386–402. <https://doi.org/10.1080/00207543.2014.999958>
- Jin, S. T., Kong, H., & Sui, D. Z. (2019). Uber, Public Transit, and Urban Transportation Equity: A Case Study in New York City. *The Professional Geographer*, *71*(2), 315–330. <https://doi.org/10.1080/00330124.2018.1531038>
- Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, *1*(1), 61–68. <https://doi.org/10.1002/nav.3800010110>
- Jung, J., Jayakrishnan, R., & Park, J. Y. (2016). Dynamic Shared-Taxi Dispatch Algorithm with Hybrid-Simulated Annealing. *Computer-Aided Civil and Infrastructure Engineering*, *31*(4), 275–291. <https://doi.org/10.1111/mice.12157>
- Karp, R. M. (1975). *On the Computational Complexity of Combinatorial Problems*.
- Kim, D.-W., Kim, K.-H., Jang, W., & Frank Chen, F. (2002). Unrelated parallel machine scheduling with setup times using simulated annealing. *Robotics and Computer-Integrated Manufacturing*, *18*(3–4), 223–231. [https://doi.org/10.1016/S0736-5845\(02\)00013-3](https://doi.org/10.1016/S0736-5845(02)00013-3)
- Kravchenko, S. A., & Werner, F. (1997). Parallel machine scheduling problems with a single server. *Mathematical and Computer Modelling*, *26*(12), 1–11. [https://doi.org/10.1016/S0895-7177\(97\)00236-7](https://doi.org/10.1016/S0895-7177(97)00236-7)

- Larsen, A., Madsen, O., & Solomon, M. (2002). Partially dynamic vehicle routing—
Models and algorithms. *Journal of the Operational Research Society*, 53(6),
637–646. <https://doi.org/10.1057/palgrave.jors.2601352>
- Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., & Shmoys, D. B. (1993).
Chapter 9 Sequencing and scheduling: Algorithms and complexity. In
Handbooks in Operations Research and Management Science (Vol. 4, pp.
445–522). Elsevier. [https://doi.org/10.1016/S0927-0507\(05\)80189-6](https://doi.org/10.1016/S0927-0507(05)80189-6)
- Lee, C.-Y., & Chen, Z.-L. (2001). Machine scheduling with transportation
considerations. *Journal of Scheduling*, 4(1), 3–24.
[https://doi.org/10.1002/1099-1425\(200101/02\)4:1<3::AID-JOS57>3.0.CO;2-D](https://doi.org/10.1002/1099-1425(200101/02)4:1<3::AID-JOS57>3.0.CO;2-D)
- Lenstra, J. K., Rinnooy Kan, A. H. G., & Brucker, P. (1977). Complexity of Machine
Scheduling Problems. In *Annals of Discrete Mathematics* (Vol. 1, pp. 343–
362). Elsevier. [https://doi.org/10.1016/S0167-5060\(08\)70743-X](https://doi.org/10.1016/S0167-5060(08)70743-X)
- Low, Y., Gonzalez, J., & Kyrola, A. (2014). *GraphLab: A New Framework For
Parallel Machine Learning*.
- Monma, C. L., & Sidney, J. B. (1979). Sequencing with Series-Parallel Precedence
Constraints. *Mathematics of Operations Research*, 4(3), 215–224.
<https://doi.org/10.1287/moor.4.3.215>
- Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., & Damas, L. (2013).
Predicting Taxi–Passenger Demand Using Streaming Data. *IEEE Transactions
on Intelligent Transportation Systems*, 14(3), 1393–1402.
<https://doi.org/10.1109/TITS.2013.2262376>

- Munier, A., Queyranne, M., & Schulz, A. S. (1998). *Approximation Bounds for a General Class of Precedence Constrained Parallel Machine Scheduling Problems*.
- Muth, J. F., Thompson, G. L., & Winters, P. R. (1963). *Industrial scheduling*. (No Title).
- Pan, Q.-K., Gao, L., Li, X.-Y., & Gao, K.-Z. (2017). Effective metaheuristics for scheduling a hybrid flowshop with sequence-dependent setup times. *Applied Mathematics and Computation*, 303, 89–112.
<https://doi.org/10.1016/j.amc.2017.01.004>
- Pinedo, M. (2005). *Planning and scheduling in manufacturing and services*. Springer.
- Pinedo, M. L. (2016). *Scheduling* (Fifth Edition). Springer International Publishing.
<https://doi.org/10.1007/978-3-319-26580-3>
- Potvin, J.-Y., Xu, Y., & Benyahia, I. (2006). Vehicle routing and scheduling with dynamic travel times. *Computers & Operations Research*, 33(4), 1129–1137.
<https://doi.org/10.1016/j.cor.2004.09.015>
- Psaraftis, H. N. (1995). Dynamic vehicle routing: Status and prospects. *Annals of Operations Research*, 61(1), 143–164. <https://doi.org/10.1007/BF02098286>
- Qarke, G. (1962). *SCHEDULING OF VEHICLES FROM A CENTRAL DEPOT TO A NUMBER OF DELIVERY POINTS*.
- Raman, N., Shah, S., & Dickerson, J. (2021). *Data-Driven Methods for Balancing Fairness and Efficiency in Ride-Pooling* (arXiv:2110.03524). arXiv.
<http://arxiv.org/abs/2110.03524>

- Schaller, B. (2021). Can sharing a ride make for less traffic? Evidence from Uber and Lyft and implications for cities. *Transport Policy*, *102*, 1–10.
<https://doi.org/10.1016/j.tranpol.2020.12.015>
- Schmid, V. (2012). Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming. *European Journal of Operational Research*, *219*(3), 611–621.
<https://doi.org/10.1016/j.ejor.2011.10.043>
- Schneider, T. W. (2018). *Analyzing 1.1 Billion NYC Taxi and Uber Trips, with a Vengeance—Todd W. Schneider*.
- Shen, Y., Zhang, H., & Zhao, J. (2018). Integrating shared autonomous vehicle in public transportation system: A supply-side simulation of the first-mile service in Singapore. *Transportation Research Part A: Policy and Practice*, *113*, 125–136. <https://doi.org/10.1016/j.tra.2018.04.004>
- Sundt, A., Luo, Q., Vincent, J., Shahabi, M., & Yin, Y. (2021). *Heuristics for Customer-focused Ride-pooling Assignment* (arXiv:2107.11318). arXiv.
<http://arxiv.org/abs/2107.11318>
- Talbi, E.-G. (2016). Combining metaheuristics with mathematical programming, constraint programming and machine learning. *Annals of Operations Research*, *240*(1), 171–215. <https://doi.org/10.1007/s10479-015-2034-y>
- Tarduno, M. (2021). The congestion costs of Uber and Lyft. *Journal of Urban Economics*, *122*, 103318. <https://doi.org/10.1016/j.jue.2020.103318>

- Taxi & Limousine Commission. (2022). *TLC Trip Record Data—TLC*.
<https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>
- Taxi and Limousine Commission (TLC). (2023). *NYC Taxi Zones | NYC Open Data*.
<https://data.cityofnewyork.us/Transportation/NYC-Taxi-Zones/d3c5-ddgc>
- TLC Rules and Local Laws*. (2025, March). <https://www.nyc.gov/site/tlc/about/tlc-rules.page>
- TLC Trip Record Data—TLC*. (2022). <https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>
- Tosoni, F., Ferragina, P., Marino, A., Resta, G., & Santi, P. (2020). Algorithms and Data Structures for Efficient Ride Sharing Platforms. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*.
- Tucker, T., Marra, M., & Friedman, J. M. (2009). Massively Parallel Sequencing: The Next Big Thing in Genetic Medicine. *The American Journal of Human Genetics*, 85(2), 142–154. <https://doi.org/10.1016/j.ajhg.2009.06.022>
- Vallada, E., & Ruiz, R. (2011). A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *European Journal of Operational Research*, 211(3), 612–622.
<https://doi.org/10.1016/j.ejor.2011.01.011>
- Vazifeh, M. M., Santi, P., Resta, G., Strogatz, S. H., & Ratti, C. (2018). Addressing the minimum fleet problem in on-demand urban mobility. *Nature*, 557(7706), 534–538. <https://doi.org/10.1038/s41586-018-0095-1>

- Verderame, P. M., Elia, J. A., Li, J., & Floudas, C. A. (2010). Planning and Scheduling under Uncertainty: A Review Across Multiple Sectors. *Industrial & Engineering Chemistry Research*, 49(9), 3993–4017.
<https://doi.org/10.1021/ie902009k>
- Wang, X., Poikonen, S., & Golden, B. (2017). The vehicle routing problem with drones: Several worst-case results. *Optimization Letters*, 11(4), 679–697.
<https://doi.org/10.1007/s11590-016-1035-3>
- Wu, X., & Che, A. (2019). A memetic differential evolution algorithm for energy-efficient parallel machine scheduling. *Omega*, 82, 155–165.
<https://doi.org/10.1016/j.omega.2018.01.001>
- Khafa, F., & Abraham, A. (Eds.). (2008). *Metaheuristics for Scheduling in Industrial and Manufacturing Applications* (Vol. 128). Springer Berlin Heidelberg.
<https://doi.org/10.1007/978-3-540-78985-7>
- Yang, Y., Yuan, Z., Fu, X., Wang, Y., & Sun, D. (2019). Optimization Model of Taxi Fleet Size Based on GPS Tracking Data. *Sustainability*, 11(3), 731.
<https://doi.org/10.3390/su11030731>
- Yao, B., Jin, L., Cao, Q., Gao, J., & Zhang, M. (2016). Fleet size and fare optimization for taxi under dynamic demand. *Journal of Transport Literature*, 10(4), 45–50.
<https://doi.org/10.1590/2238-1031.jtl.v10n4a9>
- Zhang, W., & Ukkusuri, S. V. (2016). Optimal Fleet Size and Fare Setting in Emerging Taxi Markets with Stochastic Demand. *Computer-Aided Civil and*

Infrastructure Engineering, 31(9), 647–660.

<https://doi.org/10.1111/mice.12203>

Zhou, Z., Roncoli, C., & Sipetas, C. (2023). Optimal matching for coexisting ride-hailing and ridesharing services considering pricing fairness and user choices.

Transportation Research Part C: Emerging Technologies, 156, 104326.

<https://doi.org/10.1016/j.trc.2023.104326>