CENTER FOR CONNECTED AND
AUTOMATED TRANSPORTATION

ICT-24-029
December 2024

Planning and Dynamic Management of Autonomous Modular Mobility Services

Shiyu Shen
Yuhui Zhai
Yanfeng Ouyang

ILLINOIS CENTER FOR
TRANSPORTATION

DISCLAIMER

Funding for this research was provided by the Center for Connected and Automated Transportation under Grant No. 69A3552348301 of the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (OST-R), University Transportation Centers Program. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Suggested APA Format Citation:

Shen, S., Zhai, Y., & Ouyang, Y. (2024). *Planning and dynamic management of autonomous modular mobility services* (Report No. ICT-24-029). Illinois Center for Transportation. https://doi.org/10.36501/0197-9191/24-029

Contacts

For more information:

Yanfeng Ouyang
University of Illinois Urbana Champaign
2129 E Newmark Civil Engineering Bldg
yfouyang@illinois.edu
https://ict.illinois.edu

CCAT
University of Michigan Transportation Research Institute
2901 Baxter Road
Ann Arbor, MI 48152
(734) 763-2498
uumtri-ccat@umich.edu
www.ccat.umtri.umich.edu

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. ICT-24-029	**2. Government Accession No.** N/A	**3. Recipient's Catalog No.** N/A

4. Title and Subtitle Planning and Dynamic Management of Autonomous Modular Mobility Services	**5. Report Date** December 2024
	6. Performing Organization Code N/A
7. Authors Shiyu Shen (https://orcid.org/0000-0002-0704-8766), Yuhui Zhai (https://orcid.org/0009-0008-0666-180X), and Yanfeng Ouyang (https://orcid.org/0000-0002-5944-2044)	**8. Performing Organization Report No.** ICT-24-029 UILU-ENG-2024-2029
9. Performing Organization Name and Address Illinois Center for Transportation Department of Civil and Environmental Engineering University of Illinois Urbana-Champaign 205 North Mathews Avenue, MC-250 Urbana, IL 61801	**10. Work Unit No.** N/A
	11. Contract or Grant No. Grant No. 69A3552348301
12. Sponsoring Agency Name and Address Center for Connected and Automated Transportation University of Michigan Transportation Research Institute 2901 Baxter Road Ann Arbor, MI 48152	**13. Type of Report and Period Covered** Final Report
	14. Sponsoring Agency Code

15. Supplementary Notes

Funding under Grant No. 69A3552348301 U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (OST-R), University Transportation Centers Program. https://doi.org/10.36501/0197-9191/24-029

16. Abstract

As we enter the next era of autonomous driving, robo-vehicles (which serve as low-cost and fully compliant drivers) are replacing conventional chauffeured services in the mobility market. During just the last few years, companies like Waymo Inc. and Cruise Inc. have already offered fully driverless robo-taxi services to the general public in cities like Phoenix and San Francisco. The rapid evolution of autonomous vehicles is anticipated to reshape the shared mobility market very soon. This project aims to address the following open questions. At the operational level, how should modular units be allocated across multiple categories of customers (e.g., passenger and freight cabins), and how should they be matched in real time? How do we enhance system efficiency by dynamic relocation and swap of modular chassis? At the strategic or tactical level, how should the rolling stock resources (modular chassis, passenger and freight cabins) be planned, and where shall chassis swapping sites be located? How could any potential transaction cost for a chassis swap, such as the time required for a modular chassis to be assembled with a customized cabin, affect the optimal strategy and system performance? How can customer priorities (e.g., passenger vs. freight) affect system performance, and how can service providers manage demand by specific pricing scheme or discriminative customer service strategies? We conducted the following research tasks: (i) analytically derived systems of implicit nonlinear equations in the closed form, including a set of differential equations, to analyze the modular autonomous mobility system and to estimate the expected system performance in the steady state; (ii) conducted a series of agent-based simulation experiments to verify the accuracy of the proposed analytical formulas and to demonstrate the effectiveness of the proposed modular chassis services; and (iii) designed policy instruments to enhance transportation system performance.

17. Key Words Random Bipartite Matching, Discrete Networks, Shared Mobility, Vehicle Swapping, Modular Chassis, Autonomous Vehicle	**18. Distribution Statement** No restrictions. This document is available through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report) Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages** 60 + appendices	**22. Price** N/A

ACKNOWLEDGMENT, DISCLAIMER, MANUFACTURERS' NAMES

This project was conducted in cooperation with the Center for Connected and Automated Transportation and the Illinois Center for Transportation. The contents of this report reflect the view of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of CCAT or ICT. This report does not constitute a standard, specification, or regulation. Trademark or manufacturers' names appear in this report only because they are considered essential to the object of this document and do not constitute an endorsement of the product by CCAT or ICT.

EXECUTIVE SUMMARY

As we enter the next era of autonomous driving, robo-vehicles (which serve as low-cost and fully compliant drivers) are replacing conventional chauffeured services in the mobility market. During just the last few years, companies like Waymo Inc. and Cruise Inc. have already offered fully driverless robo-taxi services to the general public in cities like Phoenix and San Francisco. The rapid evolution of autonomous vehicles is anticipated to reshape the shared mobility market very soon.

This project aimed to address the challenges faced by on-demand mobility operators in understanding and addressing spatiotemporal random bipartite matching problems (ST-RBMPs). At the planning level, we developed analytical models to estimate the expected system performance in a static RBMP. At the operational level, we designed solution algorithms to improve the overall service efficiency in ST-RBMPs with different types of supply arrivals. Although our main focus was on the application of on-demand mobility services, these models can also be applied to other contexts such as resource allocation, target detection, etc. This project aimed to address the following research objectives:

1. Propose an analytical model with closed-form formulas (without statistical curve fitting) that estimate the expectation of the optimal matching distance for static RBMP, where the bipartite vertices are distributed randomly over a discrete network. These formulas can be incorporated into queuing and optimization models to identify the best operational strategies in on-demand mobility systems with closed- or open-loop resource arrivals. It helps determine the optimal decision timing for whether newly arriving customers should be matched instantly or pooled into a batch for matching.

2. For ST-RBMPs with closed-loop resources, where arriving customers shall be matched instantly, the objective is to propose a Pareto-improving strategy that allows matched vertices to be swapped among candidates with improved matching distances as the system evolves. This strategy could enhance system efficiency by reducing the overall expected matching distance and mitigating the so-called Wild Goose Chasing (WGC) phenomenon. Approximate analytic formulas can be derived from a series of differential equations and spatial probability models to estimate the expected system performance in the steady state.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

ON-DEMAND MOBILITY SYSTEMS

A Spectrum of Services

The demand of a traveler is usually characterized by their origin and destination (OD) and desired time of travel. A spectrum of mobility systems has been designed to meet such demand, as illustrated in Figure 1. These systems range from highly flexible options with low occupancy to options with higher occupancy but lower flexibility. The most flexible options, such as auto-driving and taxis, are used for individual travel. They can take each traveler directly from the origin to destination with minimal delay. Built upon the taxi service, ride-sharing serves as an intermittent option between individual and collective travels. One traveler may need to share a ride with one or two others, which may result in a slight delay due to detours. More collective services typically require travelers to share rides with considerably more people, making them less flexible and often resulting in more delays. However, these services offer higher economies of scale. (Daganzo & Ouyang, 2019b). For example, conventional transit services offer the highest economies of scale. They are usually highly structured, with fixed routes and schedules, and require all travelers to follow certain predetermined service rules. As an option between ride-sharing and conventional transit, flexible transit offers a balance between the efficiency of having fixed routes and the convenience of allowing flexible deviations. It can be designed either as ride-sharing with higher occupancy or as conventional transit with flexible routes or schedules. These flexible options are designed to better meet the needs of travelers in real time and fall into the same category as “on-demand” (or “demand-responsive”) mobility systems.

The advancement of information and communication technologies has revolutionized the on-demand mobility industry toward one that is more accessible, flexible, and efficient. Transportation network companies (TNCs), such as Uber, Lyft, and DiDi, have transformed many conventional systems into a variety of application-based shared mobility services, such as e-hailing taxis (Salanova et al., 2011), ride-pooling (Agatz et al., 2012), shared bikes (Ricci, 2015), e-scooters (Wang et al., 2022), customized buses (Shen et al., 2021a), and demand-responsive transit (Shen et al., 2021b). Compared to older systems such as “dial-a-ride” (Daganzo, 1978), these services achieve greater efficiency by capturing more data and employing more advanced computing methods to match the demand and supply in real time. In recent years, third-party “service integrators” have emerged in the mobility market. Their platforms serve as a marketplace that lists a pool of services and resources from multiple TNCs, such that a passenger can select the best service among participating companies. For example, Baidu Map in China has been integrating DiDi and several other e-hailing service operators (Zhou et al., 2022b). By allowing matching between the larger pool of demand (from multiple customer groups) and resources (from multiple service operators), such integrated services hold the promise to further enhance the overall mobility service quality and achieve larger economies of scale/scope.

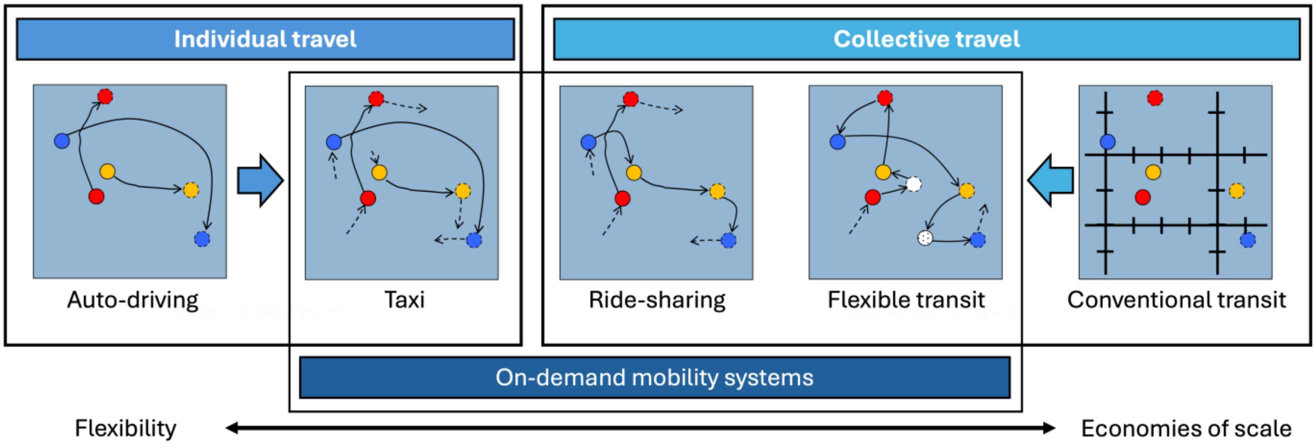


Figure 1. Graph. A spectrum of mobility services (Daganzo & Ouyang, 2019b).

As we enter the next era of autonomous driving, robo-vehicles (which serve as low-cost and fully compliant drivers) are replacing conventional chauffeured services in the mobility market. During just the last few years, robo-taxis (Li et al., 2022) and robo-buses (Varisteas et al., 2021) have been adopted worldwide for road tests, pilot projects, and even open public services. For example, since 2018, Baidu Inc. and WeRide Inc. have been implementing robo-taxi and robo-bus services in several Chinese mega-cities such as Beijing, Guangzhou, and Chongqing (Cheng, 2022; Harper, 2020). Meanwhile, in the U.S., companies like Waymo Inc. and Cruise Inc. have already offered fully driverless robo-taxi services to the general public in cities like Phoenix and San Francisco. (Cusack, 2021; Kolodny, 2022). Not only for passenger transportation, autonomous driving vehicles (e.g., delivery robots and drones) have also been widely used for on-demand freight transportation, such as in the small parcel delivery industry (e.g., for food and groceries). The rapid evolution of autonomous vehicles is anticipated to reshape the on-demand mobility market in the very near future.

Very recently, at the nexus of sharing and autonomy, a new technology called modularized vehicle platforms or modular chassis, has been explored by many automotive start-up companies, such as PixMoving Inc. (Banks, 2020) and REE Inc. (Gardner, 2021). Modular chassis is designed with built-in power and control systems to move independently and autonomously, and it can carry multiple types of customized cabins (e.g., for passenger or freight shipments). The customized cabins can be easily loaded onto (or unloaded from) a modular chassis, possibly at a convenient place with simple navigation guidance devices (e.g., roadside parking space), in a very short time (e.g., about 30–60 seconds). These modular chassis and customized cabins, just like intermodal trucks and containers, will undoubtedly help smooth service operations and enable new mobility solutions for the transportation industry. For example, in cities facing both real-time passenger and freight shipment needs (e.g., ride hailing and on-demand package delivery), traditionally, two dedicated vehicle fleets from two companies (e.g., Uber and FedEx) must be deployed for these customers. With modular chassis, passenger and freight service providers can merge their modular chassis (owned, or possibly rented from a third party). The chassis will function like mobility service vehicles pooled by third-party integrator platforms to serve multiple types of customers that are traditionally served separately.

System Design and Models

Researchers and practitioners have been actively designing various service variations to improve the operational efficiency and customer experience of these mobile systems. The system design usually consists of two aspects: the design of service protocols (e.g., service area, fleet size) at the planning level and the design of real-time control strategies (e.g., dispatch algorithms) at the operational level.

For the most flexible service types, such as robo-taxis or micromobility systems, the fundamental challenge is to find the best matches between travel demand and resource supply. While system design in this area primarily focuses on real-time operational-level problems, planning-level problems are also significant, as they could determine the upper-bound performance of any real-time decisions. This raises the need to better understand the system dynamics between demand and supply over time (e.g., a planning horizon) and space (e.g., a service region). The problem that best captures these underlying system dynamics is the random (or stochastic) bipartite matching problem. In a typical taxi operation, customers and vehicles evolve in the system as random points in a service region, and the service platform periodically (e.g., every a few seconds) makes vehicle dispatch and allocation decisions to best serve the customers. In each decision epoch, the system captures a snapshot of its current state to gather information on both idle vehicles (e.g., locations) and new customers (e.g., origin and destination locations and the elapsed waiting time). A bipartite graph can be constructed where one subset of vertices include all idle vehicles, and the other subset includes all new customer origins. Weights of the edges could be based on distance (or travel cost, time) and the customers' priority. Matches are then optimized by the platform based on a predefined objective, such as minimizing the total matching distances for pickups (between the vehicles and the customers' origins). Any unmatched customers either are assumed lost or could be retained and moved into the customer pool for the next decision epoch. This bipartite matching scheme stands out for its ease of computation and implementation. The next section provides an overview of bipartite matching problems.

BIPARTITE MATCHING PROBLEMS

The bipartite matching problem is a fundamental problem in the field of applied mathematics and combinatorial optimization (Asratian et al., 1998). In a bipartite graph, vertices are divided into two distinct subsets, and edges exist only between vertices of different subsets. The objective is to identify an optimal subset of these edges that match the vertices into disjoint pairs (i.e., no two selected edges share a common vertex). The most common static version of the bipartite matching problem has multiple types of variations. The most well-known one might be the maximum/minimum weight bipartite matching problem, where each edge carries a weight, and we seek the matching with the maximum/minimum total weight. If the two subsets of vertices have equal cardinality, we refer to this bipartite graph as balanced.

A matching is considered perfect if it covers every vertex; otherwise, if the matching covers only one subset of vertices in an unbalanced bipartite graph, it is said to saturate that particular subset of vertices. Each static problem instance can be solved quickly using linear programming methods or algorithms. For example, many well-known combinatorial optimization algorithms, such as the Hungarian algorithm, (Kuhn, 1955), Jonker–Volgenant algorithm (Jonker and & Volgenant, 1987a), and their variations, can generate near-optimal solutions in polynomial time. Even those more advanced machine-learning based algorithms, as reviewed in (Zhang et al., (2023), can effectively solve these problems within a relatively short time. These state-of-the-art computational techniques are sufficient to be implemented for real-time operational purposes.

Another notable family of bipartite matching problems that has received significant attention is the online bipartite matching problem (Mehta, 2013). Unlike in a static bipartite graph, where both sets of vertices and the connecting edges are revealed beforehand, in a classic online bipartite matching problem, one set of vertices is known in advance while the other set arrives sequentially over time. As each vertex arrives, a decision must be made immediately on whether to match it with an available vertex from the other set or leave it unmatched. This version of the problem, in its various forms, is more challenging than the static one because of the lack of future information. The simplest strategy, like the greedy algorithm, matches each incoming vertex to any available vertex. Slightly more sophisticated strategies, such as the ranking algorithm, assign random ranks to vertices and match incoming vertices based on these ranks. (Karp et al., 1990). Batching algorithms, (Feng and & Niazadeh, 2020; Ashlagi et al., 2023), which aggregate multiple incoming vertices before making matching decisions, are also being explored to improve efficiency and outcomes. This is still an evolving field, with many more algorithms being proposed to address new problem variations (Fahrbach et al., 2022; Shanks et al., 2023; Liang et al., 2023).

In addition to focusing on solving a single problem instance, another significant aspect of studying these problems deals with scenarios where the vertex arrivals or the weights on edges between them are generated according to certain probabilistic distribution. Researchers study the probability or expectation of certain properties in these random graphs, such as the expected size of the matching, the probability of successful matches, and expected matching distance. For example, Mézard and Parisi (1988) and their subsequent studies investigated random bipartite matching problems and derived the expected optimal matching distance for many problem variations, as detailed in Chapter 2. This type of analysis is valuable for understanding the behavior of systems in uncertain environments. Researchers typically employ techniques such as probabilistic or statistical models (Frieze and & Karoński, 2015) to identify the average properties of these random graphs. These models, especially analytical ones, are instrumental in designing and evaluating algorithms aimed at achieving certain objectives.

These variations of the bipartite matching problem are very versatile, and they have been applied to a variety of theoretical or practical challenges. In the field of physics, they can be used to capture important properties of various disordered complex systems, such as identifying the patterns and energy configurations of atomic magnets in spin glass systems (Mézard & Parisi, 1985). In the field of biology, the problem can be used to describe interactions between species in an ecosystem (Simmons et al., 2019) or to analyze pairwise protein-protein interactions (Tanay et al., 2004). In the field of computer science, similar matching problems are formulated for graph-based pattern recognition systems to map the underlying data structures of images/signals to their features/labels (Yu et al., 2020); or for emerging social media and e-commerce platforms to capture user/information interactions among distinct socioeconomic groups (Zhou et al., 2007; Wu et al., 2022).

SPATIOTEMPORAL MATCHING IN ON-DEMAND MOBILITY SYSTEMS

On-demand mobility systems face a special variation of the bipartite matching problem, which we refer to as the spatiotemporal random bipartite matching problem (ST-RBMP). Compared to traditional bipartite matching problems, ST-RBMPs have three distinct features: (i) bipartite vertices (e.g., demand and supply) are spatially dispersed, potentially across different dimensions, with edge weights representing distances between vertices; (ii) both sets of vertices are dynamically revealed over time following specific random distributions; and (iii) some or all vertices in one set (e.g., supply) may be reusable and return to the system in the future, with the timing of return influenced by the current matching decisions.

Specifically, on-demand mobility problems typically arise in one to three dimensions. In one-dimensional space, it can be used to manage multiple elevators in a tall building where customers arrive randomly at various floors and need to be matched with an available elevator. In two- or three-dimensional spaces, these problems can be used to describe how surface courier vehicles, idle taxis, freight drones, or passenger aerial vehicles are matched with customers within a city. The distance between any supply (e.g., vehicle) and demand (e.g., customer) points can be measured using various metrics such as Manhattan or Euclidean, depending on the specific structure of the underlying network. Regarding the arrival dynamics of supply-and-demand vertices, two types of systems can be considered: an open-loop system without vehicle conservation, where new idle vehicles could arrive from outside the system independently, possibly resembling services with freelance drivers, and a closed-loop system with a fixed fleet of vehicles, representing services with full-time drivers or robo-taxis. In a closed-loop system, vehicles return to the fleet after completing a service, and the duration of their availability is determined by their pickup times—the shorter the pickup time, the sooner the vehicle can be redeployed within the system.

Similar to other mobility systems, on-demand mobility operators primarily face two tasks: one at the operational level and the other at the planning level. At the operational level, they need to solve for the online ST-RBMP. Detailed matching solutions are often given in real time or in a rolling horizon via mathematical programming methods—for example, bipartite matching, (Xu et al., 2018), dynamic programs (Psaraftis et al., 2016), or meta-heuristics (Herbawi and & Weber, 2012; Najmi et al., 2017; Aydin et al., 2020). However, operators still face challenges in solving these problems.

Regardless of the matching strategies, the inherent pitfalls associated with dynamic decision-making in a stochastic setting dictate that many e-hailing–based shared mobility systems still suffer from the so-called wild-goose-chase (WGC) phenomenon (Arnott, 1996; Castillo et al., 2017), which describes an inefficient system equilibrium with a large number of service vehicles trapped in unproductive deadheading (for customer pickups). In this situation, because few vehicles are ready to provide service, customers will experience long waiting times before pickup, and the system efficiency is compromised (Daganzo, 1978a, 2010; Daganzo and & Ouyang, 2019a). A range of strategies have been proposed to mitigate WGC for taxi systems—for example, via zone-based surge pricing (Castillo et al., 2017; Zha et al., 2018), path-based surge pricing and dispatching, (Lei et al., 2019; Shen et al., 2021b), imposition of maximum radius and maximum waiting time for vehicle-passenger matching (Xu et al., 2020; Valadkhani and & Ramezani, 2020), repositioning of idle vehicles (Ke et al., 2021; Wang and & Wang, 2020), and/or discriminative customer service (Afèche et al., 2018). However, none of these strategies can fully resolve the WGC. More dynamic matching strategies are continually being proposed in this area.

At the planning level, operators usually need to estimate the service efficiency under a large number of possible realizations of supply-and-demand scenarios (e.g., different vehicle and customer distributions), rather than finding exact vehicle–customer matches for one particular problem instance. The average matching distance between the vehicles and the customer origins, commonly referred to as the deadheading distance, stands as a key indicator of service efficiency. It indicates the “unproductive” efforts made by both customers (i.e., waiting for pickup) and vehicles (i.e., running empty) within the mobility system. Understanding the relationship between the average matching distance and vehicle–customer distribution can help improve service efficiency in many ways. Operators, for example, often need to set standards for operation, such as determining the time between consecutive decision epochs (i.e., pooling interval). A longer pooling interval may lead to more customers/vehicles appearing in one matching problem instance, potentially reducing the resulting matching distance. However, it also implies that customers need to wait longer to find a match. Finding a balance between these conflicting objectives and identifying the optimal operational standard require knowledge of this quantitative relationship. Moreover, operators often deploy new tactical-level strategies to further enhance their service efficiency, such as the ones proposed in Chapter 3. Analyzing the effectiveness of these strategies (often measured by the reduction in matching distance) under various vehicle/customer distributions also requires such knowledge.

To the best of our knowledge, estimating the expected “optimal matching distance” for RBMP remains a challenging task. While each random realization of RBMP can be addressed as a deterministic bipartite matching problem, and one could use state-of-the-art techniques (as those employed by the TNCs) to solve a sufficiently large number of problem instances and produce statistical/simulated results, this process may pose computational challenges and consume considerable time. Moreover, the outcomes may lack the depth of analytical insights. In many cases, analytical models are favored for their efficiency, and they can provide more analytical insights compared to simulated results, such as those developed in (Daganzo et al., (2020) and (Ouyang et al., (2021) for estimating several key performance metrics (e.g., the expected vehicle distance traveled) for mobility services given certain operational standards. Moreover, this type of analytical model can be incorporated into the development of more comprehensive optimization/equilibrium models, helping operators or regulators in optimizing their service offerings to achieve higher service efficiency or social welfare (Zha et al., 2016; Ouyang et al., 2021; Liu and & Ouyang, 2021, 2023).

RESEARCH OBJECTIVES

This project aims to address the challenges faced by on-demand mobility operators in understanding and addressing the ST-RBMPs. At the planning level, we develop analytical models to estimate the expected system performance in a static RBMP. At the operational level, we design solution algorithms to improve the overall service efficiency in ST-RBMPs with different types of supply arrivals. Although our main focus is on the application of on-demand mobility services, these models can also be applied to other contexts such as resource allocation, target detection, etc. This project aims to address the following research objectives:

1. Propose an analytical model with closed-form formulas (without statistical curve fitting) that estimate the expectation of the optimal matching distance for static RBMP, where the bipartite vertices are randomly distributed over a discrete network. These formulas can be incorporated into queuing and optimization models to identify the best operational strategies in on-demand mobility systems with closed- or open-loop resource arrivals. It helps determine the optimal decision timing for whether newly arriving customers should be instantly matched or pooled into a batch for matching.

2. For ST-RBMPs with closed-loop resources, where arriving customers shall be matched instantly, the objective is to propose a Pareto-improving strategy that allows matched vertices to be swapped among candidates with improved matching distances as the system evolves. This strategy could enhance system efficiency by reducing the overall expected matching distance and mitigating the WGC. Approximate analytic formulas can be derived from a series of differential equations and spatial probability models to estimate the expected system performance in the steady state.

OUTLINE

The remainder of this report is organized as follows. Chapter 2 presents the distance formulas for estimating the average optimal matching distance in a static RBMP in discrete networks. Chapter 3 introduces a Pareto-improving swapping strategy designed for ST-RBMP with closed-loop resource arrivals and analyzes its performance in the steady state.

CHAPTER 2: AVERAGE DISTANCE OF RANDOM BIPARTITE MATCHING IN DISCRETE NETWORKS

INTRODUCTION

This chapter presents a closed-form distance formula estimation for random bipartite matching in discrete networks. In the field of transportation, bipartite matching problems are widely applicable (e.g., to find optimal matches between supply and demand points over continuous time and space). For example, in two-dimensional spaces, it can model how surface vehicles (e.g., taxis) are matched to customers, such as in ride-hailing or food delivery systems (Shen et al., 2024; Tafreshian and & Masoud, 2020). In three-dimensional spaces, it can dynamically dispatch and reposition aerial vehicles (drones) for delivering goods in the air (Aloqaily et al., 2022) or to optimize the flying trajectories of drone swarms during take-off and landing (Hernández et al., 2021). In addition, bipartite matches that span the time dimension can be useful for finding optimal schedules among a series of tasks (Ding et al., 2021; Afèche et al., 2022), such as optimizing container transshipment among freight trains (Fedtke and & Boysen, 2017) or minimizing customer/vehicle waiting in reservation-based ride-sharing services (Shen and & Ouyang, 2023).

Strictly speaking, bipartite matching applications in the transportation field are likely to be associated with a sparse discrete network, where supply and demand points are distributed along network edges (after ignoring the local access legs) (Abeywickrama et al., 2022). A special case would be on one-dimensional transportation routes or corridors. For instance, it can model the operations of drones that are used to deliver goods to customers located on different floors of a tall building (Ezaki et al., 2024; Seth et al., 2023), or vehicle–customer matching for a ride-hailing system on a single city corridor (Panigrahy et al., 2020).

Any bipartite matching problem instance can be solved very efficiently using a range of well-known algorithms, including combinatorial optimization algorithms, such as Hungarian algorithm and Jonker–Volgenant algorithm (Jonker and & Volgenant, 1987b), or newly developed machine-learning based algorithms (Georgiev and & Liò, 2020). However, in the context of service and resource planning, one is often interested in estimating the average matching cost across various problem realizations to evaluate the service efficiency under resource investments. For example, in designing mobility services, the average matching distance, or deadheading distance, is a key indicator that captures the unproductive cost spent by both service vehicles (i.e., running without passengers) and customers (i.e., waiting for pickup). Analytical formulas that reveal the relationship between the average matching distance and vehicle–customer distribution, such as those developed in Daganzo (1978b) and Yang et al. (2010), are often preferred by service operators, because these formulas can not only provide valuable managerial insights, but also be directly incorporated into mathematical models to optimize service offerings. Similar formulas have been used to design system-wide operational standards (e.g., demand pooling time interval) for customer–vehicle matching (Shen et al., 2024), evaluate the effectiveness of newly proposed customer matching strategies (Ouyang and & Yang, 2023a; Shen and & Ouyang, 2023; Stiglic et al., 2015), or analyze the impacts of pricing and market competition on social welfare (Wang et al., 2016; Zhou et al., 2022a).

The need to estimate the expected bipartite matching distance for planning decisions has led to the exploration of a stochastic version of the problem, known as the random bipartite matching problem (RBMP) in the literature. Earlier studies in the field of statistical physics were among the first to explore such a problem. Mézard and Parisi (1985) used a “replica method” to derive asymptotic formulas for the average optimal cost of a matching problem where the numbers of points in both subsets are nearly equal (i.e., balanced), and the edge weights identically and independently follow a uniform distribution. Building upon this work, Caracciolo et al. (2014) developed asymptotic approximations for the average Euclidean matching distance for balanced RBMPs in spaces with a dimension higher than two. However, these asymptotic approximations were derived under the strong assumption that the number of bipartite vertices approaches infinity and, hence, could only serve as bounds rather than exact estimates when the number of vertices is small. More importantly, their proposed formulas require curve fitting that estimates coefficients from simulated data. Daganzo and Smilowitz (2004) studied a related problem, which they called transportation linear programming (TLP), and they proposed approximated formula for estimating the average item distance among points with normally distributed demands and supplies. Through probabilistic and dimensional analysis, they introduced a bound to estimate the solution in two and higher dimensions. Very recently, Shen et al. (2024) proposed a set of closed-form formulas for arbitrary numbers of points in both subsets, an arbitrary number of spatial dimensions, and arbitrary Lebesgue distance metrics. Their model provides very accurate estimates, without curve fitting, in spaces with higher than two dimensions. However, their approach ignores the boundary of the space as well as the resulted correlation among the matched pairs, which is reasonable for higher dimension spaces but causes notable errors in one-dimensional space, especially when the two subsets are (nearly) of equal size.

For one-dimensional problems, Caracciolo et al. (2017) proposed an asymptotic formula for estimating the square of the average matching distance. However, similar to other asymptotic approximations, their formula is applicable only in a limited number of scenarios (e.g., when the number of bipartite vertices is balanced and approaches infinity). Also, their formula also require curve fitting based on simulated data. Meanwhile, Daganzo and Smilowitz (2004) also proposed an exact formula for the average minimal item distance among points with normally distributed demand and supply in a one-dimensional space. Their results cannot be applied directly to RBMP, because they assume that (i) the supply and demand points are balanced and that (ii) the supply or demand value associated with each randomly distributed point follows a normal distribution, while in RBMP, these values are either positive one or negative one. Nevertheless, their analysis provides very strong insights. One of their key findings is that the total minimal item distance for all points is equal to the size of the area enclosed by the cumulative supply curve and the x-axis. This essential property also holds for our one-dimensional RBMP.

To the best of our knowledge, no existing formula can provide accurate estimates for RBMPs (with arbitrary numbers of bipartite vertices) in a one-dimensional space or in a discrete network. This report aims to fill this gap by introducing a set of closed-form formulas (without curve fitting) that can provide sufficiently accurate estimates. This is done in two steps. First, we estimate the expected optimal matching distance for one-dimensional RBMPs. Our proposed method relates the matching distance in a balanced RBMP to the enclosed area size between the path of a random walk and the x-axis and derives a closed-form formula for the optimal matching distance for balanced RBMPs. For the more challenging unbalanced RBMPs, we develop a closed-form approximate formula by analyzing the properties of an unbalanced random walk and the optimal way to remove a subset of excessive points. Additionally, we propose a feasible point removal and swapping process to develop a set of recursive formulas that are more accurate. Next, we study the scaling property of the expected optimal matching distance in arbitrary-length lines. The insights are used as building blocks to derive formulas for RBMPs on a discrete regular network, when all points are generated from spatial Poisson processes along the edges. The expected optimal matching distance is derived as the expectation across two probabilistic matching scenarios that a point may encounter: (i) the point is locally matched with a point on the same edge or (ii) it is globally matched with a point on another edge. To verify the accuracy of the proposed formulas, we conducted a set of Monte-Carlo simulations for a variety of matching problem settings, for both one-dimensional and network problems. The results indicate that our proposed formulas have very high accuracy in all experimented problem settings. The proposed distance estimates, in simple closed forms, could be used directly in mathematical programs for strategic performance evaluation and optimization.

The remainder of this chapter is organized as follows. First, models and formulas for one-dimensional RBMP (as a building block) are presented in both balanced and unbalanced settings. Second, formulas for arbitrary-length lines and then regular discrete networks are presented. Then, numerical experiments are presented to validate the proposed formulas. Finally, concluding remarks and suggestions are provided for the future research directions.

1D RBMP DISTANCE ESTIMATORS

Problem Definition

We begin by defining the one-dimensional RBMP. Two sets of points, with given respective cardinalities $n \in Z^+$ and $m \in Z^+$, are independently and uniformly distributed on a unit length line within [0,1]. Without loss of generality, we assume $n \geq m$. For each realization of the points' locations, denote V and U as the two point sets, where $|V| = n$ and $|U| = m$. A bipartite graph can be constructed, whose set of edges E connect every pair of points in the two sets; i.e., $E = \{(u,v) : \forall u \in U, v \in V\}$. The weight on edge $(u,v) \in E$ is the absolute difference between the two points' coordinates x_u and x_v, i.e., $\|x_u - x_v\|$. Because $n \geq m$, every point $u \in U$ can be matched with exactly one point $v \in V$. We let $y_{uv} = 1$ if u is matched to v, or 0 otherwise. The objective is to find a set of matches $\{y_{uv} : \forall u \in U, v \in V\}$ that minimize the total matching distance, as follows:

$$\min \sum_{u \in U, v \in V} y_{uv} \|x_u - x_v\|; \tag{2.1}$$

$$\text{s.t.} \sum_{v \in V} y_{uv} = 1, \forall\, u \in U; \quad \sum_{u \in U} y_{uv} \leq 1, \forall\, v \in V; \quad y_{uv} \in \{0,1\}, \forall\, u \in U, v \in V. \tag{2.2}$$

Figure 2. Equation. Equation (2.1) and (2.2).

The average optimal matching distance across the realized points in U is a random variable that depends on the random realization of U and V. Its distribution is governed by parameters m and n, so we denote it $X_{m,n}$. We are looking for a closed-form formula for the expectation of the average optimal matching distance, E[$X_{m,n}$]. In order to do that, we first show that $X_{m,n}$ can be estimated based on the enclosed area between the path of a related random walk and the x-axis, and then we take the expectation of this enclosed area. For a simple balanced problem (i.e., when $n = m$), we can directly derive a closed-form formula. For an unbalanced problem (when $n > m$), we first show optimality properties for the matching and then build upon that to derive both a closed-form and a recursive formula.

Random Walk Approximation

For each realized instance of one-dimensional RBMP, let $I = U \cup V$. Sort all points in I by their x-coordinates between 0 and 1, and index them sequentially by i. For point $i \in I$, denote $x_i \in [0,1]$ as its x-coordinate and z_i as the value of its supply; i.e., $z_i = 1$ if $i \in V$ (indicating a supply point), or $z_i = -1$ if $i \in U$ (indicating a demand point). A cumulative “net” supply curve can then be constructed for any coordinate x and any subset of points $I' \subseteq I$; i.e., $S(x;I') = \sum_{\{i \in I', x_i \leq x\}} z_i$. It is a piecewise step function. There are three special cases: $S(x;I)$ represents the net supply curve constructed by the full set of points in I and $S(0;I')$ and $S(1;I')$ represent the net supply values at both ends of the curve, $x = 0$ and $x = 1$, respectively.

Every curve $S(x;I)$ can be related to the realized path of a specific type of one-dimensional random walk with $m + n$ steps, starting from $S(0;I) = 0$. Among these $m + n$ steps, exactly n steps each increase the net supply by 1 and m each decrease the net supply by 1; as such $S(1;I) = n - m$. The locations of the points in I correspond to the positions of these steps. Denote the distance (step size) from point (step) $i \in I$ to its next point (step) as $l_i, \forall i \in I \setminus \{|I|\}$. The step size varies because the points in I are uniformly distributed along the x-axis.

Denote $A(x;I') = \sum_{\{\forall i \in I \setminus \{|I|\}, x_i \leq x\}} l_i \cdot |S(x_i;I')|$ as the total absolute area between curve $S(x;I')$ and the x-axis from 0 to x. For model simplicity, we assume the step sizes $l_i, \forall i \in I \setminus \{|I|\}$ are i.i.d., with mean l. Next we show how E[$X_{m,n}$] can be derived out of such an area, for both balanced and unbalanced matchings.

Balanced Case (*m* = *n*)

We begin with the special case where $m = n$. Now set I contains $2n$ points. Let $Y_n = A(1;I)$ be the random variable that equals the total absolute area between curve $S(x;I)$ and the x-axis from 0 to 1. Daganzo and Smilowitz (2004) proved that $A(1;I)$ must equal the minimum total shipping distance of a one-dimensional TLP with an equal number of supply and demand points, and, thus, it must also equal the minimum total matching distance of the corresponding one-dimensional RBMP instance. Thus, the expected optimal matching distance per point can be estimated by the following:

$$\mathbb{E}[X_{n,n}] = \frac{1}{n} \cdot \mathbb{E}[Y_n]. \tag{2.3}$$

Figure 3. Equation. Equation (2.3).

To further estimate E[Y_n], we first consider a simpler type of related random walk with a fixed-step size of unit length. Let $B(n)$ denote the expected area between the path of such a random walk and the x-axis. Harel (1993) has provided a formula for $B(n)$, as follows:

$$B(n) = \frac{n 2^{2n-1}}{\binom{2n}{n}} \xrightarrow{n \gg 1} \frac{n\sqrt{\pi n}}{2}. \tag{2.4}$$

Figure 4. Equation. Equation (2.4).

The last step of approximation, when $n \gg 1$, comes from Stirling's approximation. The basic intuition behind this formula is as follows. In a random walk with a fixed total number of steps (e.g., 2n), there are a finite number of possible combinations of upward (e.g., $n + k$) steps and downward steps (e.g., $n-k$), when k varies from 0 to 2n. Next, for each k value, the probability and expected area of a random walk can be determined: the probability is derived using backwards induction starting from some simple cases (e.g., k = 0); the expected area, conditional on k, is computed by the absolute difference between the numbers of upward and downward steps, multiplied by the step size. (For example, for a random walk with $n + k$ upward steps and $n - k$ downward steps, its expected area between the curve and the x-axis is simply 2k.) Then, Equation (2.4) can be obtained by taking the unconditional expectation of these area sizes across all possible values of k.

Then, consider the type of random walk with varying step sizes. Under the i.i.d. assumption for l_i, we can multiply $B(n)$ by the mean step size l to obtain an approximate estimation for E[Y_n], as follows:

$$\mathbb{E}[Y_n] \approx l \cdot B(n). \tag{2.5}$$

Figure 5. Equation. Equation (2.5).

Finally, according to Equations (2.3)–(2.5), and note $l = \frac{1}{2n}$ in this case, we obtain the following closed-form formula for E[$X_{n,n}$]:

$$\mathbb{E}[X_{n,n}] = \frac{l \cdot B(n)}{n} = \frac{1}{2n} \cdot \frac{2^{2n-1}}{\binom{2n}{n}} \xrightarrow{n \gg 1} \frac{1}{4}\sqrt{\frac{\pi}{n}}. \tag{2.6}$$

Figure 6. Equation. Equation (2.6).

Unbalanced Case ($n > m$)

Next, we consider the unbalanced case where $n > m$, for which $n - m$ supply points from V will remain unmatched for each realization. Let $V' \subset V$ denote the set of these $n - m$ unmatched points. If we remove all points in V' from V, the problem will reduce to a balanced one with an equal number (i.e., m) of demand and supply points. As a result, the values on the net supply curve after point removal, $S(x; I \setminus V')$, at x = 0 and x = 1 should both equal zero; i.e.,

$$S(0; I \setminus V') = S(1; I \setminus V') = 0. \qquad (2.7)$$

Figure 7. Equation. Equation (2.7).

Figure 9 shows an example of how such a point removal process affects the net supply curve along the entire x-axis. In this figure, the points in *V* and *U* are represented by the red dots and blue triangles, respectively. The original and post-removal curves, *S*(*x*;*I*) and *S*(*x*;*I* \ *V* $'$), are represented by the red dash-dot line and blue dashed line, respectively. The points in *V* $'$ and the net supply values at their corresponding coordinates on the original curve, $S(x_{V'};I),\forall v' \in V'$, are marked by the black cross markers. Note that every time a point $v' \in V'$ is removed, the net supply values within $[x_{v'},1]$ will decrease by one. As a result, the cumulative reduction of net supply at position *x* should be determined by the total number of removed points within [0,*x*]. Sort the points in *V* $'$ by their x-coordinates, from left to right, as $\{v'_1, \cdots, v'_{n-m}\}$. If we further denote v'_0 and v'_{n-m+1} as two virtual points at the boundaries (i.e. $x_{v'_0}$= 0 and$x_{v'_{n-m+1}}$ = 1), then the net supply values on the two curves within range $[x_{v'_k}, x_{v'_{k+1}})$ must satisfy the following relationship:

$$S(x; I \setminus V') = S(x; I) - k, \forall k \in \{0, \cdots, n-m\}, x \in [x_{v'_k}, x_{v'_{k+1}}). \qquad (2.8)$$

Figure 8. Equation. Equation (2.8).

This relationship is illustrated in Figure 9 by the black arrows.

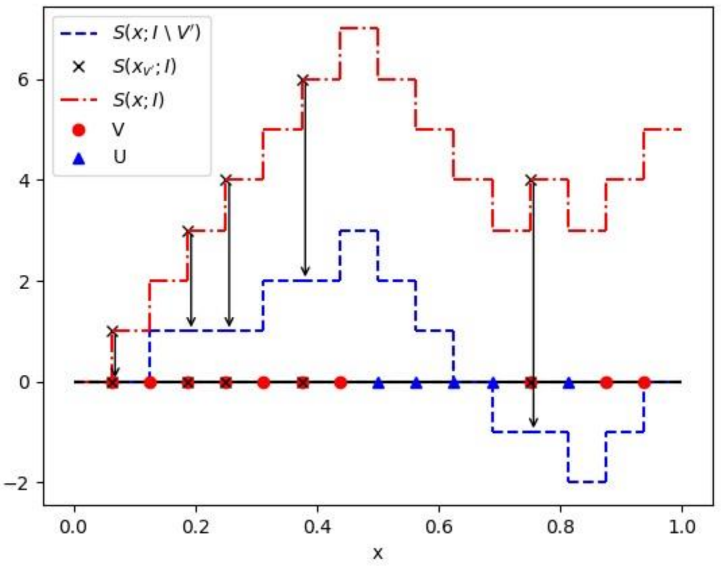


Figure 9. Graph. Point removal process in an unbalanced problem.

Among all possible combinations of points in set *V* $'$, we denote *V* * as the optimal set of removed points that minimizes the area enclosed by the post-removal curve and the x-axis:

$$V^* = \underset{\forall V' \subset V,\ |V'|=n-m}{\operatorname{argmin}} \quad A(1; I \setminus V').$$

Figure 10. Equation. Equation of V*.

The optimal post-removal area $A(1;I \setminus V^*)$, enclosed by the optimal post-removal curve $S(1;I \setminus V^*)$ and the x-axis, must equal the minimum total matching distance of the original unbalanced RBMP instance. Set random variable $Z_{m,n} = A(1;I \setminus V^*)$, which depends on the random realization of $I = U \cup V$ (where $|V| = n$ and $|U| = m$), and then, similar to how we handle the balanced case:

$$\mathbb{E}[X_{m,n}] = \frac{1}{m} \cdot \mathbb{E}[Z_{m,n}]. \qquad (2.9)$$

Figure 11. Equation. Equation (2.9).

In the following subsections, we will (i) show the optimal post-removal curve must include a series of balanced random walk segments; (ii) derive an approximate closed-form formula for $E[X_{m,n}]$ by estimating the area of each balanced segment; (iii) provide an alternative estimation for $E[X_{m,n}]$ with a recursive formula based on a feasible point selection process; and (iv) refine both estimations with a correction term.

Property of the Optimal Removal

We first show a necessary condition for the removed points to be optimal: the k-th removed point must have a net supply value of k on the original curve $S(x;I)$. This is stated in the following proposition.

Proposition 1. $S(x_{v_{k^*}};I) = k, \forall k \in \{1,...,n-m\}$.

Proof. To show the proposition holds, it is sufficient to show the following claim is true: For any point $v_k' \in V'$, if $S(x_{v_k'};I) < k$ or $S(x_{v_k'};I) > k$, we can always swap a point in V' with another point in $V \setminus V'$ to reduce $A(x;I \setminus V')$; hence, V' cannot be optimal.

We begin with the case when $S(x_{v_k'};I) < k$. According to Equation (2.8), v_k' must have a negative net supply value on the post-removal curve; i.e., $S(x_{v_k'};I \setminus V') < 0$. Figure 13(a) shows an example point v_k' (indicated by the cross marker) in such a condition. A portion of the post-removal curve, including the removal of this single point, is represented by the red dash-dot line. Now we may check the points in I to the righthand side of v_k' along the x-axis, until we encounter another supply point $v \in V$. Such a supply point v is guaranteed to be within $(x_{v_k'},1]$, otherwise $S(1;I \setminus V') \leq S(x_{v_k'};I \setminus V') < 0$, which violates Equation (2.7).

Two cases may arise here for v. The first case is when $v /\in V'$, as shown in Figure 13(a). Because v is the first supply point to the right of v_k', the points within $(x_{v_k'},x_v)$, if any, must all be demand points, as shown by the blue triangles in Figure 13(a). As all demand points have negative supply values, the net supply values on the post-removal curve within $(x_{v_k'},x_v)$ must be no larger than $S(x_{v_k'};I \setminus V')$; i.e., for $x \in$

($x_{vk'}$,x_v), $S(x;I \setminus V')$ ≤ $S(x_{vk'};I \setminus V')$ < 0. Now swap $v_k{}'$ out of V', and swap v in. Note here such a point swap would only affect the post-removal curve within [$x_{vk'}$,x_v), and after the swap, the original post-removal curve, $S(x;I \setminus V')$, will increase by one unit within [$x_{vk'}$,x_v), while all other parts remain the same. The area size under the curve is strictly reduced by the point swap, by an amount of $A(x; I \setminus V)$, as shown by the gray area in Figure 13(a). The reduced area size is shown below:

$$\sum_{\{\forall i\in I, x_{v'_k}\le x_i<x_v\}} l_i \cdot |S(x_i; I\setminus V'\setminus\{v\}\cup\{v'_k\}) - S(x_i; I\setminus V')| = \sum_{\{\forall i\in I, x_{v'_k}\le x_i<x_v\}} l_i > 0,$$

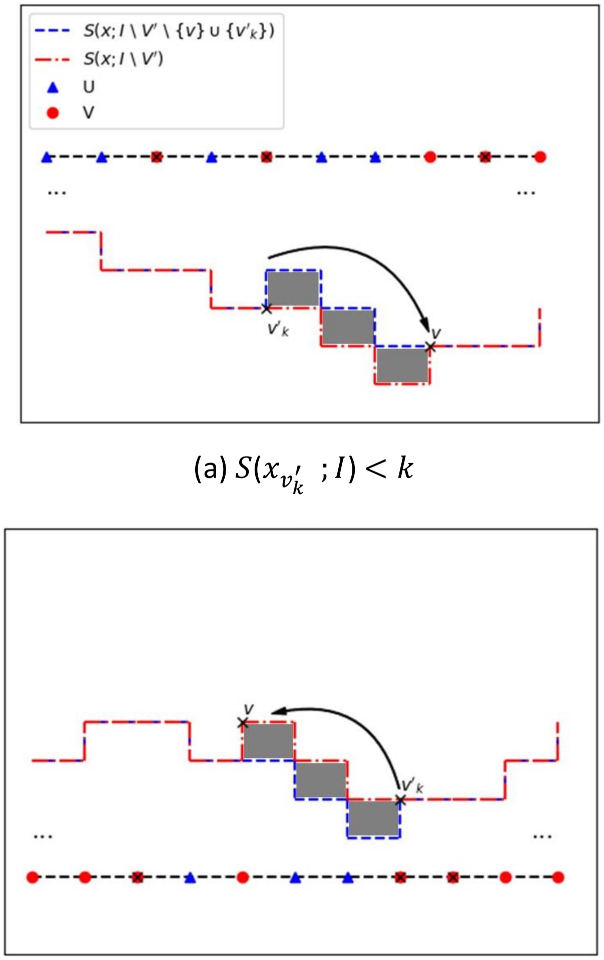
Figure 12. Equation. Equation of the reduced area size by a point removal.

The last inequality holds because v must exist. Therefore, the claim is true for $S(x_{v'_k}; I) < k$ (i.e., $S(x_{v'_k};I \setminus V') < 0$) and $v \notin V'$.

The other case occurs when $v \in V'$. Because v is the first supply point encountered in V' to the right of v'_k, v must be v'_{k+1},. Again, all points within $(x_{v'_k}, x_v)$, if any, must be demand points. In addition, because $v_k{}'_{+1}$ itself is a removed point, $S(x_{v'_{k+1}}; I \setminus V') \le S(x_{v'_k}; I \setminus V') < 0$. Then, we may simply change our focus from v'_k to v'_{k+1}, and then scanning the points on the right side of v'_{k+1}, and continue until we find a point $v' \in V'$ with $S(x_{v'_k}; I \setminus V') < 0$, and its next supply point $v \notin V'$. The proof for the previous case would apply for swapping v' and v. Hence, the claim is also true for $S(x_{v'_k}; I) < k$ and $v \in V'$.

When $S(x_{v'_k}; I) > k$, the proof is symmetric. We look for points to swap that can lead v'_k to area reduction for $A(x; I \setminus V)$ to the lefthand side along the x-axis, instead of the right. According to Equation (2.8), v'_k now must have a positive net supply value on the post-removal curve; i.e., $S(x_{v'_k};I \setminus V') > 0$. A supply point to the left must exist within $[0, x_{v'_k})$, or otherwise $S(0;I \setminus V') \ge S(x_{v'_k};I \setminus V') > 0$, which violates Equation (2.7). There are two similar cases here depending on whether v is or is not in V'. The logic of the proof is exactly symmetrical. The only difference is that, after swapping the two points, the original curve will “decrease” by one unit, instead of increase, within the interval. The area reduction is still strictly greater than zero, as shown in Figure 13(b).

We have shown that the claim is true in all possible conditions. This indicates that if any removed point v' in an arbitrary set V' does not satisfy $S(x_{v'_k}; I) = k$, then V' cannot be optimal. Therefore, for any removed point v^*_k in an optimal set V^*, $S(x_{v^*_k}; I) = k$ must hold necessarily. This completes the proof.



(a) $S(x_{v'_k} \ ; I) < k$

(b) $S(x_{v'_k} \ ; I) > k$

Figure 13. Graph. Illustration of the point-swap process.

Next, we present a useful property of the post-removal curve satisfying the optimality condition: $S(x_{v'_k}; I) = k, \forall k \in \{1,...,n-m\}$. According to Equation (2.8), the net supply values on the post-removal curve at the locations of all removed points must be zero; i.e.: $S(x_{v'_k}; I \setminus V') = S(x_{v'_k}; I) - k = 0$. As such, the following proposition must hold.

Proposition 2. *For any given V', if $S(x_{v'_k}; I) = k, \forall k \in \{0,1,...,n-m\}$, then each segment of the post-removal curve, $S(x; I \setminus V')$ within $(x_{v'_k}, x_{v'_{k+1}})$, can be regarded as the realized path of a corresponding balanced random walk.*

Approximate Closed-Form Formula

Propositions 1 and 2 show that the optimal post-removal curve $S(x; I \setminus V^*)$ contains $n - m + 1$ segments with end points $\{x_{v^*_k}\}_{k=0,1,\cdots}$, and each segment $(x_{v^*_k}, x_{v^*_{k+1}})$ can be regarded as a realized path of a balanced random walk. Accordingly, we can decompose the optimal post-removal area $A(1; I \setminus V^*)$ into $n - m + 1$ segments as follows:

$$A(1; I \setminus V^*) = \sum_{k=0}^{n-m} \left[A(x_{v^*_{k+1}}; I \setminus V^*) - A(x_{v^*_k}; I \setminus V^*) \right], \tag{2.10}$$

Figure 14. Equation. Equation of post-removal area.

where each term $A(x_{v^*_{k+1}}; I \setminus V^*) - A(x_{v^*_k}; I \setminus V^*)$ represents the area of a balanced random walk segment within range ($x_{v^*_k}$, $x_{v^*_{k+1}}$).

For model convenience, we again ignore the impact of boundaries and assume that every point $v \in V$ is probabilistically identical and independent to be selected in V^*. Based on this assumption, we can directly apply Equation (2.5) to estimate the expected area of each balanced random walk segment, which shall be dependent on only the number of (balanced) points and the mean step size within each segment. Denoting m_k as the number of demand (or supply points) in U in the k-th segment, E[$Z_{m,n}$] can then be approximately estimated as the following sum:

$$\mathbb{E}[Z_{m,n}] \approx \sum_{k=0}^{n-m} \mathbb{E}\left[l \cdot B(m_k)\right]. \tag{2.11}$$

Figure 15. Equation. Equation (2.11).

It is approximate because we simply treat the mean step size in each balanced random walk segment as the same as that of the original unbalanced random walk (i.e., $l = \frac{1}{n+m}$). This could lead to an overestimation, and a correction term could be added. With our i.i.d. assumption for V^* and disregard of the boundary effects, we must have m_k, $\forall k \in \{0,...,n - m\}$ to be i.i.d. This may result in an underestimation, because points at specific positions may have a higher probability of being selected in V^* than others due to the presence of boundaries.

Then, we may only focus on the expected area of the first segment (i.e., when k = 0). Let Pr{$m_0 = m'$} denote the probability that the first segment contains exactly $m' \in \{1, \cdots, m\}$ demand points, noticing that the total number of demand points across all $n - m + 1$ segments must equal m; i.e., $\sum_{k=0}^{n-m} m_k = m$. To derive Pr{$m_0 = m'$}, one may relate it to the well-known "stars and bars" combinatorial problem. When we use $n-m$ "bars" (points in V^*) to partition m stars (all matched pairs of demand and supply points), there are $\binom{n}{n-m}$ possible combinations for such a partition. Once the first segment has been set to contain m' stars, there remain $n-m'-1$ positions for the remaining $n-m-1$ bars to be placed, resulting in $\binom{n-m'-1}{n-m-1}$ possible combinations. That is,

$$\Pr\{m_0 = m'\} = \frac{\binom{n-m'-1}{n-m-1}}{\binom{n}{n-m}}. \tag{2.12}$$

Figure 16. Equation. Equation (2.12).

Hence, following Equations (2.6), (2.9), and (2.12), and note $l = \frac{1}{n+m}$, in this case, we now have a closed-form formula for E[$X_{m,n}$], as follows:

$$\mathbb{E}[X_{m,n}] \approx (n-m+1) \cdot \mathbb{E}\left[l \cdot B(m_0)\right] \approx (n-m+1) \cdot l \cdot \sum_{m'=0}^{m} \Pr\{m_0 = m'\} \cdot B(m')$$
$$\approx \frac{n-m+1}{m(m+n)} \cdot \sum_{m'=0}^{m} \frac{\binom{n-m'-1}{n-m-1}}{\binom{n}{n-m}} \cdot \frac{m' 2^{2m'-1}}{\binom{2m'}{m'}}. \qquad (2.13)$$

Figure 17. Equation. Equation (2.13).

Again, this formula is approximate due to our simplifying assumptions. The next section will present an alternative method to yield a more accurate estimation via a specific point removal and swapping process.

Recursive Formulas

In theory, the optimal point removal for each realization shall be solved as a dynamic program (and possibly via the Bellman equation); however, such an approach is not suitable for closed-form formulas. Therefore, this section builds upon Propositions 1 and 2 to propose a simpler point removal and swapping process that can yield a near-optimum point-removal solution for each realization. Then the area size estimation across realizations, based on this process, can be derived into a set of recursive formulas. The result provides a tight upper bound for E[$X_{m,n}$].

First, we propose a point removal process that is developed based on the necessary optimality conditions in Proposition 1, such that it provides a reasonably good (e.g., locally optimal) balanced random walk. We initialize $\hat{V}$ = ∅ and *k* = 1. Starting from *x* = 0, scan the supply points in *V* from left to right along the x-axis and check if a point satisfies the following conditions: (i) it has a net cumulative supply value of *k*; (ii) the nearest neighbor (in *I*) on the left, if existing, has a net supply value of *k*−1; and (iii) the net supply values of all points (in *I*) on the righthand side of this point is greater than or equal to *k*. If conditions (i)–(iii) are all satisfied by a point, denoted $\hat{v}_k$, then add it to the current set $\hat{V}$, increase *k* by 1, and repeat the above procedure until *k* = *n* − *m*. The net supply values of the points selected in this procedure are guaranteed to form an increasing sequence; i.e.:

$$S(x_{\hat{v}_k}; I) = k, \quad \forall\, k \in \{1, \ldots, n-m\}.$$

Figure 18. Equation. Net supply values in an increasing sequence.

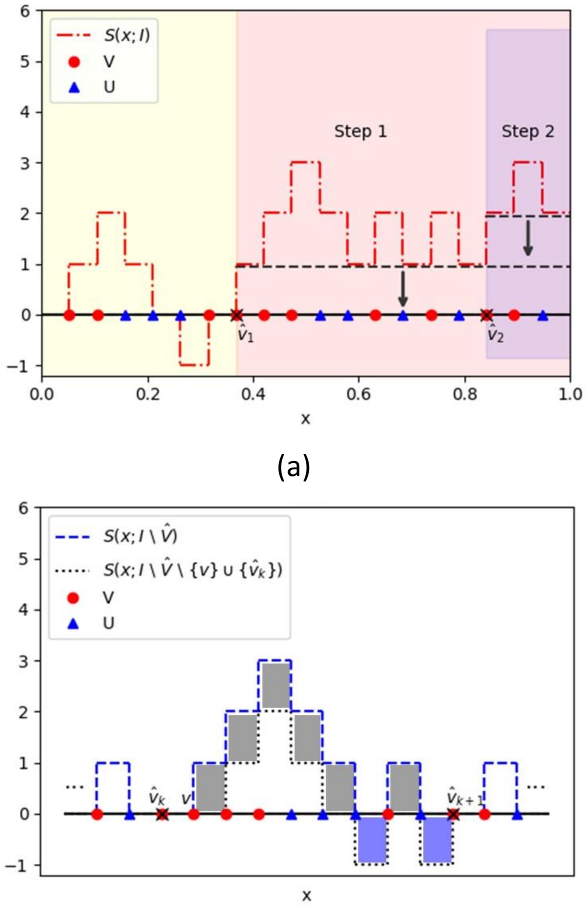
This is because, at step *k* of the above process, the curve *S*(x; *I*) must have a net supply value of *k* − 1 at the previously selected point $\hat{v}_{k-1}$ and a value of at least *n* − *m* at the final step at *x* = 1. Hence, from the intermediate value theorem, point $\hat{v}_k$ can always be found in ($x_{\hat{v}_{k-1}}$,1]. Note that the removal of $\hat{v}_{k-1}$at step *k* only affects the curve on its righthand side, with the segments on its lefthand side being balanced random walks. From the construction of the process, all points on the final post-removal curve *S*(x; *I* \ $\hat{V}$) surely have a non-negative net supply value; i.e.,

$$S(x; I \setminus \hat{V}) \geq 0, \quad \forall x \in (x_{\hat{v}_k}, 1],\ k \in \{1, \cdots, n-m\}.$$

Figure 19. Equation. Non-negative net supply condition.

Figure 20 (a) shows a simple example of the point removal process, with $n - m = 2$. The original curve, $S(x; I)$, is represented by the red dash-dot line, and two selected removal points $\hat{v}_1$ and $\hat{v}_2$ are represented by the black cross markers. At step 1 and 2, the removal of $\hat{v}_1$ and $\hat{v}_2$ decrease the net supply values of points in the red and purple shaded regions, respectively.

We again use the term "segment of the curve" but now refer to the post-removal curve $S(x; I \setminus \hat{V})$ within $(x_{\hat{v}_k}, x_{\hat{v}_{k+1}})$ as the "k-th segment" (which must be balanced). From the perspective of each point $\hat{v}_k$ the post-removal area in $(x_{\hat{v}_k}$,1] includes those within the kth segment and $(x_{\hat{v}_{k+1}}$,1], both of which depend on the length of the k-th segment. Let $\Pr\{\widehat{m}_k \mid a\}$ denote the probability for the k-th segment to contain $\widehat{m}_k$ demand (or supply) points, when there are exactly a demand points in $(x_{\hat{v}_k}$,1]. In the step to select v_{k+1}, there are $n - m - k$ supply points to be removed; thus, there are $a + n - m - k$ supply points in $(x_{\hat{v}_k}$,1].



(a)

(b)

Figure 20. Graph. Illustration of the point removal and swap procedure.

That is, the original curve *S*(x; *l*) starts from value *k* at $\hat{v}_k$ and returns to *k* after exactly $2\widehat{m}_k$ steps, which occurs with probability

$$\binom{a}{\hat{m}_k}\binom{a+n-m-k}{\hat{m}_k}/\binom{2a+n-m-k}{2\hat{m}_k}$$

Figure 21. Equation. Probability of returning to same position.

but never returns to value *k* afterwards, with probability from the well-known Ballot's theorem (Addario-Berry and & Reed, 2008):

$$\frac{n-m-k}{2a+n-m-k-2\hat{m}_k}$$

Figure 22. Equation. Probability of never hitting zeros.

As such, we have

$$\Pr\{\hat{m}_k \mid a\} = \frac{\binom{a}{\hat{m}_k}\binom{a+n-m-k}{\hat{m}_k}}{\binom{2a+n-m-k}{2\hat{m}_k}} \cdot \frac{n-m-k}{2a+n-m-k-2\hat{m}_k}. \qquad (2.14)$$

Figure 23. Equation. Equation (2.14).

Let $\hat{Z}_{k,a}$ represent the area enclosed by the x-axis and the post-removal curve *S*(x; *l* \ $\hat{V}$) within ($x_{\hat{v}_k}$,1], which contains exactly *a* demand points. From Equation (2.5), the expected areas size of the balanced random walk inside the *k*-th segment ($x_{\hat{v}_k}$, $x_{\hat{v}_{k+1}}$) is simply:

$$l \cdot B(\hat{m}_k).$$

Figure 24. Equation. Expected area size of k-th segment.

Hence, conditional on $\widehat{m}_k$, we have

$$\mathbb{E}[\hat{Z}_{k,a}] = \sum_{m'=0}^{a} \Pr\{\hat{m}_k = m' \mid a\} \cdot \left[l \cdot B(m') + \mathbb{E}[\hat{Z}_{k+1,a-m'}]\right], \\ \forall\, k \in \{0, \ldots, n-m-1\}, \quad 0 \le a \le m, \qquad (2.15)$$

Figure 25. Equation. Equation (2.15).

and when k = n – m,

$$\mathbb{E}[\hat{Z}_{n-m,a}] = l \cdot B(a), \quad \forall 0 \le a \le m.$$

Figure 26. Equation. Expected area of $\widehat{Z}_{n-m,a}$.

When *k* = 0, E[$\hat{Z}_{0,a}$] represents the expected area of the entire post-removal curve.

Next, we propose a refinement process based on local point swapping. It is intended to further reduce the post-removal area for a more accurate upper bound estimate. For all *k* ∈ {1,...,*n* − *m* − 1}, if there exists a point *v* ∈ *V* that satisfies (i) $x_{\hat{v}_k} < x_v < x_{\hat{v}_k}$ and (ii) *v* is the nearest neighbor of $\hat{v}_k$ on the right, then we swap $\hat{v}_{k+1}$ out of set $\hat{V}$ and swap point *v* in. We propose to perform exactly one such point swap to each segment.

An example of point swap is illustrated in Figure 20(b). The post-removal curve *S*(x; *I* \ $\hat{V}$) is represented by the blue dash curves. After a swap between points $\hat{v}_{k+1}$ and *v*, the *k*-th curve segment will be shifted down by one unit (while all other segments remain the same), as indicated by the black dotted curves. Such a point swap can result in both reductions (if net supply *S*(x; *I* \ $\hat{V}$) > 0 before the swap, represented by the grey rectangles) and additions (if net supply *S*(x; *I* \$\hat{V}$) = 0 before the swap, represented by the blue rectangles) to the enclosed area size. The following proposition says that the expected total area size will be reduced for each swap. This indicates that the proposed point-swapping process will yield a smaller (or at least equal) expected post-removal area size.

Suppose $\hat{m}_{k,0} \leq \hat{m}_k$ is the random number of points with zero net supplies within the *k*-th segment. According to the point-swapping process, the area size reduction can be computed as the difference between the number of positive net-supply points, 2$\hat{m}_k$ − $\hat{m}_{k,0}$, and that of the zero net-supply points $\hat{m}_{k,0}$, multiplied by the expected step size *l*. If we take the expectation of the area change with respect to $\hat{m}_{k,0}$, conditional on $\hat{m}_k$, we have

$$l \cdot (2\hat{m}_k - 2\mathbb{E}[\hat{m}_{k,0} \mid \hat{m}_k])\,, \quad \forall\, k \in \{1, \ldots, n-m-1\}. \tag{2.16}$$

Figure 27. Equation. Equation (2.16)

Note that to compute E[$\hat{m}_{k,0}$| $\hat{m}_k$], it is equivalent to compute the expected number of times a random walk with 2$\hat{m}_k$ steps returns to zero. This expectation equals zero when $\hat{m}_k$ = 0. Meanwhile, Harel (1993) derived the probability that a random walk with 2$\hat{m}_k$ steps returns to zero at the 2*j*-th step, ∀*j* ∈ {1,..., $\hat{m}_k$}:

$$\binom{2j-1}{j}\binom{2\hat{m}_k - 2j}{\hat{m}_k - j}\Big/\binom{2\hat{m}_k - 1}{\hat{m}_k}, \quad \forall\, k \in \{1, \ldots, n-m-1\}$$

Figure 28. Equation. Probability of a random walk returning to zero.

Summing these probabilities across all possible values of *j*, we have

$$\mathbb{E}[\hat{m}_{k,0} \mid \hat{m}_k] = \begin{cases} 0, & \forall\, \hat{m}_k = 0, \\ \sum_{j=1}^{\hat{m}_k} \binom{2j-1}{j}\binom{2\hat{m}_k-2j}{\hat{m}_k-j}/\binom{2\hat{m}_k-1}{\hat{m}_k}, & \forall\, \hat{m}_k \neq 0, \end{cases} \tag{2.17}$$

Figure 29. Equation. Equation (2.17)

Now, adding Equation (2.16) as a correction term into Equation (2.15), and note $l = \frac{1}{n+m}$, we have

$$\begin{aligned}
\mathbb{E}[\hat{Z}_{0,m}] &= \sum_{m'=0}^{m} \Pr\{\hat{m}_0 = m' \mid m\} \cdot \left[l \cdot B(m') + \mathbb{E}[\hat{Z}_{1,m-m'}] \right], \\
\mathbb{E}[\hat{Z}_{k,a}] &= \sum_{m'=0}^{a} \Pr\{\hat{m}_k = m' \mid a\} \cdot \left[l \cdot B(m') - l \cdot (2m' - 2\mathbb{E}[\hat{m}_{k,0} \mid m']) + \mathbb{E}[\hat{Z}_{k+1,a-m'}] \right], \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall\, k \in \{1, \ldots, n-m-1\}, \\
\mathbb{E}[\hat{Z}_{n-m,a}] &= l \cdot B(a).
\end{aligned} \tag{2.18}$$

Figure 30. Equation. Equation (2.18)

Together with Equations (2.14) and (2.17), all the above expected area sizes can be solved recursively. The expected matching distance E[$X_{m,n}$] is related to E[$\hat{Z}_{0,m}$], as follows:

$$\mathbb{E}[X_{m,n}] \approx \frac{1}{m} \cdot \mathbb{E}[\hat{Z}_{0,m}]. \tag{2.19}$$

Figure 31. Equation. Equation (2.19).

Step Length Correction

Finally, we propose a correction term to the formulas in Equations (2.13) and (2.19), for the unbalanced case, so as to address the fact that the expected step size of the balanced random walk segments, after point removals, is not the same as that of the original unbalanced random walk, $l = \frac{1}{n+m}$.

Under the current treatment, the expected step size of the balanced random walk segments, *l*, represents the minimum achievable expected matching distance between any two points. Consider the special case when *n* ≫ *m*. The estimated matching distance from both Equations (2.13) and (2.19) should converge to *l*. However, if this treatment is relaxed, because the step size l_i varies, the unmatched points are more likely to be farther away from the other points and have larger l_i values compared to matched ones. As a result, after removing the optimal unmatched points, the "true" expected step size of the balanced random walk segments, which contain only the successfully matched points, should be no larger than *l*. Therefore, this treatment may lead to an overestimation of the "true" optimal matching distance, and the resulting estimation gap is expected to be positively related to the number of unmatched points *n* − *m*.

It is non-trivial to estimate this gap explicitly. Therefore, we introduce an approximate correction term derived based on the case when *n* ≫ *m*. Recall Shen et al. (2024) proposed a set of formulas for estimating the matching distance of RBMPs in any dimension. For the one-dimensional case, they provide both a general formula and an asymptotic approximation, as follows:

$$\mathbb{E}[X_{m,n}] \approx \frac{1}{2m(n+1)} \sum_{i=1}^{m} \left[\sum_{k=1}^{i} k \left(\frac{i-1}{n} \right)^{k-1} \left(1 - \frac{i-1}{n} \right) + i \left(\frac{i-1}{n} \right)^{i} \right] \xrightarrow{n \gg m} \frac{1}{2n}. \quad (2.20)$$

Figure 32. Equation. Equation (2.20).

This formula is quite accurate when *n* ≫ *m*. Thus, when *n* ≫ *m*, the gap between the estimations given by Equations (2.13) or (2.19) and the one given by Equation (2.20) is simply $l - \frac{1}{2n} = \frac{n-m}{2n(m+n)}$. Note that when *m* = *n*, this gap is zero, which aligns with our expectations. We will use this as an approximate correction term and subtract it from the estimates provided by both Equations (2.13) and (2.19) across all *m* and *n* values. The final formulas for estimating E[$X_{m,n}$] after applying the correction are as follows:

$$\mathbb{E}[X_{m,n}] \approx \left[\frac{n-m+1}{m(m+n)} \cdot \sum_{m'=0}^{m} \frac{\binom{n-m'-1}{n-m-1}}{\binom{n}{n-m}} \cdot \frac{m' 2^{2m'-1}}{\binom{2m'}{m'}} \right] - \frac{n-m}{2n(m+n)}, \quad (2.21)$$

$$\mathbb{E}[X_{m,n}] \approx \left[\frac{1}{m} \cdot \mathbb{E}[\hat{Z}_{0,m}] \right] - \frac{n-m}{2n(m+n)}, \quad (2.22)$$

Figure 33. Equation. Equation (2.21) and (2.22).

where $\mathrm{E}[\hat{Z}_{0,m}]$ is given by Equation (2.18).

DISCRETE NETWORK RBMP ESTIMATORS

All the results so far are derived in a unit-length line segment with given numbers of points. In order to address the expected optimal matching distance when points are distributed on a discrete network, we next show two extensions: (i) when the line segment has an arbitrary length and (ii) when the line segment is an edge of a discrete network, such that some of the matches must be made across edges.

Arbitrary-Length Line

First, we see how the expected matching distance scales when the line has an arbitrary length of *L* du. To connect with the previous sections, we introduce point densities (per unit length) *μ* and *λ*, where *μ* ≤ *λ*, such that *m* = *μL* and *n* = *λL*. The average distance between any two adjacent points $l = \frac{1}{\lambda+\mu}$ du. The average optimal matching distance in this case, denoted as X_E for an "edge," should be governed by three parameters: *μ*, *λ*, and *L*.

We will consider a few possible cases. For the balanced case (*λ* = *μ*), we can simply substitute *n* = *λL* and $l = \frac{1}{2\lambda}$ into Equations (2.6), which gives an updated expected matching distance as:

$$\mathbb{E}[X_{\mathrm{E}}] = \frac{l \cdot B(\lambda L)}{\lambda L} = \frac{2^{2\lambda L-1}}{2\lambda\binom{2\lambda L}{\lambda L}} \xrightarrow{\lambda L \gg 1} \frac{1}{4}\sqrt{\frac{\pi L}{\lambda}}, \quad \text{if } \lambda = \mu. \tag{2.23}$$

Figure 34. Equation. Equation (2.23).

We observed from this formula that the expected matching distance scales with *L*. For the highly unbalanced cases ($\lambda \gg \mu$), we can scale the asymptotic approximation in Equation (2.20) by multiplying *L* and substitute $n = \lambda L$, which gives the following:

$$\mathbb{E}[X_{\mathrm{E}}] \approx \frac{L}{2\lambda L} = \frac{1}{2\lambda}, \quad \text{if } \lambda \gg \mu. \tag{2.24}$$

Figure 35. Equation. Equation (2.24).

In this case, the formula shows that the expected matching distance is independent of *L*. For other unbalanced cases ($\lambda \gtrapprox \mu$), we may substitute $n = \lambda L, m = \mu L, l = \frac{1}{\lambda+\mu}$, and the correction term $l - \frac{1}{2\lambda}$ into Equations (2.21)–(2.22), which leads to the following:

$$\mathbb{E}[X_{\mathrm{E}}] \approx \frac{\lambda L - \mu L + 1}{\lambda + \mu} \cdot \sum_{m'=0}^{\mu L} \Pr\{m_0 = m'\} \cdot B(m') - \frac{\lambda - \mu}{2\lambda(\mu + \lambda)}, \quad \text{if } \lambda \gtrapprox \mu, \text{ or} \tag{2.25}$$

$$\mathbb{E}[X_{\mathrm{E}}] \approx \frac{1}{\mu L} \cdot \mathbb{E}[\hat{Z}_{0,\mu L}] - \frac{\lambda - \mu}{2\lambda(\mu + \lambda)}, \quad \text{if } \lambda \gtrapprox \mu. \tag{2.26}$$

Figure 36. Equation. Equation (2.25) and (2.26).

These formulas do not directly tell how the expected matching distance scales with *L*. However, our numerical experiments show that when $\frac{\lambda}{\mu} \approx 1$, the distance formula behaves more similarly to the balanced case and increases monotonically with $\sqrt{L}$. As $\frac{\lambda}{\mu}$ increases, the formula value quickly converges to a fixed value, regardless of *L*.

The logic behind this somewhat counterintuitive property is that when one subset of vertices is dominating over the other, the vertices in the dominated subset are more likely to find matches locally (i.e., they only interact with the points nearby), so the expected matching distance does not increase with the length of the line. However, when the numbers of vertices in both subsets are similar, especially when they are equal, the optimal point matches tend to be found globally and they are less independent of one another (i.e., some points need to be matched with other points across the entire line). This is exactly the "correlation" issue that was discussed in Mézard and Parisi (1988). Further discussion on the scaling properties of the expected matching distance in higher-dimension continuous spaces can be found in Shen et al. (2024).

Poisson Points on Networks

Next, we are ready for the matching problem in a discrete network. Let G = (V,E) be an undirected graph with node set V and edge set E. On each edge $e \in$ E, the point subsets U_e and V_e are generated according to homogeneous Poisson processes, where $m_e = |U_e|$ and $n_e = |V_e|$ are now random variables. We now conduct matching between the two sets of points on all edges: $U = \bigcup_{e\in E} U_e$ and $V = \bigcup_{e\in E} V_e$. The average optimal matching distance in this case, denoted as X_G for a "graph," should be influenced by the graph's topology. In this report, we focus on a special type of graph with the following properties: (i) all nodes in V have the same degree, D, so the graph is D-regular; (ii) all edges in E have the same length L; and (iii) the Poisson densities, μ and λ, are respectively identical across all edges. Because all edges are translationally symmetric, we can start from one arbitrary edge and study how X_G is determined by four key parameters: μ, λ, L, and D.

We first categorize the matches occurring on an edge e into two types. We say an arbitrary point $u \in U_e$ is "locally" matched if its corresponding match point v is on the same edge, with matching distance X_G^{l} ; otherwise, it is "globally" matched, with matching distance X_G^{g} . Let α represent the probability for global matching, then from the law of total expectation, the expected distance E[X_G] can be expressed as follows:

$$\mathbb{E}[X_{\mathrm{G}}] = (1-\alpha)\cdot\mathbb{E}[X_{\mathrm{G}}^{\mathrm{l}}] + \alpha\cdot\mathbb{E}[X_{\mathrm{G}}^{\mathrm{g}}]. \tag{2.27}$$

Figure 37. Equation. Equation (2.27).

We approximate the local matching distance E[X_G^{l}] from Equations (2.23)–(2.24) as if it were from an arbitrary-length line under parameters μ, λ, L, i.e.,

$$\mathbb{E}[X_{\mathrm{G}}^{\mathrm{l}}] \approx \mathbb{E}[X_{\mathrm{E}}]. \tag{2.28}$$

Figure 38. Equation. Equation (2.28).

Next, to estimate α and E[X_G^{g}], we propose a feasible process that is expected to generate a reasonably good matching solution for every realization. It prioritizes local matching and works as follows. For each edge $e \in$ E, if $n_e \geq m_e$, we match all the m_e points in U_e with those in V_e as if they were in an isolated line segment. If $n_e < m_e$, we select n_e points from U_e that are closer to the middle of the edge and match them with all the points in V_e. The remaining $m_e - n_e$ unmatched points from U_e will seek global matches. We denote $U_e^+ \subseteq U_e$ and $V_e^+ \subseteq V_e$ as the remaining point sets on edge e after the above local matching process, respectively. For each edge e, exactly one of these two sets will be empty, and the other set will be concentrated near the ends of the edge.

As such, we estimate α by the expected fraction of globally matched points in U_e. Global matching can happen to a point in U_e only when $m_e > n_e$, so α can be estimated by the conditional expectation as follows:

$$\alpha \approx \frac{1}{\mu L} \cdot \Pr\{m_e > n_e\} \cdot \mathbb{E}[m_e - n_e \mid m_e > n_e]. \tag{2.29}$$

Figure 39. Equation. Equation (2.29).

We see that $m_e - n_e$ is the difference between two Poisson random variables, which follows a Skellam distribution and can be approximated by a normal distribution with mean $(\mu - \lambda)L$ and variance $(\lambda + \mu)L$. As such, we have:

$$\Pr\{m_e > n_e\} \approx \Phi\left(\frac{-\frac{1}{2} + (\mu - \lambda)L}{\sqrt{(\lambda + \mu)L}}\right), \tag{2.30}$$

Figure 40. Equation. Equation (2.30).

where $\Phi(\cdot)$ is the cumulative distribution function of standard normal distribution. Then, the conditional expectation is:

$$\mathbb{E}[m_e - n_e \mid m_e > n_e] = (\mu - \lambda)L + \sqrt{(\lambda + \mu)L} \cdot \frac{\phi\left(\frac{-\frac{1}{2}+(\lambda-\mu)L}{\sqrt{(\lambda+\mu)L}}\right)}{1 - \Phi\left(\frac{-\frac{1}{2}+(\lambda-\mu)L}{\sqrt{(\lambda+\mu)L}}\right)}, \tag{2.31}$$

Figure 41. Equation. Equation (2.31).

where $\phi(\cdot)$ is the probability density function of the standard normal distribution. Then, α is obtained by plugging Equations (2.30) and (2.31) into Equation (2.29). Similarly, by symmetry,

$$\Pr\{n_e > m_e\} \approx \Phi\left(\frac{-\frac{1}{2} + (\lambda - \mu)L}{\sqrt{(\lambda + \mu)L}}\right), \tag{2.32}$$

$$\mathbb{E}[n_e - m_e \mid n_e > m_e] = (\lambda - \mu)L + \sqrt{(\lambda + \mu)L} \cdot \frac{\phi\left(\frac{-\frac{1}{2}+(\mu-\lambda)L}{\sqrt{(\lambda+\mu)L}}\right)}{1 - \Phi\left(\frac{-\frac{1}{2}+(\mu-\lambda)L}{\sqrt{(\lambda+\mu)L}}\right)}. \tag{2.33}$$

Figure 42. Equation. Equation (2.32) and (2.33).

Now, all that is left is to derive an estimate of E[X_G^g]. In so doing, for all unmatched point $u \in U_e^+$, $\forall e \in$ E, we perform the following breadth-first search procedure throughout the network (as illustrated in Figure 43) to identify a feasible match globally. Step (i) is to find the nearer end of edge e from u, denoted as o_0. Identify the layer of edges, N_k, whose nearer end is kL distance away from o_0, for all k = 0,1,⋯. Step (ii), starting from k = 0, is to check whether there exists any edge $e' \in N_k$ such that $V_{e'}^+ \neq \emptyset$. If yes, match u with the nearest point $v \in \bigcup_{e' \in N_k} V_{e'}^+$ (e.g., shown as the labeled red circle in Figure

43), mark the edge containing *v* as e^* and the nearer end of e^* (to o_0) as o_k. Repeat (i) and (ii) until all points in $U_e^+, \forall e \in$ E have found a match, or all points in $V_e^+, \forall e \in$ E have been used for a match.

According to the above matching process, the distance between a specific $u \in U_e^+$ and its match $v \in V_{e'}^+$ consists of three parts (as indicated by the three gray curves in Figure 43): (i) the distance from *u* to o_0 on edge *e*; (ii) the distance from o_0 to o_k, where $e^* \in N_k$; and (iii) the distance from o_k to *v* on edge e^*.

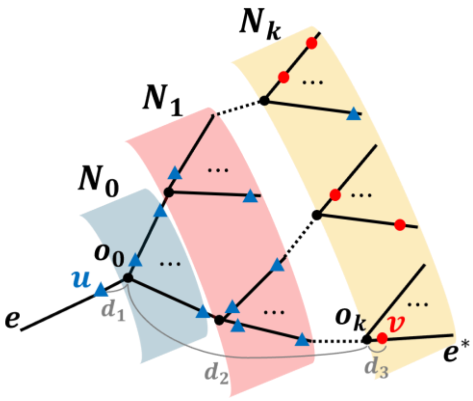


Figure 43. Graph. Illustration of the breadth-first search procedure.

Let random variables d_1, d_2, and d_3 represent these three distances, and E[X_G^g] can be written as the sum of their individual expectations; i.e.,

$$\mathbb{E}[X_{\mathrm{G}}^{\mathrm{g}}] = \mathbb{E}[d_1] + \mathbb{E}[d_2] + \mathbb{E}[d_3]. \qquad (2.34)$$

Figure 44. Equation. Equation (2.34).

First, we look at d_2. The probability of finding a match in the *k*-th layer should be no larger than the probability for at least one edge $e' \in N_k$ to have $n_{e'} > m_{e'}$. Because all edges are translationally symmetric, Pr{$n_{e'} > m_{e'}$} = Pr{$n_e > m_e$}. Also, because each node has an equal degree *D*, we have $|N_k| \leq (D-1)^{k+1}, \forall k$. Hence, approximately,[1] the probability of finding a match in the *k*-th layer is:

$$1 - (1 - \Pr\{n_e > m_e\})^{(D-1)^{k+1}}, \quad k = 0, 1, \cdots . \qquad (2.35)$$

Figure 45. Equation. Equation (2.35).

We would have $d_2 = kL$ (for $k = 0,1,2,\cdots$) if a match is successfully found in the *k*-th layer, but not in the previous layers (if any); hence, for $k = 0,1,2,\cdots$:

[1] The exactly cardinality of the layers, $|N_k|, \forall k$, can be easily counted for any *D*-regular network. However, $|N_k| \approx (D-1)^{k+1}$ is a good approximation when *k* is relatively small, and the probability quickly approaches zero as *k* increases. Hence, we use the approximation for simplicity.

$$\Pr\{d_2 = kL\} \approx (1 - \Pr\{n_e > m_e\})^{\sum_{i=0}^{k-1}(D-1)^{i+1}} \cdot [1 - (1 - \Pr\{n_e > m_e\})^{(D-1)^{k+1}}]. \quad (2.36)$$

Figure 46. Equation. Equation (2.36).

As k increases, the first term in Equation (2.36) quickly approaches zero, while the second term approaches a constant. Thus, E[d_2] can be approximated by the following truncation:

$$\mathbb{E}[d_2] \approx \sum_{k=0}^{\kappa} kL \cdot \Pr\{d_2 = kL\}, \quad (2.37)$$

Figure 47. Equation. Equation (2.37).

where κ is a relative small value (e.g., about 10).

Next, we look at d_1 and d_3. Recall that all unmatched points in U_e^+ , if any, after local matching, will be located near the two ends of edge e. The expected number of unmatched points near each end is $\frac{\mathrm{E}[|U_e^+|]}{2} = \frac{1}{2}\mathrm{E}[m_e - n_e|m_e > n_e]$. The average distance between two adjacent points among them is $\frac{1}{\mu}$. Then, the distance between an arbitrary point $u \in U_e^+$ and its nearer end o_0, i.e., d_1, is approximately uniformly distributed in the interval $(0, \frac{1}{2\mu}\mathrm{E}[m_e - n_e|m_e > n_e])$, and hence the average, E[d_1], is approximately equal to half of the interval length, i.e.,

$$\mathbb{E}[d_1] \approx \frac{1}{4\mu} \cdot \mathbb{E}[m_e - n_e \mid m_e > n_e], \quad (2.38)$$

Figure 48. Equation. Equation (2.38).

where E[m_e–n_e | $m_e > n_e$] is given by Equation (2.31). The derivation of E[d_3] is similar, but through symmetrical analysis of the unmatched points in $V_{e^*}^+$ where $e^* \in N_k$. The distance between o_k and an arbitrary unmatched point in $V_{e^*}^+$ near o_k is approximately uniformly distributed in the interval $(0, \frac{1}{2\mu}\mathrm{E}[n_e^* - m_e^*|n_e^* > m_e^*])$, and again, E[$n_{e^*} - m_{e^*}$ | $n_{e^*} > m_{e^*}$] = E[n_e–m_e | $n_e > m_e$]. However, competition may occur, and not all points in $V_{e^*}^+$ must be matched. From the perspective of a specific point $u \in U_e^+$, during the breadth-first search process, it will be matched with the nearest available point $v \in \cup_{e' \in N_k} V_{e'}^+$. However, other competing points in $\cup_{e \in E} U_e^+$ (as indicated by the other blue triangles in Figure 43) may also have the chance to be matched with the points in $\cup_{e' \in N_k} V_{e'}^+$ (shown as the red circles in Figure 43). The expected ratio between the total number of competing points of u and the total number of available points in $\cup_{e' \in N_k} V_{e'}^+$ is approximately $\frac{|U|}{|V|} = \frac{\mu}{\lambda}$. This indicates that at the end of the breadth-first search process, $\frac{\mu}{\lambda}$ of the points in $\cup_{e' \in N_k} V_{e'}^+$ will be matched.

Because $e^* \in N_k$, the distance between o_k and an arbitrary matched point $v \in V_{e^*}^+$ near o_k, i.e., d_3, is approximately uniformly distributed in the interval $(0, \frac{\mu}{2\lambda^2} \mathrm{E}[n_e - m_e | n_e > m_e])$, and hence the average, E[d_3], can be approximately estimated as follows:

$$\mathbb{E}[d_3] \approx \frac{\mu}{4\lambda^2} \cdot \mathbb{E}[n_e - m_e \mid n_e > m_e], \tag{2.39}$$

Figure 49. Equation. Equation (2.39).

where E[$n_e - m_e \mid n_e > m_e$] is given by Equation (2.33).

Summarizing all the above, E[X_G] can be estimated out of Equations (2.27)–(2.39).

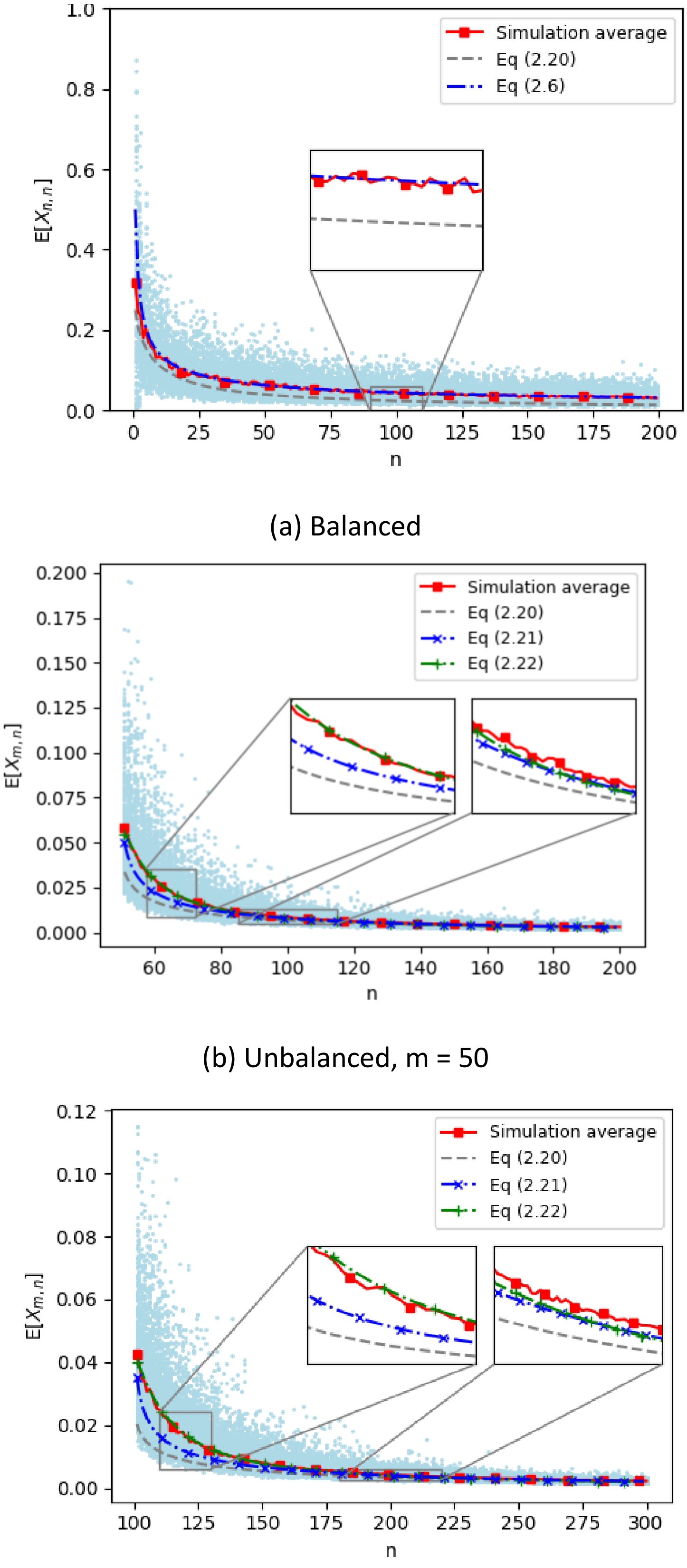
NUMERICAL EXPERIMENT

Verification of 1D RBNP

In this section, we validate the accuracy of the proposed formulas of 1D RBMP using a series of Monte-Carlo simulations. For each combination of *n* and *m* values, 100 RBMP realizations are randomly generated. For each realized instance, the optimal matching is solved by a standard linear program solver GLPK (Makhorin, 2011). The average optimal matching distances for each (*m,n*) combination is recorded as the sample mean across the 100 realizations.

Figure 50 compares the simulation results with the formulas developed for both balanced and unbalanced cases, including Equations (2.6), (2.21) (2.22), and (2.20). The optimal matching distances solved for each instance from the Monte-Carlo simulation is represented by the light-blue dots, and their sample mean is represented by the red solid curve with square markers. The estimations from Equations (2.6), (2.21), (2.22), and (2.20) are marked by the blue dash-dot curves, blue dash-dot curves with cross markers, green dash-dot curves with plus markers, and grey dash curves, respectively.

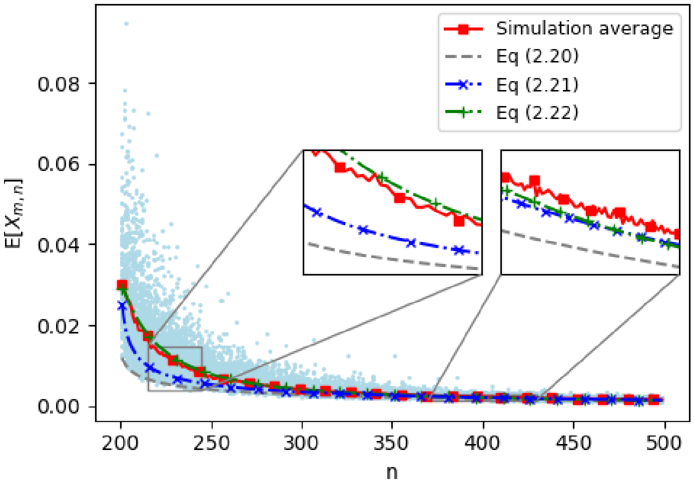
We first try the balanced cases. Let the value of *n* = *m* vary from 1 to 200. Figure 49(a) shows the results. The estimations by Equation (2.6) closely match with the simulation averages, with an average relative error of 4.57%. Meanwhile, Equation (2.20) has a larger average relative error of 43.2%. This indicates that Equation (2.6) performs significantly better than Equation (2.20). In addition, when *n* and *m* are considerably small, Equation (2.6) tends to overestimate the simulation average. However, this error diminishes rapidly as *n* increases. For instance, the relative error is 57.8% when *n* = 1, but drops significantly to 9.2% when *n* = 6. This deviation is likely due to the assumption of i.i.d. step sizes, as the correlation among the step sizes l_i generally decreases with increasing *n*. When *n* is sufficiently large (e.g., *n* > *m* + 100), Equation (2.6) can provide a very accurate estimation.



(a) Balanced

(b) Unbalanced, m = 50

(c) Unbalanced, m = 100



(d) Unbalanced, m = 200

Figure 50. Graph. Accuracy of discrete estimators.

Next, we try the unbalanced cases. We set $m \in \{50, 100, 200\}$, and let n range from $m + 1$ to a sufficiently large number $2m + 100$. Figures 50 (b)–(d) show the results. The estimations from Equation (2.22) closely match with the simulation averages across all n and m values. The average relative errors are 7.0%, 6.4%, and 5.8% for $m = 50, 100, 200$, respectively. This indicates that, in general, Equation (2.22) can provide very accurate distance predictions. We then look at the estimations from Equations (2.21) and (2.20). When $n \gg m$, both equations also match quite well with the simulation averages. Specifically, for $n \geq 2m$, the average relative errors for Equation (2.21) are 5.4%, 5.6%, and 6.2%, for $m = 50, 100, 200$, respectively. In the meantime, Equation (2.20) yields average relative errors of 12.1%, 15.2%, and 17.7% for the same m values, respectively. When $n \approx m$, larger discrepancies can be observed between the equations and the simulation averages. While Equation (2.21) still outperforms Equation (2.20), the discrepancy is notable. Recall that this discrepancy may arise from the i.i.d assumption for point selections in V^*. Observations from various V^* instances show that, when $n \approx m$, points at certain specific positions (e.g., the first or the last point when $n = m + 1$) are more likely to be selected in V^* than the others. Nevertheless, Equation (2.21) performs generally better than Equation (2.20) and provides very good estimates when $n \geq 2m$.

In summary, one can choose the most suitable formula given the specific problem setup and the required accuracy. For balanced cases, Equation (2.6) should be used. For unbalanced cases, when $n \gg m$, Equation (2.21) is recommended as it can already provide a good estimation and is computationally more efficient than Equation (2.22); otherwise, Equation (2.22) is more suitable as it provides the most accurate estimates.

Verification of Discrete Network RBMP

In this section, we validate the accuracy of the proposed formulas of arbitrary-length line RBMP and discrete network RBMP using a series of Monte-Carlo simulations. For the former, we first fix μ but vary L and λ. For the latter, we fix L and μ, but vary D and λ. For any parameter combination, 100

RBMP instances are randomly generated, each solved through Equation (2.1)–(2.2) by a standard linear program solver, and the optimal matching distances are averaged across the 100 realizations.

Figure 51 compares the simulated average distances of arbitrary-length line RBMP, represented by markers, with estimations from Equations (2.23) (for λ/μ = 1) and (2.26) (for $\lambda/\mu \in \{1.1, 1.5, 3\}$), represented by lines. The line length L varies from $\{1,3,5,7,9\}$ [du], and μ = 10 [1/du]. In general, the estimations by Equations (2.23) and (2.26) fit tightly to the simulation averages across all L and λ/μ ratios. The average relative errors by estimations are 5.07%, 6.21%, 2.97%, and 8.84% for λ/μ = $1, 1.1, 1.5, 3$, respectively. Also, the optimal matching distance increases concavely with L when λ/μ = 1. Yet, as the ratio λ/μ becomes slightly larger, both the simulated averages and the formula estimations become flatter. At higher values of λ/μ, such as 1.5 or 3, there is no significant change in the optimal matching distance with L. These observations support the earlier discussion and visually illustrate how the average optimal matching distance scales with $\sqrt{L}$ in the balanced RBMP, but is largely independent of L in the unbalanced RBMP with $\lambda \gg \mu$.

Next, we build a series of D-regular discrete networks with node degree $D \in \{3,4,6\}$. Each network has 36 total number of unit-length edges (i.e., L = 1 [du]), and μ = 5 [1/du]. We further vary λ from 5 to 25 [1/du]. Figures 52 (a)-(c) compare the simulation averages (black solid curve) with estimations by Equation (2.26) and (2.27) (dashed and dotted curves). Each Monte-Carlo simulation instance, represented by a light-blue dot, is also plotted. The estimations by Equation (2.27) fit tightly to the simulation average across all parameter combinations. The average relative errors are 8.53%, 4.73%, and 3.40% for D = $3,4,6$, respectively. This indicates that, in general, Equation (2.27) can provide very accurate predictions in a wide range of $D, L, \lambda/\mu$ combinations. In the meantime, we observe that Equation (2.26) also estimates the average distance accurately when $\lambda \gg \mu$. Specifically, when $\lambda \geq 2\mu$, the average relative errors of Equation (2.26) are 9.31%, 6.72%, and 5.96% for D = $3,4,6$, respectively. Recall that, as $\lambda \gg \mu$, there are more chances for a point in U to be matched locally, which indicates that α converges to 0 and Equation (2.26) will predict the distance almost as well as Equation (2.27).

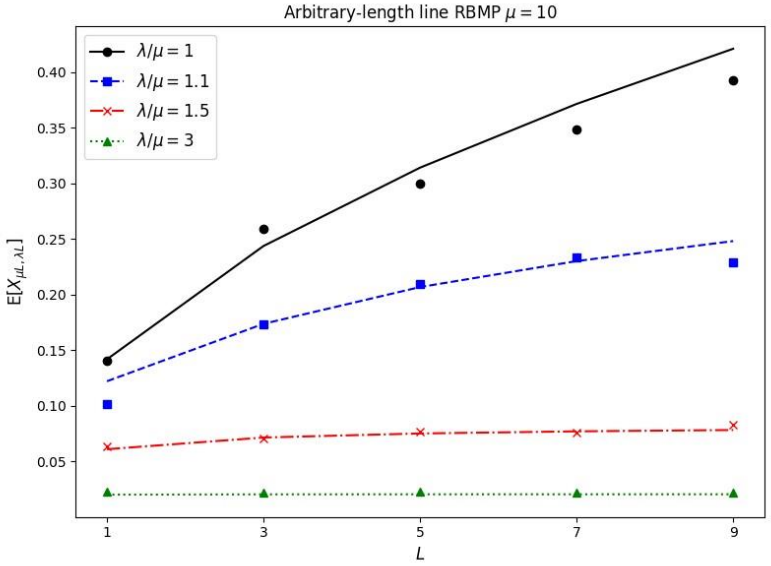
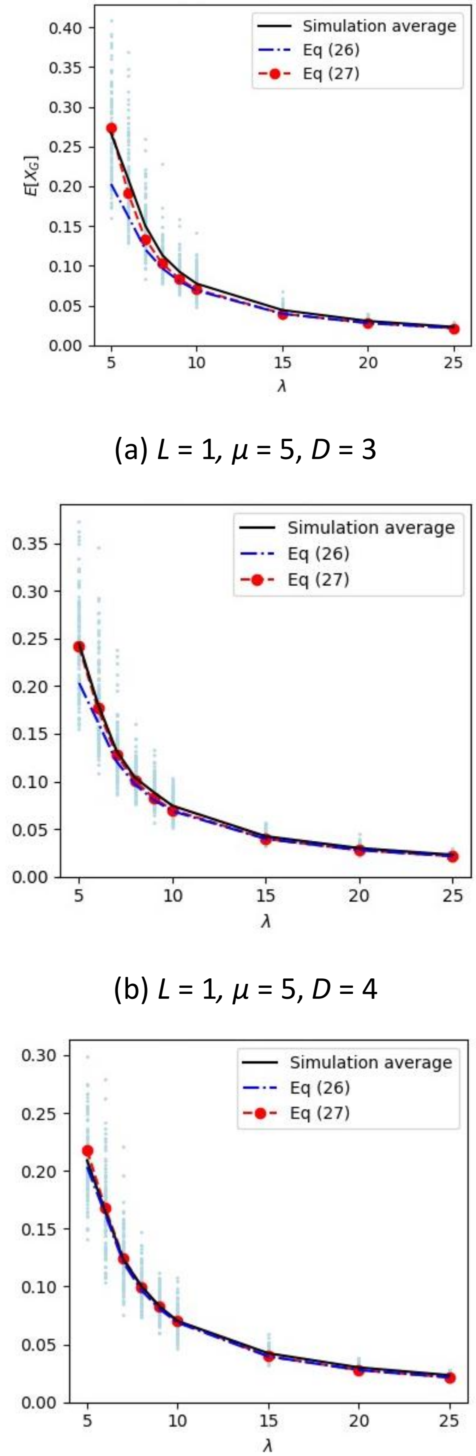


Figure 51. Graph. Verification of arbitrary-length 1D RBMP.



(a) *L* = 1, *μ* = 5, *D* = 3

(b) *L* = 1, *μ* = 5, *D* = 4

(c) *L* = 1, *μ* = 5, *D* = 6

Figure 52. Graph. Verification of discrete network RBMP.

CONCLUSION

This report presents a set of closed-form formulas, without curve fitting or statistical parameter estimation, that can provide accurate estimates for random bipartite matching problems in one-dimensional spaces and discrete networks. These formulas can be used directly in mathematical programs to evaluate and plan resources for many transportation services. In one-dimensional space, we relate the matching distance to the area between a random walk path and the x-axis, and then derive a closed-form formula for balanced matching. For unbalanced matching, we first develop a closed-form but approximate formula by analyzing the properties of unbalanced random walks following the optimal removal of a subset of unmatched points. Then, we introduce a set of recursive formulas that yields tight upper bounds based on the analysis of unbalanced random walks. The scaling property of the matching distance in arbitrary-length line segments is also discussed. Building upon these results, we derive the expected optimal matching distance in a regular graph, in which points are distributed on equal-length edges based on spatial Poisson processes, by quantifying the expected distance when the point is locally matched (i.e., matched within the same edge) or globally matched (i.e., matched across different edges). Results indicate that our proposed formulas all provide quite accurate distance estimations for one-dimensional line segments and discrete networks under different conditions.

Nevertheless, our proposed models build upon several assumptions and approximations, which may be relaxed in the future. For example, we assumed each random walk step size as an i.i.d random variable with mean l, and treat the mean step size in each balanced random walk segment as the same as that of the original unbalanced random walk. This approach directly leads to an overestimation of the matching distance, particularly when $n \gg m$. Although we propose an approximate correction term to address this issue, alternative (better) models could be explored in the future. Further analysis could also be conducted to understand how such correlations among matched pairs would affect the optimal point removals for unbalanced problems. In addition, while the distance estimator upper bound is quite accurate, it requires solving a recursive formula, which is computationally more cumbersome. Exploring alternative methods could provide simpler estimates of similar accuracy. For discrete networks, this report opens the door to many interesting new questions. For example, future research should develop methods to estimate the expected matching distance in networks with varying edge lengths, varying degrees, and heterogeneous point densities.

CHAPTER 3: PARETO-IMPROVING SWAPPING STRATEGY FOR SPATIOTEMPORAL RANDOM BIPARTITE MATCHING WITH CLOSED-LOOP RESOURCES

INTRODUCTION

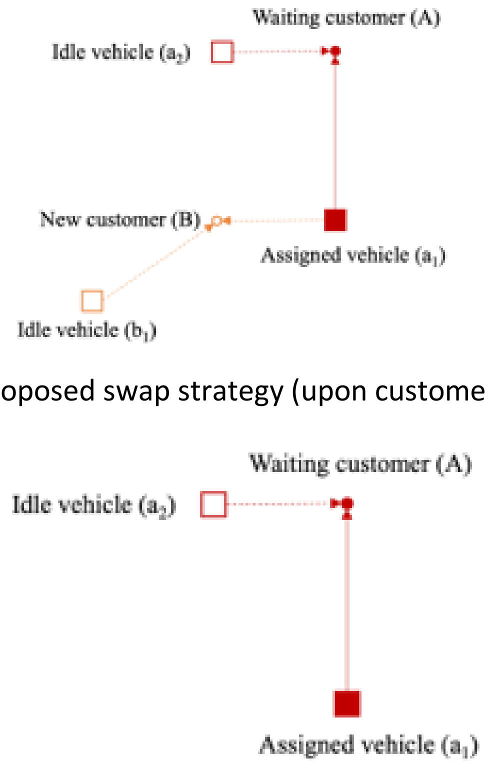
This chapter presents a newly proposed dynamic matching strategy for ST-RBMP and analyzes its performance in a closed-loop system where instant matching may already be the best operational strategy. It is inspired by a recently proposed strategy in (Ouyang and Yang, (2023b) to counteract the WGC phenomenon in e-hailing taxi systems. This strategy allows a vehicle currently assigned to pick up an existing customer to possibly surrender its duty to a newly idle vehicle closer to that customer, so as to reduce deadheading time. System performance was predicted analytically and corroborated with simulations to show significant enhancement to service efficiency against WGC. The only potential shortcoming, however, is that the originally deadheading vehicle would lose its customer and, hence, may consider this swap unfavorable, especially if the involved vehicles are chauffeured. More generally, the issue could be more severe if the two involved vehicles belong to different service operators (but are pooled via service integrators or modular chassis). Masoud et al. (2017) proposed a related strategy, referred to as ride exchange, to improve matching rate and customer retention, in which unmatched customers are allowed to purchase existing itineraries from other customers (who will be compensated via pricing mechanisms for using alternative itineraries). Simulation experiments (Masoud et al., 2017; Masoud and & Jayakrishnan, 2017) showed that this ride exchange strategy can lead to a higher matching rate than a standard first-come, first-served strategy. Yet, some open questions remain at the planning level—for example, how system performance varies with fleet sizes, with or without ride exchange, and how these relationships can be used to guide service design.

The new Pareto-improving dynamic swap strategy aims to create a win-win situation, in terms of reducing waiting/deadheading times, for all involved participants. We analytically derive a system of implicit nonlinear equations in the closed form, including a set of differential equations, to analyze the mobility system with the proposed swap strategy and to estimate the expected system performance in the steady state. Such analytical models are particularly useful, compared to simulation models, not only in revealing the intrinsic relationships between input and output quantities, but also for further analysis by the service providers or government agencies (e.g., as constraints in mathematical programs [in Ouyang et al., 2021]) to optimize resource planning and/or service offerings (e.g., eligibility for swapping and/or service integration) that can achieve a desired level of service. This system of equations is solved numerically to yield key metrics needed to quantify the system performance at equilibrium, as functions of the customer demand level and the fleet size. A series of agent-based simulation experiments are conducted to verify the accuracy of the proposed analytical formulas and to demonstrate the effectiveness of the proposed swap strategy. The results show that the proposed mathematical model can predict accurately the performance of mobility systems under the proposed swap strategy in a variety of application scenarios (e.g., integrated or modular services).

The remainder of this chapter is organized as follows. First, the basic idea of the proposed Pareto-improving vehicle swap strategy is introduced. Then, a set of analytic formulas are derived, which can predict the performance of the proposed swap strategy. Then, numerical experiments are presented, in which a set of key metrics from simulation measurements are compared with theoretical predictions. It also discusses the effectiveness of swap strategy in multiple application settings. Finally, concluding remarks and possible directions are provided for future research.

STRATEGY

Most conventional mobility systems (e.g., taxis) only use an idle vehicle to serve a new customer. The proposed Pareto-improving vehicle swap strategy allows a vehicle originally enroute to pick up an existing customer to serve newly arriving customers, and if this happens, compensates that existing customer with an available idle vehicle. Whenever a new customer (e.g., customer B in Figure 53[a]) arrives into the system, the platform will find customer B's nearest idle vehicle, b_1, as a candidate for pickup. At the same time, the platform will scan through the currently waiting customer (say, customer A) one by one to locate its currently assigned vehicle (vehicle a_1) as well as its nearest idle vehicle (vehicle a_2). If (i) customer B is closer to vehicle a_1 than customer A, (ii) vehicle a_2 is closer to customer A than vehicle a_1, and (iii) vehicle a_1 is closer to customer B than vehicle b_1, then it is beneficial to let vehicle a_1 go pick up customer B, while at the same time use vehicle a_2 to take over customer A.



(a) The proposed swap strategy (upon customer B's arrival)

(b) Simpler strategy (upon vehicle a_2 becoming idle)

Figure 53. Graph. Illustration of two dynamic vehicle swap strategies.

In practical implementations, the vehicle swap suggestion can be sent to vehicles with an expiration time limit. This time limit could be only a few seconds, similar to what conventional e-hailing taxis have to react to a new customer assignment. Once both vehicles accept the suggestion, the swaps will be conducted, and the two involved customers will simply receive a notice with updated vehicle information. If the system is operated with fully compliant drivers (e.g., full-time company employees or autonomous vehicles), such a swap can be performed instantly without any delay. Otherwise, even if one or both of the vehicles fail to respond positively within the time limit, the system can simply resend the swap suggestion later (e.g., after 1–2 min), or seek other feasible swaps that may arise in the future.

The proposed swap strategy is “Pareto improving,” as all five involved participants will experience Pareto improvements: (i) both customers A and B expect shorter waiting times because nearer vehicles are now used to pick them up; (ii) vehicle a_1 now picks up a nearer passenger, shortening its unproductive deadheading time; and (iii) idle vehicles a_2 and b_1 would not hold negative feelings because they are never aware of the swap. Such a win-win outcome is superior to the simpler swap strategy proposed in Ouyang and Yang (2023), which, as illustrated by Figure 53(b), seeks opportunities to use a newly idle vehicle a_2 to substitute for other deadheading vehicles a_1 as long as the swap shortens the waiting time of vehicles a_1’s current customer A—even though vehicle a_1 will lose customer A and may consider this swap unfavorable. These two strategies differ in terms of not only the swap time (i.e., when a customer newly arrives vs. when a vehicle newly becomes idle), but also the need to satisfy a win-win condition for all five, instead of three, involved parties.

Note that the proposed strategy, per the above description, instantly assigns a vehicle to a newly arrived customer, but this choice is for modeling convenience only. The idea of vehicle swaps can be applied regardless of how the customer-vehicle matching decisions are made, as long as potentially improving matches are sought dynamically. Other matching strategies may pool passengers and vehicles from within a spatial distance for a short period of time (e.g., a few seconds) so as to perform a many-to-many bipartite matching (Xu et al., 2018; Valadkhani and & Ramezani, 2020). We choose to assign a vehicle instantly, and allow dynamic swaps later, mainly for three reasons. First, instant vehicle assignment is simple, reasonably good, and easy to implement for many application scenarios. Second, it helps decouple the system toward analytically closed-form formulas that not only provide insights, but also serve as a convenient way to quantify system performance. Third, the disadvantage of instant vehicle assignment is exactly remedied by the proposed dynamic vehicle swaps. As a more suitable vehicle becomes available (later), we always have the opportunity to swap the originally assigned vehicle. From the perspective of a customer, the quality of service could be arguably similar (or even better), because the customer can get a vehicle assigned instantly instead of being held waiting before an assignment is made.

It is not trivial to analytically quantify the effectiveness of such a strategy on mitigating WGC and enhancing system performance in a stochastic operating environment. For a swap to be feasible, a set of conditions in the form of distance inequalities need to be satisfied simultaneously, and these conditions may be correlated to each other. Therefore, one of the tasks is to derive the probability for one possible swap to be feasible (i.e., Pareto-improving for all five involved parties). Furthermore, each new customer, upon arrival, may see multiple feasible candidates for a swap, and likewise each waiting customer may also see multiple feasible opportunities for a swap when its assigned vehicle is deadheading. Careful considerations must be made, hence, regarding such competitions among multiple feasible swap candidates. Finally, each swap may reduce a random amount of waiting/deadheading time for either involved customer/vehicle, so we must develop a method to estimate the new system equilibrium under the proposed strategy, so that the expected system performance (e.g., the expected waiting/deadheading time) can be estimated for arbitrary demand/supply settings. All these efforts will be described in the next section.

MATHEMATICAL MODEL

Queuing Network Model for Taxi Service

We consider a square service region with area size *R* km^2, where the streets form an infinitely dense grid. The generation of customer trips follow a homogeneous Poisson process in both time and space, with rate Λ trips/km^2-hour, and as such, each generated trip's origin and destination are uniformly distributed across the service region. The region employs a total number of *m* identical vehicles operating with an average speed of *v* km/hour to serve the customers. If proper units are chosen for distance and time, the value of area size and vehicle speed can both be 1, and the equivalent unitless demand $\lambda = \Lambda R^{3/2}/v$ represents the expected number of trips generated during the time in which a vehicle travels across the region.

Following the aspatial queuing network model in Daganzo (2010), we know all vehicles in the system transition among three states: idle, assigned, and in-service. Upon each customer's arrival, the system instantaneously assigns the nearest idle vehicle to the customer. Then, the assigned vehicle will start a deadheading trip from its current location to the customer's origin, while at the same time, the customer waits for pickup. After reaching the customer's origin, the vehicle becomes in-service and starts to move toward the customer's destination. When the in-service vehicle drops off the customer, it immediately becomes idle and ready for new customers.

In the steady state, the number of vehicles in each of the states are denoted as n_i, n_a, and n_s, respectively. We also denote *Y* as the total deadheading distance needed to pick up a customer and *L* the in-service distance to deliver a customer. Their respective expectations E[*Y*] and E[*L*] must satisfy the following, per Little's formula:

$$\lambda = \frac{n_s}{\mathrm{E}[L]} = \frac{n_a}{\mathrm{E}[Y]}. \qquad (3.1)$$

Figure 54. Equation. Equation (3.1).

Under the Manhattan distance metric, it is well-known that E[*L*] = α and $\mathrm{E}[Y] = \frac{\kappa}{\sqrt{n_i+1}}$, respectively, where $\alpha = \frac{2}{3}$ and $\kappa = \sqrt{\frac{\pi}{8}}$ (Daganzo, 1978). By substituting these results into Equation (3.1) and noting the fleet size $m = n_i + n_s + n_a$, we have:

$$m = n_i + \alpha\lambda + \frac{\kappa\lambda}{\sqrt{n_i + 1}}. \tag{3.2}$$

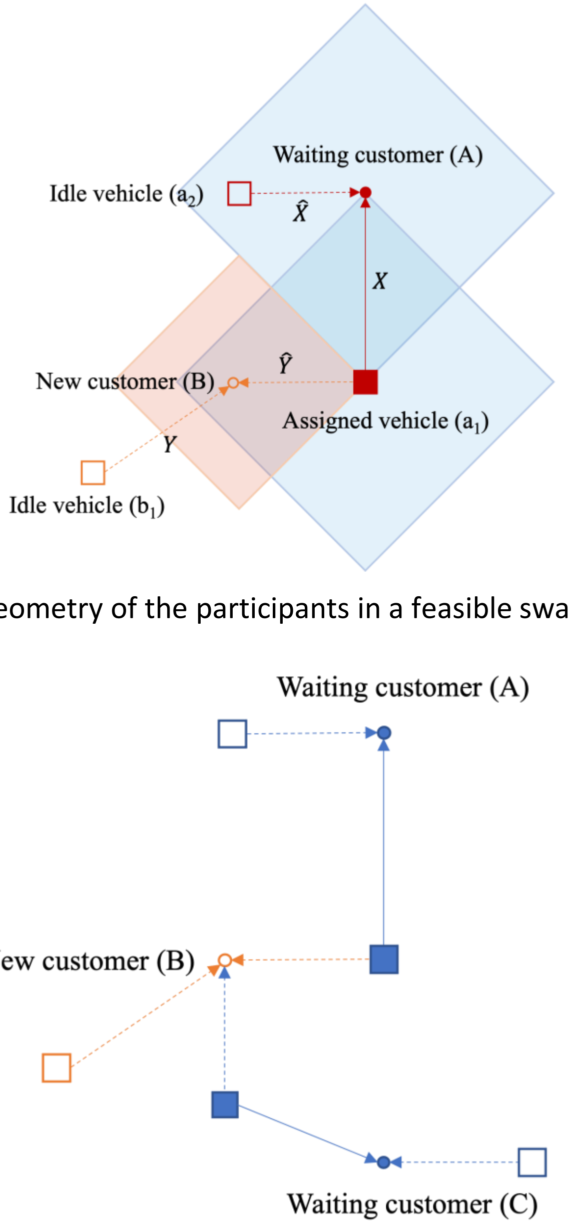
Figure 55. Equation. Equation (3.2).

It is easy to analyze the solution to Equation (3.2) in the form of equilibrium vehicle distribution, (n_i,n_a,n_s), where $n_a = \frac{\kappa\lambda}{\sqrt{n_i+1}}$, $n_s = \alpha\lambda$. At a minimum fleet size $m_{\min} = \alpha\lambda + 3\left(\frac{\kappa\lambda}{2}\right)^{\frac{2}{3}} - 1$, the equation has a single solution that satisfies $2(n_i + 1) = n_a$. When the fleet size is too small, $m < m_{\min}$, the equation has no solution (i.e., the system has no equilibrium and cannot sustain the described service). When the fleet size is sufficiently large, $m > \lambda(\alpha + \kappa)$, the equation has only one root (i.e., the system has only one equilibrium as well). For any fleet size in between, Equation (3.2) has two real-valued roots that correspond to two equilibria: one with $n_i > (m - \alpha\lambda - 2)/3 \gg 1$ and the system operates efficiently with a large number of idle vehicles; the other with a small $n_i < (m - \alpha\lambda - 2)/3$ and the system suffers from the WGC phenomenon. This type of queuing network model assumes that n_i is notably larger than 1 so as to absorb the impacts of stochasticity as well those of the region boundary on the expected deadheading distances.

To mitigate the WGC phenomenon, the proposed dynamic swap strategy aims to push the system toward the efficient equilibrium by rescuing vehicles trapped out of long deadheading trips and reducing the expected deadheading/pickup time per customer (i.e., E[Y]). To derive this expectation, we take each customer as a basic analysis unit and see what could possibly happen from arrival to pick up. In so doing, the following sections first quantifies the probability for a new customer and an arbitrary waiting customer to feasibly swap vehicles. If the new customer sees at least one feasible swap upon arrival, then an "initial" swap will occur to this new customer. If this new customer sees multiple feasible swaps upon arrival, then the competition among these feasible swaps must be addressed because only one swap per new customer can be conducted successfully. Once the new customer is assigned a vehicle (with or without an initial swap), they become a waiting customer. While waiting for pickup, their assigned vehicle may be considered by (multiple) other newly arriving customers as a candidate for a feasible swap, and some of them may be successful. Next, it is discussed that how these successful swaps (including the initial swap or those during waiting) could reduce the total waiting time experienced by this customer, and at the same time reduce the total deadheading time spent by all involved vehicles. Finally, a solution method for the derived system of analytic equations is proposed.

Probability of a Feasible Swap

As the first step, we study how a swap would be geometrically feasible to all those involved. As shown in Figure 56(a), the system checks the relative locations of the five participants to determine the feasibility of a swap whenever a new customer arrives. A swap is feasible when the following three conditions are all satisfied: (a) from customer B's perspective, the pickup distance after the swap, $\hat{Y} = |Ba_1|$, is less than or equal to the pickup distance before the swap, $Y = |Bb_1|$; (b) from vehicle a_1's perspective, the deadheading distance after the swap, $\hat{Y}$, should be no larger than the remaining deadheading distance before the swap, $X = |Aa_1|$; (c) from customer A's perspective, the pickup distance after the swap, $\hat{X}$ = |Aa$_2$|, is less than or equal to its remaining pickup distance *X*. As such, any swap satisfying all these conditions, hence deemed feasible, will improve the service experience for all involved participants and create a win-win situation.



(a) Geometry of the participants in a feasible swap

(b) Competition between candidates A and C

Figure 56. Graph. Illustration of a swap and a competition.

Now we follow the perspective of customer A. Assume customer B has just arrived, and customer A's remaining pickup distance is *x*. Conditional on *X* = *x*, the probability for a swap between customers A and B to be feasible, denoted *p*(*x*) = Pr{$\hat{Y}$ ≤ *Y*, $\hat{Y}$ ≤ *X*, $\hat{X}$ ≤ *X*|*X* = *x*}, can be derived. To control the number of swaps, system operators could introduce additional conditions into the definition of a feasible swap. For example, a set of minimal distance saving thresholds δ_A, δ_B, and δ_a can be imposed for customers A and B and vehicle a_1, respectively, by using the following feasibility conditions: $\hat{X} \leq X - \delta_A$, $\hat{Y} \leq Y - \delta_B$, and $\hat{Y} \leq X - \delta_a$. Here, we ignore the influence of region boundary and assume the location of customer B is uniformly distributed in the region. The appendix gives detailed derivations that lead to the following approximate formula:

$$p(x) \approx \frac{1}{n_i + 1} \left[1 - (1 - 2x^2)^{n_i+1}\right] \left[1 - (1 - 2x^2)^{n_i}\right]. \qquad (3.3)$$

Figure 57. Equation. Equation (3.3).

Note that Equation (3.3) is a nondecreasing function of *x*. It converges to 0 as $x \rightarrow 0^+$, and it has an upper bound of $\frac{1}{n_i+1}$ when $2x^2$ approaches 1 (i.e., when *x* approaches the longest pickup distance in the unit square).

Competition for a Successful Swap

The previous section derives, from the perspective of a waiting customer A, the conditional probability for a newly arrived customer to form a feasible swap with a waiting customer *x* distance away from its assigned vehicle. However, not every feasible swap will actually be executed, because the new customer B may see multiple feasible swap opportunities at the same time (upon its arrival). Figure 56(b), for example, shows two waiting customers (i.e., A and C) that both satisfy the feasibility conditions, so they are competitors for only one successful swap. Here, we assume that one of these feasible candidates is randomly selected for a swap. The “random” selection assumption is made for two main reasons: (i) model simplicity, as it avoids deriving the extreme value distribution of a set of random variables (e.g., distance savings from all candidates) in an analytical form; and (ii) implementation convenience, as it allows the first found feasible swap candidate (possibly via random sampling) to be implemented—other options that seek an “optimal” swap would require all potential candidates to be evaluated, which could be time-consuming. We use ω to denote the conditional probability of a feasible swap to be successfully conducted, given the new customer sees at least one feasible swap candidate. The appendix shows that, following the analysis of bipartite matching with Poisson arrivals (Ouyang and & Yang, 2023b), ω can be analytically written as follows:

$$\omega = \frac{1}{e^{\mu} - 1} \left[\mathrm{Ei}(\mu) - \ln(\mu) - \gamma\right], \qquad (3.4)$$

Figure 58. Equation. Equation (3.4).

where μ denotes the expected total number of feasible swap candidates a waiting customer (e.g., customer A) may encounter during its entire wait for pickup, $\mathrm{Ei}(\mu) = \int_{-\infty}^{\mu} \frac{e^t}{t} dt$ is the exponential integral function, and $\gamma = 0.5772$ is the Euler-Mascheroni constant. The appendix also explains why the expected number of feasible swaps seen by a new customer is also μ, and hence if we further assume a Poisson distribution, the probability for a new customer to experience an initial swap upon arrival (i.e., seeing at least one feasible swap) is $1 - e^{-\mu}$.

Interestingly, because the expected total number of initial swaps experienced by all new customers must be equal to the expected total number of successful swaps experienced by all waiting customers, and each customer is counted as both a new customer and a waiting customer, we know that the expected number of successful swaps per waiting customer must also equal $1 - e^{-\mu} < 1$. See the appendix for a more detailed explanation. This property should be considered an advantage because the customers and/or drivers (in a chauffeured system) may not favor too many swaps that could possibly make them anxious or confused.

To further derive μ, we let $\mu(x)$ be the expected total number of feasible swap candidates a waiting customer shall encounter in the remaining wait duration while its current assigned vehicle is x distance away. From customer A's perspective, in an infinitesimal time increment d_t: a new customer B shows up with probability λd_t, independent from other geometric conditions; customer A is a feasible swap candidate with probability $p(x)$; and, under that condition, customer A will be chosen by customer B for a successful swap with probability ω. Hence, there could be three possible scenarios. First, with probability $\lambda d_t p(x)\omega$, a feasible swap shows up and is actually conducted successfully for customer A in the d_t time; the remaining pickup distance for customer A jumps immediately from x to $\hat{x}$, and the value of $\mu(x)$ equals $1 + \mu(\hat{x}) = 1 + \mathrm{E}[\mu(\hat{x})|\hat{x} \leq x] = 1 + \frac{1}{x^2}\int_0^x \mu(\hat{x})2\hat{x}d\hat{x}$. The second equality comes from the uniform distribution of the location of a_2 within the diamond area centered at A (see Figure 56[a]). Second, with probability $1-\lambda d_t p(x)$, no feasible swap candidate shows up during d_t time, and vehicle a_1 will continue to move along its deadheading trip; the expected number of remaining feasible swaps equals $\mu(x-d_t)$. Finally, with probability $\lambda d_t p(x)(1-\omega)$, a feasible swap shows up in d_t time but it is not successful due to competition; the expected number of remaining feasible swaps should equal $1 + \mu(x - d_t)$. Putting all these together, function $\mu(x)$ must satisfy the following equation:

$$\mu(x) = \mu(x - d_t)\left[1 - \lambda d_t p(x)\right] + \left[1 + \mu(x - d_t)\right]\lambda d_t p(x)(1 - \omega) + \left[1 + \frac{\int_0^x \mu(\hat{x})2\hat{x}d\hat{x}}{x^2}\right]\lambda d_t p(x)\omega. \tag{3.5}$$

Figure 59. Equation. Equation (3.5).

The above can be further simplified into the following differential equation with boundary conditions $\mu(0) = 0$ and $\mu'(0) = 0$:

$$\mu'(x) = \lambda p(x)\left[1 + \frac{\omega}{x^2}\int_0^x \mu(\hat{x})2\hat{x}d\hat{x} - \omega\mu(x)\right]. \tag{3.6}$$

Figure 60. Equation. Equation (3.6).

From Equations (3.6), function $\mu(x)$ can be solved. Then, if we know a customer's initial pickup distance, denoted Z, with a probability density function of $f_Z(z)$, we can easily obtain the unconditional expectation μ as follows:

$$\mu = \int_0^{\sqrt{\frac{1}{2}}} \mu(z) f_Z(z) dz. \tag{3.7}$$

Figure 61. Equation. Equation (3.7).

Waiting/Deadheading Time

Now we show how to derive the expected waiting time experienced by a waiting customer, denoted τ. Because there is always a one-to-one correspondence between a waiting customer (e.g., customer A in Figure 56[a]) and an assigned vehicle (e.g., vehicle a_1) at any time, this expectation is equal to the expected total deadheading time of all vehicles ever assigned to this customer (i.e., $E[Y]$). Similarly to the derivation of μ, we first define $\tau(x)$ as the expected remaining waiting time of the waiting customer whose current assigned vehicle is x distance away. The value of $\tau(x)$ becomes $\tau(\hat{x}) = \frac{1}{x^2}\int_0^x \tau(\hat{x}) 2\hat{x}\, d\hat{x}$ if a swap is conducted successfully in d_t time (with a probability of $\lambda d_t p(x)\omega$), or $\tau(x–d_t)$ otherwise. As such, $\tau(x)$ satisfies the following equation:

$$\tau(x) = [d_t + \tau(x - d_t)]\,[1 - \lambda d_t p(x)\omega] + \left[d_t + \frac{1}{x^2}\int_0^x \tau(\hat{x}) 2\hat{x} d\hat{x}\right] \lambda d_t p(x)\omega, \tag{3.8}$$

Figure 62. Equation. Equation (3.8).

which can be further simplified into a differential equation with boundary conditions $\tau(0) = 0$ and $\tau'(0) = 1$, as follows:

$$\tau'(x) = \lambda p(x)\omega \left[\frac{1}{x^2}\int_0^x \tau(\hat{x}) 2\hat{x} d\hat{x} - \tau(x)\right] + 1. \tag{3.9}$$

Figure 63. Equation. Equation (3.9).

From Equations (3.9), function $\tau(x)$ can be solved. Again, if we know the probability density function of a customer's initial pickup distance $f_Z(z)$, the expected waiting/deadheading time τ is given by:

$$\tau = \int_0^{\sqrt{\frac{1}{2}}} \tau(z) f_Z(z) dz. \tag{3.10}$$

Figure 64. Equation. Equation (3.10).

Initial Pickup Distance

Now note that we need the probability distribution of initial pickup distance z to derive both μ and τ. We next derive the cumulative density function of Z, denoted $F_Z(z)$.

Recall that for any newly arrived customer (e.g., customer B in Figure 56[a]), there are two scenarios for the initial pickup distance: if there is a successful initial swap, the distance should be $\hat{Y}$, or otherwise the distance shall be Y. The approximate probability distributions of Y and $\hat{Y}$ in a closed form are discussed in the appendix. However, to derive $F_Z(z)$, we need to obtain the probabilities for Y and $\hat{Y}$ to be greater than a certain value z, conditional on whether an initial swap is conducted. To capture the probability of an initial swap, we must also obtain (i) the probability density function for a randomly chosen assigned vehicle to have a remaining deadhead distance of x at a random time, denoted $f_X(x)$; and (ii) the conditional probability for a customer, while waiting, to ever see its assigned vehicle being x distance away, denoted $P(x)$. The appendix provides the detailed derivation of $f_X(x)$, $P(x)$, and $F_Z(z)$, and the results are summarized as follows:

$$\begin{aligned} F_Z(z) =& 1-(1-2z^2)^{n_i}+\int_0^z 2x^2(1-2z^2)^{n_i}\left[1-(1-2x^2)^{n_i}\right]f_X(x)dx \\ &+\int_z^{\sqrt{\frac{1}{2}}} 2z^2(1-2z^2)^{n_i}\left[1-(1-2x^2)^{n_i}\right]f_X(x)dx, \end{aligned} \tag{3.11}$$

$$P(x)=1-F_Z(x)-\int_{z=x}^{\sqrt{\frac{1}{2}}}\int_{\hat{x}=x}^{z}\frac{x^2}{\hat{x}^2}P(\hat{x})p(\hat{x})\omega d\hat{x} f_Z(z)dz, \tag{3.12}$$

Figure 65. Equation. Equation (3.11) and (3.12).

and

$$f_X(x)=\frac{P(x)}{\int_0^{\sqrt{\frac{1}{2}}}P(x)dx}. \tag{3.13}$$

Figure 66. Equation. Equation (3.13).

Note that the system of Equations (3.3)–(3.7) and (3.11)–(3.13) must be solved together to yield $F_Z(z)$, and then the probability distribution function $f_Z(z) = dF_Z(z)/dz$.

Solving System Equilibrium and Performance Metrics

Now, we can update the queuing model. By replacing the original expected deadheading time $\mathrm{E}_l(l)$ with τ, Equation (3.2) becomes:

$$m = n_i + \alpha\lambda + \tau\lambda. \tag{3.14}$$

Figure 67. Equation. Equation (3.14).

As such, the value of n_i (as well as those of $n_a = \tau\lambda$ and $n_s = \alpha\lambda$) can be solved numerically through the system of nonlinear equations (3.3)–(3.14). A possible iterative algorithm is described as follows.

- Step 1: Initialize n_i by solving Equation (3.2); set $\omega = 1$, $f_Z(z) = 4n_i z(1 - 2z^2)^{n_i - 1}$ and $P(x) = 1$.
- Step 2: In every iteration, do the following sub-steps until the value of ω converges:
    - Solve $\mu(x)$ from Equation (3.6), then compute the new value of μ from Equation (3.7).
    - Find $P(x)$ that satisfies Equation (3.12), update $f_X(x)$ from Equation (3.13), and update $f_Z(z)$ from Equation (3.11).
    - Update the value of ω from Equation (3.4).
- Step 3: Based on current values of n_i and ω, solve $\tau(x)$ from Equation (3.9), then compute the value of τ from Equation (3.10). Update the value of n_i from Equation (3.14) and compute $n_a = \lambda\tau$ and $n_s = m - n_a - n_i$.
- Step 4: Repeat steps 2–3 until the value of n_i converges.

Depending on the value of m, this algorithm may yield zero, one, or two solutions for the vehicle distributions at equilibrium (n_i, n_a, n_s). For each equilibrium, Equations (3.3)–(3.13) also yield a number of system performance metrics—for example, the expected number of feasible swaps encountered per waiting customer μ, the expected number of successful swaps per waiting customer (or equivalently, the probability for a new customer to have an initial swap) $1 - e^{-\mu}$, the expected waiting/deadheading time τ, as well as the expected initial pickup distance $\mathrm{E}[Z] = \int_0^{\sqrt{1/2}} z dF_Z(z)$.

NUMERICAL RESULTS

Model Verification

To demonstrate the accuracy of the proposed model and algorithm, an agent-based simulation program is used to simulate a mobility service system for one type of customer (e.g., taxi). The trips are generated in a unit square from a homogeneous Poisson process with rate λ, and they are served by a fleet of size m. The program generates and tracks the customers' and vehicles' entire travel experience, including arrivals, assignments, pickups and drop-offs. To take a closer look at the effects of vehicle swaps, upon each new customer's arrival, the program will record the current number of idle vehicles (i.e., n_i) existing in the system, because varying values of n_i implies drastically different experiences for the customers. If there is no idle vehicle in the system when a new customer arrives, this customer will be considered lost; an effective customer arrival/service rate that belong to each equilibrium point, $\hat{\lambda}$, is recorded across all simulation trials as $\hat{\lambda} = \frac{1}{\alpha}(m - n_i - n_a)$, respectively.

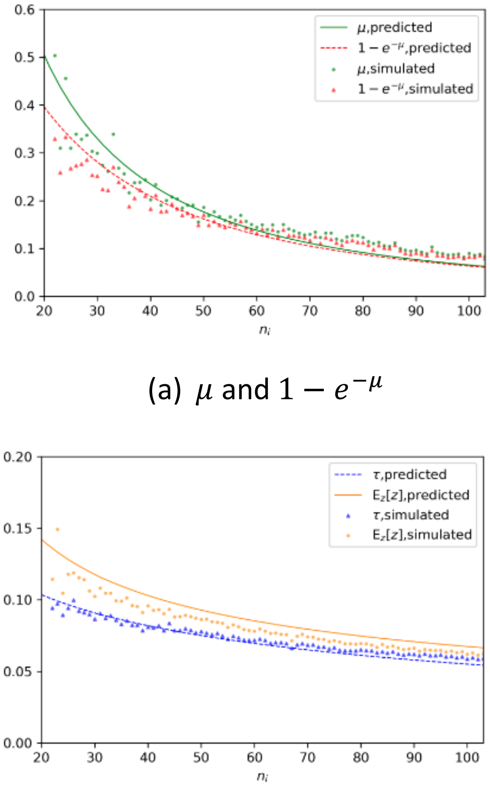
Following the data processing method by Ouyang and Yang, we also measure out of each simulation run: (i) a tuple of vehicle distributions (n_i, n_a) that represent each possible equilibrium point of the system and (ii) the subset of customers upon whose arrival the n_i value falls within the region of attraction of each equilibrium point (if multiple exist). Then, the key quantities associated with system performance metrics will also be recorded for each customer throughout its entire travel—e.g., the total number of feasible swaps seen by it upon arrival (as a new customer), the total numbers of feasible and successful swaps it encounters (as a waiting customer), its initial pickup distance, and its total waiting time. Then, we take averages of these metrics across each subset of customers, and the results are, respectively, sample estimations of μ, $1-e^{-\mu}$, E[Z] and τ for the corresponding equilibrium.

A series of numerical experiments with varying demand levels are conducted so that simulation measurements are compared with theoretical predictions. For a small town like Urbana-Champaign, assuming that the unitless customer arrival rate λ is approximately 200, the estimated minimum fleet size without swaps, from Equation (3.2), is approximately $m_{\min} = 180$. For a medium-sized city with a larger customer arrival rate $\lambda = 1{,}000$, the minimum required fleet size is approximately $m_{\min} = 804$. As such, for each of the two demand levels $\lambda \in \{200, 1000\}$, we choose a series of fleet sizes (i.e., m) that are slightly larger than $m_{\min}$. For each pair of m and λ, a total of 15 runs (with different random seeds) are conducted, each for a duration of 100 time units. For comparison, the proposed model is solved by the proposed algorithm for each pair of parameters m and the actual served demand rate $\hat{\lambda}$. At both efficient and inefficient equilibria (if both exist), the theoretical predictions of n_i,n_a, μ, $1 - e^{-\mu}$, τ and E[Z] are computed.

The results are summarized in Table 1. When the swap strategy is applied, inefficient equilibrium may still arise for very small fleets, but it occurs only very rarely. In the total 150 simulation runs (i.e., 10 cases, each 15 runs), the inefficient equilibrium is only observed in 4 runs for $\lambda = 200$, $m = 185$, and only once for $\lambda = 200$, $m = 190$. This finding is consistent with the model prediction: the swap strategy is intended to shorten the deadheading time, which is most pronounced under the inefficient equilibrium, and hence it alters the two equilibria differently and helps reduce the occurrence of the WGC. Because we only have sufficient observations for the efficient equilibrium, we compare the simulated μ, $1 - e^{-\mu}$, τ and E[Z] with their model predictions at the efficient equilibrium. Overall, Table 1 shows that our predicted values match quite well with the simulated values. This demonstrates that the proposed model is accurate in predicting the system performance under the swap strategy.

To draw further insights, we pick the case with $\lambda = 1{,}000$ and $m = 820$ and further analyze how the key performance metrics vary across different subsets of customers. The sample averages of all observed performance metrics μ, $1 - e^{-\mu}$, τ and E[Z] are computed for each subset of customers (according to $n_i \in [20, 100]$) and then plotted as the scattered points in Figure 68. The four continuous curves represent the corresponding model predictions from Equations (3.3)–(3.13). Again, the model predictions and simulation measurements match fairly well, confirming the accuracy of the model.

Figure 68 also helps us further understand qualitatively why the swap strategy helps change the system equilibrium toward reducing/mitigating the WGC phenomenon. As shown, the values of μ, $1 - e^{-\mu}$, τ and E[Z] all decrease monotonically with the number of idle vehicles n_i. This is expected because a smaller n_i implies that every customer is farther away from its nearest idle vehicle, and in this situation, per Figure 56(b): (i) a new customer B expects a longer initial pickup distance to vehicle b_1, and (ii) the waiting customers A and C may currently have longer pickup distances to their assigned vehicles as well (since they also likely have started from longer initial pickup distances). As such, an initial swap between this new customer B to either of the other waiting customers A and C is more likely to be feasible. Similarly, there is correspondingly a higher chance for each waiting customer to encounter feasible swaps, and for some of them to be successfully conducted. Hence, a swap is more likely to occur and be effective when n_i is smaller. This is aligned with our objective of rescuing vehicles trapped in long deadheading trips, and in so doing moving the system away from the inefficient equilibrium.



(a) μ and $1 - e^{-\mu}$

(b) τ and $\mathrm{E}[Z]$

Figure 68. Graph. Values of the key metrices against number of idle vehicles when λ = 1,000 and m = 820.

One may be curious whether vehicle swaps may occur frequently (or rarely) in practice. While multiple conditions need to be satisfied to trigger a feasible swap, there is a very large number of combinations of an assigned vehicle (e.g., vehicle a_1) and an idle vehicle (e.g., vehicle a_2) for each newly arriving customer (e.g., customer B). Table 1 presents the value of the probability for a new customer to have an initial swap (i.e., $1 - e^{-\mu}$) at the efficient equilibrium for a number of test cases, both predicted and simulated, which falls within the range of 0.01 to 0.1. Additionally, Figure 68(a) illustrates the value of this probability for a medium-sized case (λ = 1,000 and m = 820), which ranges from approximately 0.05 to 0.5. These results suggest that the occurrence of swaps in practical applications falls within an acceptable range (i.e., not too frequent to the extent to annoy or confuse the involved parties and also not too rare to the extent to be practically ineffective).

Table 1. Comparison of Key Metrics Under System Equilibrium: Simulation vs. Prediction

λ	m	Efficient equilibrium													Inefficient equilibrium				
		$\hat{\lambda}$	n_i		n_a		μ		$1-e^{-\mu}$		τ		E[Z]		$\hat{\lambda}$	n_i		n_a	
			Simulated	Predicted	Simulated	Predicted	Simulated	Predicted	Simulated	Predicted	Simulated	Predicted	Simulated	Predicted		Simulated	Predicted	Simulated	Predicted
200	185	198.73	29.73	32.88	22.79	19.63	0.1020	0.0767	0.0952	0.0738	0.1079	0.0988	0.1151	0.1129	209.80	12.25	1.50	32.88	51.01
	190	200.55	35.35	37.70	20.95	18.60	0.0854	0.0637	0.0809	0.0617	0.0999	0.0928	0.1054	0.1060	215.52	13.27	1.10	33.05	55.20
	195	200.14	42.06	44.40	19.51	17.17	0.0746	0.0498	0.0706	0.0486	0.0963	0.0858	0.1009	0.0982	–	–	< 1	–	60.57
	200	199.98	48.41	50.58	18.27	16.10	0.0614	0.0409	0.0595	0.0401	0.0905	0.0805	0.0940	0.0924	–	–	< 1	–	65.68
	205	198.95	55.08	57.31	17.28	15.06	0.0504	0.0336	0.0490	0.0330	0.0847	0.0757	0.0872	0.0872	–	–	<1	–	71.37
1000	810	1000.8	76.00	82.24	66.80	60.56	0.1090	0.0880	0.1017	0.0842	0.0638	0.0605	0.0683	0.0738	–	–	7.70	–	135.10
	820	1002.0	89.60	95.50	62.40	56.50	0.0971	0.0704	0.0911	0.0680	0.0613	0.0564	0.0651	0.0689	–	–	5.50	–	146.50
	830	1000.8	104.40	110.16	58.40	52.64	0.0811	0.0564	0.0771	0.0548	0.0581	0.0526	0.0611	0.0645	–	–	3.90	–	158.90
	840	990.60	125.40	132.00	54.20	47.60	0.0689	0.0419	0.0660	0.0410	0.0552	0.0480	0.0576	0.0594	–	–	2.50	–	177.10
	850	1004.7	126.00	131.93	54.20	48.27	0.0676	0.0425	0.0649	0.0416	0.0544	0.0480	0.0567	0.0594	–	–	2.60	–	177.60

Table 2. Comparison of Key Metrics Under Efficient Equilibrium: Simpler Swap vs. Pareto Swap

λ	m	n_i			μ			$1-e^{-\mu}$			τ			E[Z]		
		No swap	Simpler swap	Pareto swap	No swap	Simpler swap	Pareto swap	No swap	Simpler swap	Pareto swap	No swap	Simpler swap	Pareto swap	No swap	Simpler swap	Pareto swap
200	185	28.65	31.84	31.53	0	0.2610	0.0819	0	0.2297	0.0786	0.1151	0.0993	0.1007	0.1154	0.1154	0.1152
	190	36.09	38.19	38.24	0	0.2090	0.0622	0	0.1886	0.0603	0.1029	0.0923	0.0921	0.1055	0.1055	0.1053
	195	42.71	44.24	44.54	0	0.1740	0.0496	0	0.1597	0.0484	0.0948	0.0873	0.0857	0.0982	0.0982	0.0980
	200	48.93	50.12	50.56	0	0.1480	0.0409	0	0.1376	0.0401	0.0887	0.0823	0.0805	0.0925	0.0925	0.0924
	205	54.90	55.87	56.41	0	0.1290	0.0346	0	0.1210	0.0340	0.8381	0.0793	0.0763	0.0879	0.0879	0.0878

Furthermore, in Figure 68(a), the vertical difference between the curves of μ and $1 - e^{-\mu}$, which indicates the expected number of feasible swaps encountered per waiting customer that are not successfully conducted, decreases monotonically with n_i as well, and it gradually diminishes to 0. This means when n_i is large enough, a feasible swap will more likely be formed between a new customer with only one waiting customer, without any other competitors. Yet, when n_i is smaller, each new customer would see more competitors/candidates upon arrival, adding redundancy to the system. That is, even if some assigned vehicles could not perform swaps due to any unexpected disfunctions (e.g., driver refusal or communication disruption), the system would still have a good chance of conducting a successful swap to rescue a deadheading vehicle.

Similarly, note that in Figure 68(b), the vertical difference between the curves of τ and E[Z] represents the expected reduction of waiting time per customer due to intermediate vehicle swaps. This quantity also decreases monotonically with n_i. This is reasonable because the reduction of waiting time is positively correlated with the expected number of successful swaps per customer $1 - e^{-\mu}$ (which shows a similar trend). Note, too, that the expected reduction (both observed or predicted) is always greater than 0 across all n_i values. This implies that the swap strategy is able to effectively reduce the waiting time even when there are a large number of idle vehicles in the system.

In summary, the above results not only demonstrate reasonable accuracy of the proposed analytic model in predicting the steady-state performance of a shared mobility system with the proposed swap strategy, but also show that the strategy holds the promise to effectively help the system jump out of an inefficient equilibrium. The extent of the effectiveness is studied next by comparing system performance metrics with or without swaps.

Effectiveness of the Pareto Swap Strategy

To begin with, we conduct a comparison of system performance under three strategies: (i) no swap; (ii) the swap strategy from Ouyang and Yang (2023) (depicted in Figure 53[b], referred to as the simpler strategy); and (iii) the Pareto swap strategy proposed in this chapter (illustrated in Figure 53[a], referred to as the Pareto strategy). Table 2 provides an overview of several key performance metrics under efficient equilibrium when $\lambda = 200$. Both swap strategies effectively reduce the customer waiting time τ and slightly increase the number of idle vehicles n_i in the system, which is consistent with our expectations. Interestingly, the values of n_i and τ from both strategies are quite similar, while the values of μ and $1{-}e^{-\mu}$ from the Pareto strategy are much smaller than those from the simpler strategy. This indicates that the Pareto strategy employs fewer swaps than the simpler strategy, while achieving similar benefits in terms of reducing customer waiting time. One possible reason for this observation is that the Pareto strategy is able to reduce the initial pickup distance (i.e., Z) upon the new customer's arrival, which would not be counted as a "swap" from the perspective of that new customer, but effectively reduces the remaining pickup trip distance in a similar way.

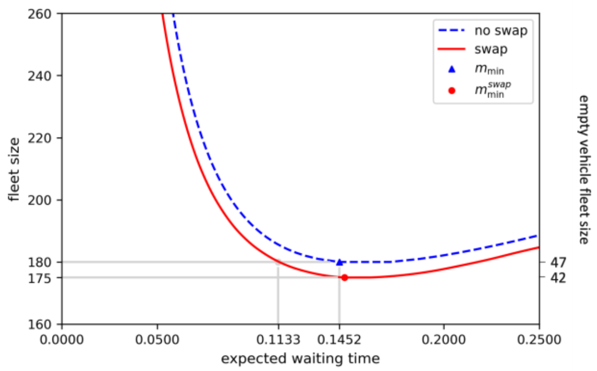
Next, we delve deeper into the Pareto strategy and compare the system performance under two practical conditions: one-type customer and multiple-type customer systems.

One Type of Customers

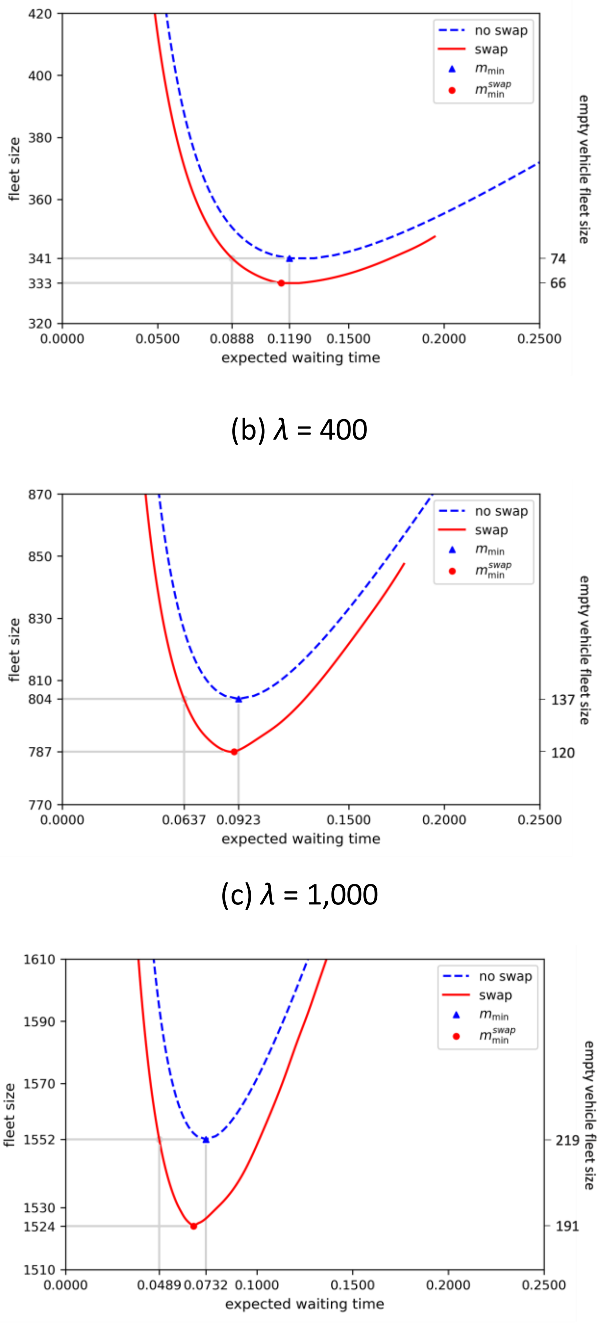
First, we compare the relationships between the expected waiting time per customer vs. the fleet size, with or without swaps, under a variety of customer arrival rates. The results are plotted in Figure 69. Only equilibrium points satisfying $n_i > 1$ are plotted.

Figure 69 shows that the swap strategy, for all demand levels, moves the curve toward the bottom and slightly toward the left. This indicates reductions of fleet size and/or waiting time, and the benefits are especially stronger when the fleet size is relatively smaller. To measure the former benefit, we note the minimum required fleet size m_{min} is reduced to $m^{\text{swap}}{}_{\text{min}}$. The expected number of in-service vehicles in the steady state $n_s = \alpha\lambda$ should be independent of the use of the swap strategy, Hence, the swap strategy must have only reduced the number of vehicles that are not yet with a passenger onboard, which we call "empty" vehicles—the reduction of empty vehicles will be used as a more direct metric to measure the change in fleet size requirements. To measure the latter benefit, we measure the change of waiting time with a specific fleet size $m = m_{\text{min}}$ for the corresponding demand; see Figure 69.

As shown in Figure 69, when $\lambda \in \{200, 400, 1000, 2000\}$, the swap strategy notably reduces the required empty vehicle fleet size from 47 to 42 (10.64% reduction), from 74 to 66 (10.81% reduction), from 137 to 120 (12.41% reduction), and from 219 to 191 (12.79% reduction), respectively. Meanwhile, the expected waiting times at fleet sizes $m = m_{\text{min}}$ are reduced from 0.1452 to 0.1133 (21.99% reduction), from 0.1190 to 0.0888 (25.32% reduction), from 0.0923 to 0.0637 (31.00% reduction), and from 0.0732 to 0.0489 (33.26% reduction), respectively. When λ increases, the benefits from the swap strategy also increase, possibly because under higher demand and a larger fleet size, there are generally more opportunities to match customers with vehicles and more opportunities for them to form feasible swaps.



(a) $\lambda = 200$



(b) λ = 400

(c) λ = 1,000

(d) λ = 2,000

Figure 69. Graph. Fleet size against waiting times for one-type customer system.

Multiple Types of Customers

Now, we consider a slightly different situation where the mobility system is serving multiple types of customers (e.g., Uber vs. Lyft customers or passenger vs. freight customers) through a pooled fleet that is enabled by either a service integrator and/or use of modular chassis. We aim to see how the swap strategy may impact the system, and how it can be synergistic with the pooling of vehicle fleet.

To this end, we assume there are Q types of customer trips in the same service region, each of which, indexed by $q = 1, \cdots, Q$, is generated from an independent Poisson process with rate $\lambda_q > 0$.

Conventionally, the demand needs to be served by multiple distinct vehicle fleets of sizes $\{m_q, \forall q\}$, which are probably managed by different operators. To stay focused, we further assume that in a competitive market, the operator(s) may feel compelled to provide a similar level of service (e.g., waiting time) to all customers. As such, without vehicle swaps, the required fleet sizes, m_q, and the number of idle vehicles at equilibrium, n_{iq}, must satisfy the following variant of Equation (3.2):

$$m_q = n_{iq} + \alpha\lambda_q + \frac{\kappa\lambda_q}{\sqrt{n_{iq}+1}}, \forall q \in \{1, \cdots, Q\}$$

Figure 70. Equation. Variant of Equation (3.2).

Note, too, that the expected waiting time of type-q customers, $\frac{\kappa}{\sqrt{n_{iq}+1}}, \forall q \in \{1, \cdots, Q\}$, is solely dependent on n_{iq}. Hence, equal waiting time across all customers implies that n_{iq} should be equal for all $q \in \{1, \cdots, Q\}$, and their summation should be the total idle vehicle number for the entire system n_i; i.e., $n_{iq} = n_i/Q, \forall q \in \{1, \cdots, Q\}$. Hence,

$$\begin{aligned} m = \sum_{q\in\{1,\cdots,Q\}} m_q &= \sum_{q\in\{1,\cdots,Q\}} n_{iq} + \alpha \sum_{q\in\{1,\cdots,Q\}} \lambda_q + \sum_{q\in\{1,\cdots,Q\}} \frac{\kappa\lambda_q}{\sqrt{n_{iq}+1}} \\ &= n_i + \alpha \sum_{q\in\{1,\cdots,Q\}} \lambda_q + \frac{\kappa \sum_{q\in\{1,\cdots,Q\}} \lambda_q}{\sqrt{n_i/Q+1}}. \end{aligned} \tag{3.15}$$

Figure 71. Equation. Equation (3.15).

When the fleets are pooled together, all customers may be served by any vehicle. The system becomes one with a single type of customer, with a total demand rate $^{\text{P}}{}_{q\in\{1,\cdots,Q\}}\lambda_q$. The queuing model directly yields the following:

$$m = n_i + \alpha \sum_{q\in\{1,\cdots,Q\}} \lambda_q + \frac{\kappa \sum_{q\in\{1,\cdots,Q\}} \lambda_q}{\sqrt{n_i+1}}. \tag{3.16}$$

Figure 72. Equation. Equation (3.16).

It can be immediately seen from Equation (3.15) and Equation (3.16) that, in order to achieve the same waiting time (i.e., the last terms), the pooled fleet requires a smaller total number of idle vehicles in the system.

Next, we examine the impacts of the swap strategy on the “distinct fleet” service. We take a system with $Q = 2$, $\lambda_1 = 1{,}000$, and $\lambda_2 = 2{,}000$ as an example. The relationships between the expected waiting time per customer vs. the total fleet size in the system, with or without vehicle swaps, are plotted as the dash-dotted and dotted curves in Figure 73, respectively. As expected, the swap strategy moves the curve toward the left and toward the bottom and reduces the required empty vehicle number from 361 to 320 (11.36% reduction). Meanwhile, the expected customer waiting time at fleet size $m = m_{\min}$ of the distinct fleet service is also reduced from 0.0786 to 0.0545 (30.61% reduction).

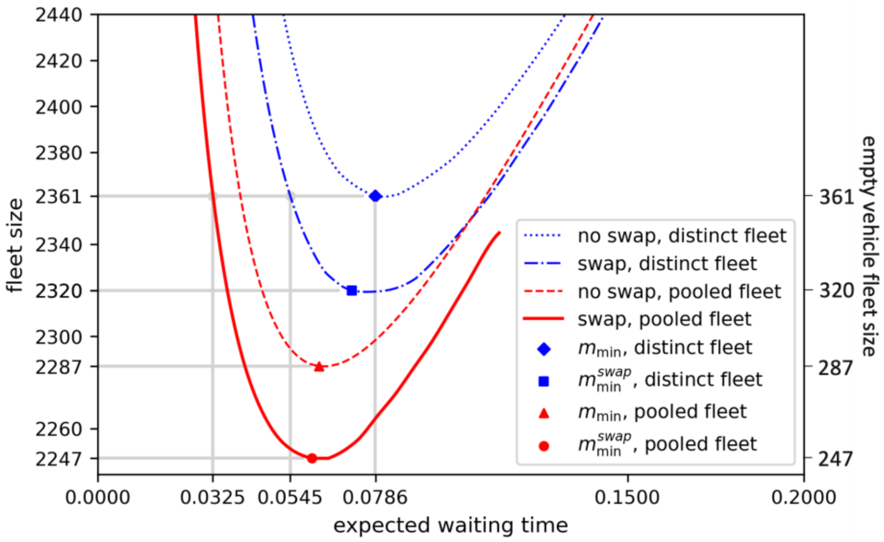


Figure 73. Graph. Fleet sizes against waiting times for two-type customer system.

Then, we further examine the effect of the swap strategy when it is used together with the "pooled fleet" service. For the same example, the total customer demand rate is $\lambda = \lambda_1 + \lambda_2 = 3{,}000$. The relationship between the expected waiting time per customer vs. the total fleet size, with or without swaps, are also plotted as the solid and dashed curves in Figure 73, respectively. It is interesting to observe how, for this case, the pooled-fleet service alone can provide notable benefits, even compared to the distinct-fleet service with swaps. Finally, the system performance can be further improved by combining the swap strategy with the pooled-fleet service, as shown by the solid curve in Figure 73. As compared to the distinct-fleet service without swaps, the minimum required empty vehicle fleet size (besides 2,000 in-service vehicles) decreases from 361 to 247, representing a 31.58% reduction. Meanwhile, when $m = m_{\min}$ of the distinct-fleet service, the expected customer waiting time moves from 0.0786 to 0.0325, which is a 58.59% drop. These superior performance demonstrate the synergistic multiplier effect provided by combining both the pooled-fleet service and the vehicle swap strategy.

CONCLUSION

This chapter proposes a generalized Pareto-improving dynamic swap strategy for shared mobility systems so as to reduce the expected waiting/deadheading time and enhance the overall operational efficiency. The proposed strategy is expected to mitigate the so-called WGC phenomenon and reduce the total required vehicle resources. A set of approximate analytic formulas are derived to predict the system performance in the steady state, and they can be solved numerically as a system of nonlinear equations. The accuracy of the proposed models is verified by comparing the model outputs with agent-based simulations through a series of experiments. Then, the impacts of the swap strategy on distinct and pooled fleets are studied by comparing the performance of mobility systems with and without swaps. The results show that the proposed swap strategy can provide benefits from reducing both customer waiting time and fleet size requirements for serving one or multiple type(s) of customers.

The current model incorporates several simplifying assumptions, which could be relaxed or adapted in future explorations. First, the queuing model presented assumes instant vehicle assignments (in order to derive closed-form formulas), while many real-world mobility services use demand pooling and batch vehicle assignments. Given that the idea of vehicle swaps is applicable regardless of the method used for customer-vehicle matching decisions, we are particularly interested in deriving analytical formulas to estimate batch matching distances. By doing so, we can further explore the benefits of the swap strategy in the context of batch matching and compare them with those under instant matching, which would be very insightful. Second, while deriving the approximated conditional probabilities (e.g., $p(x)$), we make the assumption of independence among some distance variables and geometric conditions. For instance, condition (c) in the appendix is currently considered relatively independent of conditions (a) and (b). Relaxing such independence assumptions could result in a more accurate representation of the system dynamics. Third, we assume that a swap is selected randomly among all feasible candidates. Alternative and more advanced criteria for selecting an “optimal” swap candidate can be explored (e.g., one that maximizes the reduced deadheading distance from the swap). Further research in this direction can be done either analytically or numerically to provide additional insights into the effectiveness of the proposed vehicle swap strategy. Lastly, several of the derived approximate formulas (e.g., those in the appendix) assume some sort of Poisson distribution; this assumption ignores the impact of the service region boundary and potential spatial heterogeneity across vehicles and customers. Such a heterogeneity may be addressed approximately by assuming that the Poisson expectation of each vehicle or customer follows a gamma expectation, which will lead to a negative binomial distribution instead of a Poisson distribution. Examples of such treatments can be found in Ouyang et al. (2021).

More importantly, the proposed swap strategy may also entail certain costs and challenges in real-world settings, and these practical issues should be addressed in future research. For example, the proposed model does not take into account any potential transaction cost for a vehicle swap, such as the time needed for a driver to accept/reject a new pickup task, the potential time required for a modular chassis to be assembled with a customized cabin, or the potential confusion and anxiety that a swap may bring to the involved participants. In reality, operators may also want to control the number of swaps per customer or vehicle—e.g., by ensuring that the potential benefits from a swap exceed certain thresholds (e.g., transaction costs). An extension of the model, therefore, is to introduce stronger feasibility conditions that must be satisfied in order to trigger a swap. In addition, the model currently does not discriminate customers with priorities. In the real world, however, customers may have different service quality expectations (e.g., some customers may desire faster service in accordance with a specific pricing scheme or passenger demand may have a higher time value than freight demand). Therefore, the model can be extended by considering discriminative customer service strategies. Finally, because the system performance is determined by several key input parameters such as customer arrival rate and available fleet size, an optimization model can be further developed to help the operators to better optimize their service offerings.

PROJECT OUTPUTS, OUTCOMES, AND IMPACTS

OUTPUTS

In this project, we propose a new dynamic vehicle swap strategy that can be used to enhance system efficiency by reducing the expected waiting/deadheading time. We also present a set of closed-form formulas, without curve-fitting, that can provide accurate average distance estimates for one-dimensional random bipartite matching problems. Results are concluded in two papers listed below:

- Journal article: Shiyu Shen & Yanfeng Ouyang. (2023). Dynamic and Pareto-improving swapping of vehicles to enhance integrated and modular mobility services. *Transportation Research Part C: Emerging Technologies*, https://doi.org/10.1016/j.trc.2023.104366

- Preprint: Yuhui Zhai, Shiyu Shen & Yanfeng Ouyang. Average Distance of Random Bipartite Matching in Discrete Networks, Arxiv, https://arxiv.org/abs/2409.18292.

OUTCOMES

In this project, we propose an analytical model with closed-form formulas (without statistical curve fitting) that estimate the expectation of the optimal matching distance for static RBMP, where the bipartite vertices are distributed randomly over a discrete network. These formulas can be incorporated into queuing and optimization models to identify the best operational strategies in on-demand mobility systems with closed- or open-loop resource arrivals. It helps determine the optimal decision timing for whether newly arriving customers should be matched instantly or pooled into a batch for matching and for ST-RBMPs with closed-loop resources, where arriving customers shall be matched instantly. The objective is to propose a Pareto-improving strategy that allows matched vertices to be swapped among candidates with improved matching distances as the system evolves. This strategy could enhance system efficiency by reducing the overall expected matching distance and mitigating the so-called Wild Goose Chase phenomenon. Approximate analytic formulas can be derived from a series of differential equations and spatial probability models to estimate the expected system performance in the steady state.

IMPACTS

The results from this project aimed to address the challenges faced by on-demand mobility operators in understanding and addressing spatiotemporal random bipartite matching problems (ST-RBMPs). At the planning level, we developed analytical models to estimate the expected system performance in a static RBMP. Better understanding of system performance will guide the current operators to improve the service quality. At the operational level, we designed solution algorithms to improve the overall service efficiency in ST-RBMPs with different types of supply arrivals. Although our main focus was on the application of on-demand mobility services, these models can also be applied to other contexts such as resource allocation, target detection, etc. Results found in this project may provide a clearance guidance toward the rapid evolution of autonomous vehicles that is anticipated to reshape the shared mobility market very soon.

REFERENCES

Abeywickrama, T., Liang, V., & Tan, K.-L. (2022). Bipartite matching: What to do in the real world when computing assignment costs dominates finding the optimal assignment. *SIGMOD Rec.*, *51*(1), 51–58.

Addario-Berry, L., & Reed, B. A. (2008). *Ballot theorems, old and new*. Springer Berlin Heidelberg.

Afèche, P., Caldentey, R., & Gupta, V. (2022). On the optimal design of a bipartite matching queueing system. *Operations Research*, *70*(1), 363–401.

Afèche, P., Liu, Z., & Maglaras, C. (2018). Ride-hailing networks with strategic drivers: The impact of platform control capabilities on performance. *SSRN Journal*, *25*(5), 1623–1998. https://doi.org/10.1287/msom.2023.1221

Agatz, N., Erera, A., Savelsbergh, M., & Wang, X. (2012). Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, *223*(2), 295–303.

Aloqaily, M., Bouachir, O., Al Ridhawi, I., & Tzes, A. (2022). An adaptive UAV positioning model for sustainable smart transportation. *Sustainable Cities and Society*, 78, 103617.

Arnott, R. (1996). Taxi travel should be subsidized. *Journal of Urban Economics*, *40*(3), 316– 333.

Ashlagi, I., Burq, M., Dutta, C., Jaillet, P., Saberi, A., & Sholley, C. (2023). Edge-weighted online windowed matching. *Mathematics of Operations Research*, *48*(2), 999–1016.

Asratian, A. S., Denley, T. M. J., & Häggkvist, R. (1998). *Bipartite graphs and their applications*. Cambridge University Press.

Aydin, O. F., Gokasar, I., & Kalan, O. (2020). Matching algorithm for improving ridesharing by incorporating route splits and social factors. *PLoS ONE*, *15*(3), e0229674.

Banks, N. (2020). Tech start-up pix moving uses self-driving ideas to make flexible cities. *Forbes,* February 9. https://www.forbes.com/sites/nargessbanks/2020/02/09/autonomous-drive-pix-moving/

Caracciolo, S., D’Achille, M., & Sicuro, G. (2017). Random Euclidean matching problems in one dimension. *Physical Review E*, *96*(4).

Caracciolo, S., Lucibello, C., Parisi, G., & Sicuro, G. (2014). Scaling hypothesis for the Euclidean bipartite matching problem. *Physical Review E*, *90*(1).

Castillo, J. C., Knoepfle, D., & Weyl, G. (2017). Surge pricing solves the wild goose chase. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pp. 241–242.

Cheng, E. (2022). China’s capital city loosens robotaxi restrictions for Baidu, Pony.ai in a big step toward removing human taxi drivers. *CNBC*, April 27.

Cusack, J. (2021). How driverless cars will change our world. *BBC*, November 29.

Daganzo, C. F. (1978). An approximate analytic model of many-to-many demand responsive transportation systems. *Transportation Research*, *12*(5), 325–333.

Daganzo, C. F. (2010). *Public transportation systems: Basic principles of system design, operations planning and real-time control*. Institute of Transportation Studies, University of California, Berkeley.

Daganzo, C. F., & Ouyang, Y. (2019a). A general model of demand-responsive transportation services: From taxi to ridesharing to dial-a-ride. *Transportation Research Part B: Methodological*, *126*, 213–224.

Daganzo, C. F., & Ouyang, Y. (2019b). *Public Transportation Systems*. World Scientific.

Daganzo, C. F., Ouyang, Y., & Yang, H. (2020). Analysis of ride-sharing with service time and detour guarantees. *Transportation Research Part B: Methodological*, 140, 130–150.

Daganzo, C. F., & Smilowitz, K. R. (2004). Bounds and approximations for the transportation problem of linear programming and other scalable network problems. *Transportation Science*, *38*(3), 343–356.

Ding, Y., McCormick, S. T., & Nagarajan, M. (2021). A fluid model for one-sided bipartite matching queues with match-dependent rewards. *Operations Research*, *69*(4), 1256–1281.

Ezaki, T., Fujitsuka, K., Imura, N., & Nishinari, K. (2024). Drone-based vertical delivery system for high-rise buildings: Multiple drones vs. a single elevator. *Communications in Transportation Research*, 4, 100130.

Fahrbach, M., Huang, Z., Tao, R., & Zadimoghaddam, M. (2022). Edge-weighted online bipartite matching. *Journal of the ACM*, *69*(6), 1–35.

Fedtke, S., & Boysen, N. (2017). A comparison of different container sorting systems in modern rail-rail transshipment yards. *Transportation Research Part C: Emerging Technologies*, *82*, 63–87.

Feng, Y., & Niazadeh, R. (2020). Batching and optimal multi-stage bipartite allocations. *SSRN Electronic Journal*.

Frieze, A., & Karoński, M. (2015). *Introduction to random graphs*. Cambridge University Press.

Gardner, G. (2021). Hino Motors, Ree Automotive Partner to bring new technologies to commercial vehicles. *Forbes*.

Georgiev, D., & Liò, P. (2020). Neural bipartite matching.

Harel, A. (1993). Random walk and the area below its path. *Mathematics of Operations Research*, *18*(3), 566–577.

Harper, J. (2020). Can robotaxis ease public transport fears in China? *BBC News*.

Herbawi, W. M., & Weber, M. (2012). A genetic and insertion heuristic algorithm for solving the dynamic ridematching problem with time windows. In *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference - GECCO '12*, pp. 385.

Hernández, D., Cecília, J. M., Calafate, C. T., Cano, J.-C., & Manzoni, P. (2021). The Kuhn-Munkres algorithm for efficient vertical takeoff of UAV swarms. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pp. 1–5.

Jonker, R., & Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, *38*(4), 325–340.

Karp, R. M., Vazirani, U. V., & Vazirani, V. V. (1990). An optimal algorithm for on-line bipartite matching. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC ’90, pp. 352–358, Association for Computing Machinery.

Ke, J., Wu, G., Xu, Z., Yang, H., Yin, Y., & Ye, J. (2021). System and method for determining passenger-seeking ride-sourcing vehicle navigation.

Kolodny, L. (2022). Cruise gets green light for commercial robotaxi service in San Francisco. *CNBC*.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*, 83–97.

Lei, C., Jiang, Z., & Ouyang, Y. (2019). Path-based dynamic pricing for vehicle allocation in ridesharing systems with fully compliant drivers. *Transportation Research Procedia*, *38*, 77–97.

Li, D., Huang, Y., & Qian, L. (2022). Potential adoption of robotaxi service: The roles of perceived benefits to multiple stakeholders and environmental awareness. *Transport Policy*, S0967070X22001858.

Liang, Y., Li, D., Zhao, J., Ding, X., Lian, H., Hao, J., & He, R. (2023). Enhancing dynamic on-demand food order dispatching via future-informed and spatial-temporal extended decisions. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 4702–4708.

Liu, Y., & Ouyang, Y. (2021). Mobility service design via joint optimization of transit networks and demand-responsive services. *Transportation Research Part B: Methodological*, 151, 22–41.

Liu, Y., & Ouyang, Y. (2023). Planning ride-pooling services with detour restrictions for spatially heterogeneous demand: A multi-zone queuing network approach. *Transportation Research Part B: Methodological*, 174, 102779.

Makhorin, A. (2011). *GLPK (GNU Linear Programming Kit) v4.65*. Department for Applied Informatics, Moscow Aviation Institute, Moscow. https://www.gnu.org/software/glpk/glpk.html.

Masoud, N., & Jayakrishnan, R. (2017). A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system. *Transportation Research Part B: Methodological*, 106, 218–236.

Masoud, N., Lloret-Batlle, R., & Jayakrishnan, R. (2017). Using bilateral trading to increase ridership and user permanence in ridesharing systems. *Transportation Research Part E: Logistics and Transportation Review*, 102, 60–77.

Mehta, A. (2013). Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science*, 8(4), 265–368.

Mézard, M., & Parisi, G. (1985). Replicas and optimization. *Journal de Physique Lettres*, *46*(17), 771–778. http://dx.doi.org/10.1051/jphyslet:019850046017077100

Mézard, M., & Parisi, G. (1988). The Euclidean matching problem. *Journal de Physique*, *49*(12), 2019–2025.

Najmi, A., Rey, D., & Rashidi, T. H. (2017). Novel dynamic formulations for real-time ridesharing systems. *Transportation Research Part E: Logistics and Transportation Review*, *108*, 122–140.

Ouyang, Y., & Yang, H. (2023). Measurement and mitigation of the “wild goose chase” phenomenon in taxi services. *Transportation Research Part B: Methodological*, 167, 217–234.

Ouyang, Y., Yang, H., & Daganzo, C. F. (2021). Performance of reservation-based carpooling services under detour and waiting time restrictions. *Transportation Research Part B: Methodological*, 150, 370–385.

Panigrahy, N. K., Basu, P., Nain, P., Towsley, D., Swami, A., Chan, K. S., & Leung, K. K. (2020). Resource allocation in one-dimensional distributed service networks with applications. *Performance Evaluation*, 142, 102110.

Psaraftis, H. N., Wen, M., & Kontovas, C. A. (2016). Dynamic vehicle routing problems: Three decades and counting. *NETWORKS*, *67*(1), 3–31.

Ricci, M. (2015). Bike sharing: A review of evidence on impacts and processes of implementation and operation. *Research in Transportation Business & Management*, *15*, 28–38.

Salanova, J. M., Estrada, M., Aifadopoulou, G., & Mitsakis, E. (2011). A review of the modeling of taxi services. *Procedia - Social and Behavioral Sciences*, *20*, 150–161.

Seth, A., James, A., Kuantama, E., Mukhopadhyay, S., & Han, R. (2023). Drone high-rise aerial delivery with vertical grid screening. *Drones*, 7(5).

Shanks, M., Yu, G., & Jacobson, S. H. (2023). Approximation algorithms for stochastic online matching with reusable resources. *Mathematical Methods of Operations Research*, *98*(1), 43–56.

Shen, S., & Ouyang, Y. (2023). Dynamic and pareto-improving swapping of vehicles to enhance integrated and modular mobility services. *Transportation Research Part C: Emerging Technologies*, *157*, 104366.

Shen, S., Ouyang, Y., Ren, S., Chen, M., & Zhao, L. (2021a). Design and implementation of zone-to-zone demand responsive transportation systems. *Transportation Research Record*, *2675*(7), 275–287.

Shen, S., Ouyang, Y., Ren, S., & Zhao, L. (2021b). Path-based dynamic vehicle dispatch strategy for demand responsive transit systems. *Transportation Research Record*, *2675*(10), 948–959.

Shen, S., Zhai, Y., & Ouyang, Y. (2024). Expected bipartite matching distance in a D-dimensional L^p space: Approximate closed-form formulas and applications to mobility services. https://arxiv.org/abs/2406.12174

Simmons, B. I., Cirtwill, A. R., Baker, N. J., Wauchope, H. S., Dicks, L. V., Stouffer, D. B., & Sutherland, W. J. (2019). Motifs in bipartite ecological networks: uncovering indirect interactions. *Oikos*, *128*(2), 154–170.

Stiglic, M., Agatz, N., Savelsbergh, M., & Gradisar, M. (2015). The benefits of meeting points in ride-sharing systems. *Transportation Research Part B: Methodological*, *82*, 36–53.

Tafreshian, A. & Masoud, N. (2020). Trip-based graph partitioning in dynamic ridesharing. *Transportation Research Part C: Emerging Technologies*, *114*, 532–553.

Tanay, A., Sharan, R., Kupiec, M., & Shamir, R. (2004). Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proceedings of the National Academy of Sciences*, *101*(9), 2981–2986.

Valadkhani, A. H., & Ramezani, M. (2020). *Designing a dynamic matching method for ride-sourcing systems*. Institute of Transport and Logistics Studies (ITLS) Series: ITLS-WP-20-01.

Varisteas, G., Frank, R., & Robinet, F. (2021). RoboBus: A diverse and cross-border public transport dataset. In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 269–274.

Wang, H., & Wang, Z. (2020). Short-term repositioning for empty vehicles on ride-sourcing platforms. In *Proceedings of the Second Triennial Conference*, Arlington, VA, 2020 May 27-29.

Wang, K., Qian, X., Fitch, D. T., Lee, Y., Malik, J., & Circella, G. (2022). What travel modes do shared e-scooters displace? A review of recent research findings. *Transport Reviews*, 1–27.

Wang, X., He, F., Yang, H., & Oliver Gao, H. (2016). Pricing strategies for a taxi-hailing platform. *Transportation Research Part E: Logistics and Transportation Review*, *93*, 212–231.

Wu, S., Sun, F., Zhang, W., Xie, X., & Cui, B. (2022). Graph neural networks in recommender systems: A Survey. *ACM Computing Surveys*, *55*(5), 1–37.

Xu, Z., Li, Z., Guan, Q., Zhang, D., Li, Q., Nan, J., Liu, C., Bian, W., & Ye, J. (2018). Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 905–913.

Xu, Z., Yin, Y., & Ye, J. (2020). On the supply curve of ride-hailing systems. *Transportation Research Part B: Methodological*, *132*, 29–43.

Yang, H., Leung, C., Wong, S., & Bell, M. (2010). Equilibria of bilateral taxi–customer searching and meeting on networks. *Transportation Research Part B: Methodological*, *44*, 1067–1083.

Yu, H., Ye, W., Feng, Y., Bao, H., & Zhang, G. (2020). Learning bipartite graph matching for robust visual localization. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 146–155.

Zha, L., Yin, Y., & Xu, Z. (2018). Geometric matching and spatial pricing in ride-sourcing markets. *Transportation Research Part C: Emerging Technologies*, *92*, 58–75.

Zha, L., Yin, Y., & Yang, H. (2016). Economic analysis of ride-sourcing markets. *Transportation Research Part C: Emerging Technologies*, *71*, 249–266.

Zhang, J., Liu, C., Li, X., Zhen, H.-L., Yuan, M., Li, Y., & Yan, J. (2023). A survey for solving mixed integer programming via machine learning. *Neurocomputing*, *519*, 205–217.

Zhou, T., Ren, J., Medo, M., & Zhang, Y. (2007). Bipartite network projection and personal recommendation. *Physical Review E*, *76*(4), 046115.

Zhou, Y., Yang, H., & Ke, J. (2022a). Price of competition and fragmentation in ridesourcing markets. *Transportation Research Part C: Emerging Technologies*, *143*, 103851.

Zhou, Y., Yang, H., Ke, J., Wang, H., & Li, X. (2022b). Competition and third-party platform-integration in ride-sourcing markets. *Transportation Research Part B: Methodological*, *159*, 76–103.

APPENDIX A: LIST OF NOTATIONS

Table 3. Notations

Notation	Description
m,n	Cardinalities of the two point sets in an one-dimensional RBMP
U, V	Two sets of points for each one-dimensional RBMP realization
I	Set containing all points in U and V
E	Set of edges connecting $u \in U$ and $v \in V$
y_{uv}	1 if $u \in U$ is matched to $v \in V$, 0 otherwise
$X_{m,n}$	Average optimal matching distance per point in an one-dimensional RBMP
x_i	x-coordinate of a point $i \in I$
z_i	Supply value of a point $i \in I$
l_i	Distance (step size) from a point $i \in I$ to its next point
l	Mean step size between any two consecutive points in I
$S(x;I')$	Net supply curve for any coordinate x and subset of points $I' \subseteq I$
$A(x;I')$	Total absolute area between curve $S(x;I')$ and x-axis from 0 to x
Y_n	Total absolute area between any balanced supply curve and x-axis from 0 to 1
$B(n)$	Expected total absolute area between the path of a random walk with 2n unit-length steps and x-axis
V'	An arbitrary set of unmatched/removed points in V
V^*	Optimal set of removed points
$\hat{V}$	Set of removed points from the proposed point removal process
$v'_k, v^*_k, \hat{v}_k$	k-th removed point along the x-axis in V', V^*, $\hat{V}$
m_k	Number of demand points in the k-th segment of the post-removal curve $S(x;I\setminus V^*)$
$\hat{m}_k$	Number of demand points in the k-th segment of the post-removal curve $S(x;I \setminus \hat{V})$
$\hat{m}_{k,0}$	Number of demand points with zero net supplies in the k-th segment of the postremoval curve $S(x;I \setminus \hat{V})$
$\hat{Z}_{k,a}$	Total absolute area enclosed by $S(x;I \setminus \hat{V})$ within $(x_{\hat{v}_k},1]$, which contains exactly a demand points
$Z_{m,n}$	Total absolute area enclosed by $S(x;I \setminus V^*)$ from 0 to 1
L	Length of a line (edge)
X_E	Average optimal matching distance per point on an arbitrary-length line
μ,λ	Densities of the two point sets on an arbitrary-length line
$\mathrm{G} = (\mathrm{V},\mathrm{E})$	Graph with node set V and edge set E
D	Degree of node in V
U_e,V_e	Realized point sets on an edge $e \in$ E
m_e,n_e	Cardinality of point sets U_e and V_e

Notation	Description
X_{G}	Average optimal matching distance per point in a graph
$X_{\mathrm{Gl}}, X_{\mathrm{Gg}}$	Average optimal local and global matching distances in a graph
α	Probability for global matching
$\phi(\cdot), \Phi(\cdot)$	Probability density function and cumulative distribution function of standard normal distribution
U_{e+}, V_{e+}	Remaining point sets on an edge $e \in$ E after local matching process
N_k	k-th layer of edges for a point $u \in U_e^+$ in breadth-first search
e_*	Edge containing the matched point v for a point $u \in U_e^+$
o_0	Nearer end of e to $u \in U_e^+$
o_k	Nearer end of e^* to o_0
d_1	Distance from $u \in U_e^+$ to o_0
d_2	Distance from o_0 to o_k
d_3	Distance from o_k to the matched point v of $u \in U_e^+$

APPENDIX B: DERIVATIONS IN CHAPTER 3

DERIVATION OF *P*(*X*)

Here we derive the probability *p*(*x*) for a swap between a new customer and a waiting customer to be feasible, conditional on the distance between the waiting customer and its assigned vehicle *X* to be *x*. Recall that a swap is feasible when conditions (a)-(c) are all satisfied, as illustrated by Figure 56a. Explicitly deriving a closed-form formula for *p*(*x*) is nontrivial, because the three conditions (a)-(c) may be correlated. For modeling convenience, we make the following assumptions: (i) the distance variables X, ${}^{c}\hat{X}$,Y, $\hat{Y}$ are independent of each other, (ii) condition (c) is (relatively) independent of conditions (a) and (b) because it involves only one random variable *X*ˆ and a given value *x*, and (iii) conditions (a) and (b) are correlated, because the same random variable *Y*ˆ needs to be simultaneously less than or equal to both variables *Y* and *X* = *x*. Under the Manhattan distance metric, condition (c) is satisfied if vehicle a_2 is located within the upper blue diamond area in Figure 56a. Because its location is uniformly distributed, condition (c) occurs with a probability of:

$$\Pr\{\hat{X} \le X | X = x\} = 1 - (1 - 2x^2)^{n_i}.$$

Figure 74. Equation. Probability of condition (c) occurs.

Similarly, condition (a) is satisfied when vehicle b_1 is located outside of the orange diamond area, and condition (b) is satisfied if customer B arrives within the lower blue diamond area (see Figure 56a). We assume the distributions of *Y* and *Y*ˆ are characterized by their cumulative distribution functions $F_Y(y)$, $F_{Y^\wedge}(y)$, and probability density functions $f_Y(y)$, $f_{Y^\wedge}(y)$, respectively. They satisfy the following:

$$\begin{aligned}
&F_Y(y) = 1 - (1 - 2y^2)^{n_i},\\
&F_{\hat{Y}}(y) = 2y^2,\\
&f_Y(y) = dF_y(y)/dy = 4n_i y(1 - 2y^2)^{n_i - 1},\\
&f_{\hat{Y}}(y) = dF_{\hat{y}}(y)/dy = 4y.
\end{aligned}$$

Figure 75. Equation. Cumulative distribution functions and probability density functions.

Then, we can derive the probability for conditions (a) and (b) to be satisfied simultaneously, conditional on the relative values of *X* and *Y* :

$$\begin{aligned}
\Pr\{\hat{Y} \le Y, \hat{Y} \le X | X = x\} &= \Pr\{\hat{Y} \le Y \le x\} + \Pr\{\hat{Y} \le x < Y\}\\
&= \int_0^x F_{\hat{Y}}(y) f_Y(y) dy + \int_x^{\sqrt{\frac{1}{2}}} F_{\hat{Y}}(x) f_Y(y) dy\\
&= \frac{1}{n_i + 1}\left[1 - (1 - 2x^2/R)^{n_i + 1}\right].
\end{aligned}$$

Figure 76. Equation. Probability of satisfying condition (a) and (b).

Putting the above together, we come up with:

$$p(x) = \Pr\{\hat{Y} \leq Y, \hat{Y} \leq X, \hat{X} \leq X | X = x\} \approx \Pr\{\hat{Y} \leq Y, \hat{Y} \leq X | X = x\} \cdot \Pr\{\hat{X} \leq X | X = x\}$$
$$= \frac{1}{n_i + 1} \left[1 - (1 - 2x^2)^{n_i+1}\right] \left[1 - (1 - 2x^2)^{n_i}\right].$$

Figure 77. Equation. Probability p(x).

DERIVATION FOR *ω*

In this appendix, we borrow the ideas in Ouyang and Yang (2023b) to present an analytical formula for the conditional probability of a feasible swap to be successful under competition, i.e. *ω*, when the new customer sees at least one feasible swap candidate.

First, a bipartite graph can be constructed by drawing connections between new customers and waiting customers during a sufficiently long time period in the steady state, as shown in Figure 74. Each time a new customer arrives, we mark it as a vertex (e.g., an orange hollow circle) on the upper side of the graph. This new customer immediately gets assigned a vehicle and starts to wait for pickup, so we add another corresponding vertex (e.g., a blue solid circle) on the lower side of the bipartite graph. Each waiting customer is expected to wait for a certain amount of time until pickup. During this waiting period, they may become a feasible swap candidate for some other customers who arrive later. The black dashed arrows in Figure 74 show the life-cycle of each customer, from arrival to waiting to pickup. Upon the arrival of each new customer, we check whether a swap is feasible between this new customer and all the currently waiting customers (i.e., those not yet picked up); if yes, we then add a solid-line edge to the graph connecting the corresponding vertices. For instance, in Figure 74, customers A, B and C arrive consecutively into the system. Upon customer C's arrival, customers A and B are still waiting for pickup. If A and B (as well as their vehicles) both satisfy the conditions to be feasible swap candidates for C, then we add edges |AC| and |BC| into the graph correspondingly. As such, the degree of a vertex on the upper side represents the number of all feasible swaps instantly seen by the corresponding new customer, denoted as a random variable *K*. Similarly, the degree of a vertex on the lower side is the number of feasible swaps the corresponding waiting customer may encounter during its entire waiting period, denoted by a random variable *F*. By definition, we have *μ* = E[*F*].

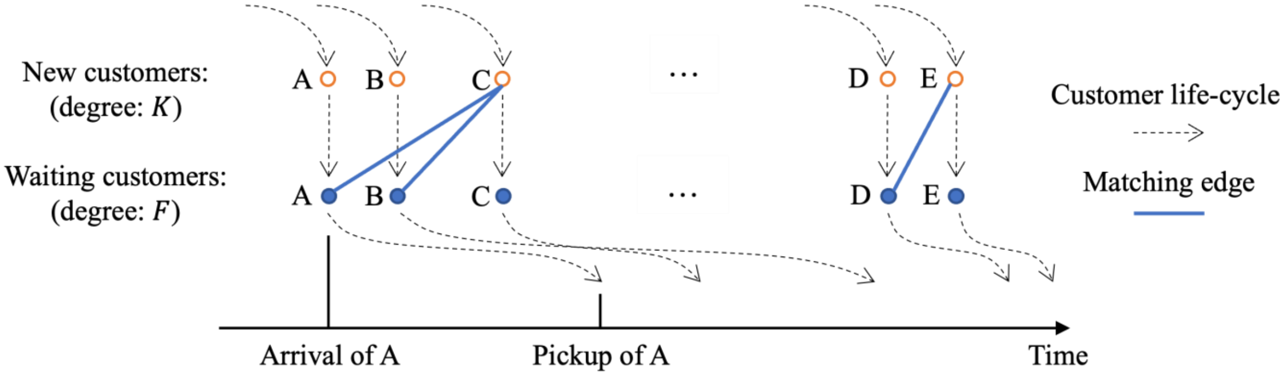


Figure 78. Graph. Bipartite graph between the new and waiting customers.

Because each edge contributes equally (i.e., exactly one degree) to a pair of vertices on both sides, the total numbers of degrees must be equal on both sides of the bipartite graph. Additionally, in the steady state, the new and waiting customer appear with the same arrival rate, thus the cumulative numbers of new customers and waiting customers (i.e., the numbers of vertices on both sides) in this sufficiently long time period must be the same, which equals λ multiplied by the length of the period. As such, the average degree per vertex must also be equal. Hence we shall have $\mu = E[K] = E[F]$. For modeling convenience, it is assumed that every new customer is probabilistically identical, and K approximately follows a Poisson distribution. Then, given the new customer sees at least one feasible swap candidates (i.e., $K \neq 0$), and each swap candidate has an equal probability of $1/K$ for being successful, the conditional probability of a feasible swap to be successfully conducted can be analytically expressed as follows:

$$\omega = \frac{\sum_{k=1}^{\infty} \frac{1}{k} \Pr\{K = k\}}{\Pr\{K \neq 0\}} = \frac{1}{e^{\mu} - 1} \left[\mathrm{Ei}(\mu) - \ln(\mu) - \gamma \right].$$

Figure 79. Equation. Equation of *ω*

This is the expression for ω.

A similar bipartite graph can be constructed to estimate the expected total number of successful swaps. The graph contains the same vertices as those in Figure 74, but only the edges corresponding to successfully swaps. As such, the degree of a new customer vertex is either 1 or 0, indicating whether or not an initial swap is experience. The degree of a waiting customer vertex now is the number of successful swaps it may encounter during its entire wait for pickup. Clearly, the total degrees on both sides are equal, and so are the number of vertices. Hence, the average degrees per vertex are also equal on both sides; i.e., the expected number of successful swaps per waiting customer must equal the probability for a new customer to have an initial swap, which is $1 - e^{-\mu}$. Note that, interestingly, this value will never exceed 1.

DERIVATIONS OF $F_Z(Z)$ AND $F_X(X)$

First, we show how to derive the cumulative distribution function of the initial pickup distance $F_Z(z)$. As illustrated in Figure 56a, conditional on whether an initial swap is conducted, the initial pickup distance equals to either $\hat{Y}$ or Y. As such, the probability for the initial pickup distance to be greater than a certain value z, $1 - F_Z(z)$, should equal $\Pr\{Y > z \cap \text{no initial swap}\} + \Pr\{Y > \hat{z} \cap \text{initial swap}\}$. Note too that $\Pr\{Y > z \cap \text{no initial swap}\} = 1 - \Pr\{Y \le z\} - \Pr\{Y > z \cap \text{initial swap}\}$. Hence,

$$\begin{aligned} F_Z(z) &= 1 - \Pr\{Y > z \cap \text{no initial swap}\} - \Pr\{\hat{Y} > z \cap \text{initial swap}\} \\ &= \Pr\{Y \le z\} + \Pr\{Y > z \cap \text{initial swap}\} - \Pr\{\hat{Y} > z \cap \text{initial swap}\}, \end{aligned}$$

Figure 80. Equation. Cumulative distribution function of the initial pickup distance.

where $\Pr\{Y \le z\}$ is simply given by $F_Y(z) = 1 - (1 - 2z^2)^{n_i}$.

Then, we need to further derive the probabilities of Pr{$Y > \hat{z}$ ∩initial swap} and Pr{$Y > z$ ∩ initial swap}. First of all, for an initial swap to be conducted, a feasible swap candidate should have already been chosen from the perspective of a new customer. As such, conditions (a)-(c) discussed in Section 3.3.2 (i.e., $\hat{Y} \leq Y$, $\hat{Y} \leq X$ and $\hat{X} \leq X$), conditional on $X = x$, should all be satisfied, while value of x could probabilistically take any value within its domain (0,$\sqrt{1/2}$]. Hence the two probabilities can be analytically derived, by integrating on all possible values of x, as follows:

$$
\begin{aligned}
&\Pr\{\hat{Y} > z \cap \text{initial swap}\} = \int_0^{\sqrt{\frac{1}{2}}} \Pr\{\hat{Y} > z, \hat{Y} \leq Y, \hat{Y} \leq X, \hat{X} \leq X | X = x\} f_X(x)dx \\
&\approx \int_z^{\sqrt{\frac{1}{2}}} \left(\Pr\{z < \hat{Y} < Y < x\} + \Pr\{z < \hat{Y} < x < Y\}\right) \Pr\{\hat{X} \leq x\} f_X(x)dx \\
&= \int_z^{\sqrt{\frac{1}{2}}} \left(\int_z^x \left[F_{\hat{Y}}(y) - F_{\hat{Y}}(z)\right] f_Y(y)dy + \int_x^{\sqrt{\frac{1}{2}}} \left[F_{\hat{Y}}(x) - F_{\hat{Y}}(z)\right] f_Y(y)dy\right) \Pr\{\hat{X} \leq x\} f_X(x)dx \\
&= \int_z^{\sqrt{\frac{1}{2}}} \frac{1}{n_i+1} \left[(1-2z^2)^{n_i+1} - (1-2x^2)^{n_i+1}\right] \left[1 - (1-2x^2)^{n_i}\right] f_X(x)dx,
\end{aligned}
$$

Figure 81. Equation. Probability of $\widehat{Y} > z$ and initial swap.

and

$$
\begin{aligned}
&\Pr\{Y > z \cap \text{initial swap}\} = \int_0^{\sqrt{\frac{1}{2}}} \Pr\{Y > z, \hat{Y} \leq Y, \hat{Y} \leq X, \hat{X} \leq X | X = x\} f_X(x)dx \\
&\approx \int_0^{\sqrt{\frac{1}{2}}} \Pr\{Y > z, \hat{Y} \leq Y, \hat{Y} \leq x\} \Pr\{\hat{X} \leq x\} f_X(x)dx \\
&= \int_0^z \Pr\{\hat{Y} \leq x, Y > z\} \Pr\{\hat{X} \leq x\} f_X(x)dx \\
&\quad + \int_z^{\sqrt{\frac{1}{2}}} (\Pr\{\hat{Y} \leq Y, z < Y < x\} + \Pr\{\hat{Y} \leq x, Y > x\}) \Pr\{\hat{X} \leq x\} f_X(x)dx \\
&= \int_0^z \int_z^{\sqrt{\frac{1}{2}}} F_{\hat{Y}}(x) f_Y(y)dy \Pr\{\hat{X} \leq x\} f_X(x)dx \\
&\quad + \int_z^{\sqrt{\frac{1}{2}}} \left[\int_z^x F_{\hat{Y}}(y) f_Y(y)dy + \int_x^{\sqrt{\frac{1}{2}}} F_{\hat{Y}}(x) f_Y(y)dy\right] \Pr\{\hat{X} \leq x\} f_X(x)dx \\
&= \int_0^z 2x^2(1-2z^2)^{n_i} \left[1 - (1-2x^2)^{n_i}\right] f_X(x)dx \\
&\quad + \int_z^{\sqrt{\frac{1}{2}}} \left[(1-2z^2)^{n_i} - \frac{(1-2x^2)^{n_i+1}}{n_i+1} - \frac{n_i(1-2z^2)^{n_i+1}}{n_i+1}\right] \left[1 - (1-2x^2)^{n_i}\right] f_X(x)dx
\end{aligned}
$$

Figure 82. Equation. Probability of $Y > z$ and initial swap.

Then, $F_Z(z)$ can be obtained by substituting Equations (18) and (19) into Equation (17); i.e.,

$$F_Z(z) = 1 - (1-2z^2)^{n_i} + \int_0^z 2x^2(1-2z^2)^{n_i}\left[1-(1-2x^2)^{n_i}\right] f_X(x)dx$$
$$+ \int_z^{\sqrt{\frac{1}{2}}} 2z^2(1-2z^2)^{n_i}\left[1-(1-2x^2)^{n_i}\right] f_X(x)dx.$$

Figure 83. Equation. Equation of $F_Z(z)$.

Note that the above derivation depends on the probability distribution of the remaining pickup distance, $f_X(x)$, observed at an arbitrary time. We next show the detailed derivations for $f_X(x)$. Because a swap could happen during a deadheading trip, the remaining distance *x* may experience abrupt jumps to a smaller value, and hence is not equally likely to be observed.

When *P*(*x*) is the conditional probability for a waiting customer to ever be *x* distance away from its assigned vehicle, then by definition of probability, $f_X(x)$ is proportional to *P*(*x*), as expressed in the following:

$$f_X(x) = \frac{P(x)}{\int_0^{\sqrt{\frac{1}{2}}} P(x)dx}$$

Figure 84. Equation. Equation of $f_X(x)$.

Then, we further derive *P*(*x*) from the perspective of a waiting customer. When the initial pickup distance *Z* is smaller than the value of *x*, which occurs with a probability of $\Pr\{Z < x\} = F_Z(x)$, then this customer will never see its vehicle *x* distance away. Otherwise, when $Z \geq x$, this customer will never see its vehicle *x* distance away only when the following conditions are satisfied: the waiting customer was $\hat{x} \in (x,z]$ distance away from its assigned vehicle, which occurs with probability $P(\hat{x})$; a swap is successful when the assigned vehicle is exactly $\hat{x}$ distance away, which occurs with a probability of $p(\hat{x})\omega$; the remaining pickup distance after the swap jumps to some value less than *x*, which occurs with a probability of $\frac{x^2}{\hat{x}^2}$. These conditions for $Z \geq x$ are illustrated in Figure 75, where the customer at the origin was initially assigned to a vehicle at distance *z*. When the distance reduces to $\hat{x}$, a swap occurs and a vehicle at a distance less than *x* takes over.

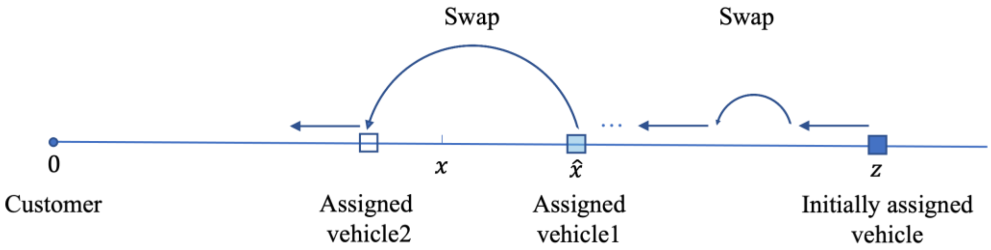


Figure 85. Graph. Illustration of distance x is skipped along the pickup trip.

Putting the probabilities for the conditions together, we know that *P*(*x*) satisfies the following equation:

$$P(x) = 1 - F_Z(x) - \int_{z=x}^{\sqrt{\frac{1}{2}}} \int_{\hat{x}=x}^{z} \frac{x^2}{\hat{x}^2} P(\hat{x}) p(\hat{x}) \omega d\hat{x} f_Z(z) dz.$$

Figure 86. Equation. Equation of P(x).