UNIVERSITY OF CALIFORNIA,
IRVINE

# Operational Benefits and Challenges of Shared-ride Mobility Services with Walking Trip Legs

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Civil Engineering

by

Zifan Wang

Thesis Committee:
Assistant Professor Michael Hyland, Chair
Associate Professor Douglas Houston
Professor R. Jayakrishnan
Distinguished Professor Will Recker

2021

# Table of Contents

# List of Figures

# List of Tables

# Acknowledgements

I owe my biggest thanks to Prof. Hyland. Mike has been very patient in listening the progress that I have made, the problems that I have encountered, and my ideas about the next step in the research process. He also has been very helpful in advising me when I have any concerns, and he gives a lot of good advice on improving my paper structure, graphics, citation etc.

It is also my great honor to have Prof. Houston, Prof. Jay, and Prof. Recker as my committee members. I have taken one or more classes related to transportation with them, and I really appreciate their teaching and mentoring.

I would like to thank Prof. Alan Murray. When I was an undergraduate student, Prof. Murray brought me to the world of optimization, and he gave me a chance to participate in research so that I gained skills and experiences useful in conducting my own thesis research.

I would like to thank Jin Xu, who encouraged me to continue my thesis research when I was frustrating with the progress.

I would like to thank four anonymous reviewers who provided feedbacks on a paper draft (which unfortunately was rejected) that was largely based on my class term paper; their feedbacks were valuable in guiding my thesis writing.

I would like to thank Younghun Bahk and Navjyoth Sarma JS. My discussion with them during meetings on an ongoing research project has been inspirational for me to endeavor beyond the techniques used in this thesis.

I would like to thank Priscilla Chu, Janelle Halog, and Siwei Hu. I have worked with them either on several class group projects or grading homework of undergraduate students. They also gave me advice along the path towards graduation.

I would like to thank Lin Xiao, the first person who told me the existence of Uber Express Pool.

Lastly, I would like to thank the staff, TAs, faculty, friends, and family who have supported my academic endeavor.

# Abstract of the Thesis

Operational Benefits and Challenges of Shared-ride Mobility Services with Walking Trip Legs

By

Zifan Wang

Master of Science in Civil Engineering

University of California, Irvine, 2021

Assistant Professor Michael Hyland, Chair

The possibilities of enabling travelers to walk a short distance to reach their prescribed pickup locations or final destinations in ridesharing services have received a lot of attention in recent years. This thesis classifies the existing literature on ridesharing with walking trip legs into three different classes. Additionally, this thesis presents a variant of the single-vehicle pickup and delivery problem with time windows (PDPTW), called the pickup and delivery problem with time windows and walking legs (PDPTWWL), that explicitly incorporates the access and egress walking trips of travelers into the decision space. The proposed model targets a ridesharing service that can pool together two to three requests, and it is solved to optimality using a commercial mixed integer programming (MIP) solver. The candidate pickup/drop-off (PUDO) locations for each request are either the road intersections or street centerline midpoints within a pre-defined maximum allowable walking time. The study applies the model to two road network datasets—Isla Vista and Chicago downtown—reflecting different real-world application scenarios, and it presents various performance metrics based on a large number of randomly generated 2-request, 3-request, and 4-requesst PDPTWWL instances. The results indicate that a relatively short allowable walking time has a comparatively high vehicle driving time reduction per second of walking (DTRPSW) relative to a long allowable walking time. Moreover, pooling a larger number of requests (e.g., four requests) in one

vehicle can lead to a much larger reduction in vehicle driving time and a higher DTRPSW, compared to pooling only two requests in one vehicle.

# 1 Introduction

Shared-ride mobility services typically involve pooling multiple travelers traveling in the same direction at the same time into one vehicle. Although the single tour that serves multiple requests in a shared-ride mobility service may take longer than the time required for non-shared, conventional taxi-like services, the vehicle travel time for serving multiple requests with one vehicle is expected to be lower than the vehicle travel time to serve each of the multiple requests in separate vehicles. Reducing system-wide vehicle-mile-traveled (VMT) or vehicle-hour-traveled (VHT) via shared-ride mobility services have significant human health and environmental implications such as a reduction in the vehicle-related air pollution (McCubbin and Delucchi, 1999; Zhang and Batterman, 2013) and energy consumption (Zhang et al., 2019; Zhou et al., 2016), as well as the potential to relieve traffic congestion (Alisoltani et al., 2021).

Several pooling systems in the academic literature or in the real world prescribe ways to pool travelers, including: carpooling, in which drivers (e.g. a company's employees) pickup riders (e.g. colleagues) while driving to/from a common destination (Hartman et al., 2014); vanpooling, in which a system manager assigns commuters and vans to park-and-ride locations (Kaan and Olinick, 2013); and general ridesharing services (Mourad et al., 2019). Additionally, Transportation Network Companies Uber and Lyft offer door-to-door shared-ride mobility-on-demand (SRMOD) services called UberPool and Lyft Shared (previously known as Lyft Line), respectively.

Even though those classic pooling systems may reduce VMT or VHT, the pickup and drop-off processes of such a door-to-door ridesharing scheme often entail vehicle detours that require extra VHT and VMT and thereby are quite inefficient. Of course, travelers generally prefer a door-to-door service that does not require walking. Nevertheless, some travelers may not mind walking and other travelers may be amenable to walking a short distance before being picked up or after being dropped off. The value of instructing travelers to walk to nearby locations for pickup and drop-off is that it can increase the efficiency of vehicle routes in a ridesharing service via reducing VHT and VMT. The reduction in VHT and VMT is likely to further reduce air pollution and energy consumption. In fact, the conventional public transport

system including bus and rail services can be considered as the simplest form of a non-door-to-door ridesharing service that involves walking trip legs. Yet, public transport typically is characterized by fixed schedules and routes, which are less flexible than SRMOD services, and public transport often provides poor service in low demand areas. Fortunately, some SRMOD service providers recently started experimenting with a walking-enabled ridesharing service, which offers a middle ground option between the door-to-door ridesharing service and the public transport service. Particularly, Uber has officially launched its ridesharing service that allows walking to nearby corners for a ride—known as Express Pool—around early 2018 (Lo and Morseman, 2018). In 2019, Lyft also rolled out a similar service known as Shared Saver.

Naturally, this thesis addresses two research questions: compared to a conventional door-to-door ridesharing service, what are the benefits and trade-offs of a ridesharing service that enables a short distance of walking? How do different ridesharing service designs impact the performance of a walking-enabled ridesharing service? This thesis addresses these two research questions by modeling and analyzing a deterministic, single vehicle ridesharing problem. The candidate pickup and drop-off (PUDO) locations for each ridesharing request are either the road junctions (Case Study I) or street centerline midpoints (Case Study II) within a pre-defined maximum allowable walking time. The performance of the ridesharing services is evaluated by a variety of metrics such as the vehicle driving time, the traveler walking time, the travel time needed for riders to reach destinations (or simply the total trip time), and the vehicle service time. The thesis examines how performance varies with the number of requests being pooled in one vehicle (2-request, 3-request, 4-request) and with the maximum allowable walking time per walking leg (zero to six minutes). The thesis also examines the effect of vehicle travel time from the vehicle's initial location to the first pickup location on the various performance metrics, and it evaluates the service performance of enabling walking only in the downtown area under morning commute and evening commute scenarios.

To prescribe the ridesharing vehicle route along with the PUDO locations for each request, this thesis extends the single-vehicle pickup and delivery problem with time windows (PDPTW) formulation to explicitly incorporate the access and egress walking trips of travelers into both the objective function and

time window constraints. The alternative PUDO locations of each request are modeled as vertices in the corresponding clusters, and the problem is to prescribe the fastest vehicle route that traverses each of those pickup and drop-off clusters subject to precedence constraints, time-window constraints, and vehicle capacity constraints.

This idea of traversing through clusters of vertices has its root in the generalized traveling salesman problem (GTSP). In the GTSP, a set of vertices are partitioned into clusters or subsets, and the goal is to find the minimum-cost path that visits exactly one vertex in each cluster (Laporte et al., 1987; Rice and Tsotras, 2013).

The proposed ILP in this thesis, called the pickup and delivery problem with time windows and walking legs (PDPTWWL), has a primary objective of minimizing the total vehicle driving time and a secondary objective of minimizing the amount of walking time of all requests. The secondary objective ensures the minimum walking path for each request to reach the shortest vehicle route determined by the primary objective.

This thesis presents two case studies. Case Study I—characterized by a small network—models the ridesharing trips going from a small community to a shopping mall. Case Study II—characterized by a north-south oriented corridor-like study area—models the ridesharing trips entering the downtown in the morning.

The remainder of the thesis is structured as follows. Section 2 reviews papers on ridesharing services that involve walking or flexible PUDO locations in general. Section 3 describes the conceptual model design, graph construction, and model formulation. Section 4 and Section 5 present two case studies applying the proposed model. Section 6 gives a summary and discussion of key findings of the two case studies. Section 7 concludes the study.

# 2 Background and Literature Review

The vehicle routing problem (VRP) and its many variants are an important class of problems in combinatorial optimization. In its simplest form, the VRP determines vehicle routes to visit a set of locations at the minimum cost (Toth and Vigo, 2002). In many applications, a location has to be visited before visiting another location (e.g. picking-up goods or travelers before dropping them off), so the vehicle routing problem can be extended to a pickup and delivery problem (PDP) (Sol and Savelsbergh, 1995). If the locations must be visited within specific time windows, the PDP can be extended to the PDP with time windows (PDPTW) (Desaulniers et al., 2002; Dumas et al., 1991). The dial-a-ride problem (DARP) is similar to the PDPTW, although its applications tend to involve transporting people rather than goods/freight. The DARP also typically imposes additional constraints on detours and waiting times of travelers (Cordeau and Laporte, 2007). Each of these classic problems involve door-to-door transport services. In other words, they do not model the possibility of allowing travelers to walk to pick-up locations or final destinations.

There are dozens of studies that utilize the concept of meeting points to allow walking to pickup locations and final destinations in ridesharing vehicle route design, in an attempt to reduce the vehicle travel distance or time and possibly increase the matching rate of the ridesharing system (Aissat and Oulamara, 2014; Czioska et al., 2019, 2017a, 2017b; Martínez et al., 2014; Miklas-Kalczynska and Kalczynski, 2020; Stiglic et al., 2015). Those studies emphasize the possibility to enable all or a subset of travelers sharing the same vehicle to walk from the requests' origins to one common pickup location and to walk from one common drop-off location to the destinations. The reduction in the number of intermediate vehicle stops is a key characteristic of those studies. In the strictest case, multiple travelers sharing a vehicle would be collectively picked up at one location and then collectively dropped off at another location; picking up multiple riders sequentially is not allowed. Section 2.1. reviews literature on meeting points with the strict assumption of only one intermediate pickup stop and one intermediate drop-off stop.

Other studies on meeting points relax this strict assumption. In those studies, one or more intermediate PUDO stops are allowed, and picking up multiple riders sequentially is possible. Those studies often have

artificial mechanisms that promote the sharing of a common PUDO location among multiple travelers. Section 2.2 reviews studies in this literature.

There are also several other studies that allow walking in ridesharing services without focusing on the reduction in the number of intermediate vehicle stops. The alternative PUDO locations are explicitly incorporated into the decision space. Assigning multiple travelers to a common location for picking up or dropping off is an optional and non-essential step; the occurrence of a location where multiple travelers are simultaneously picked up and/or dropped off is spontaneous and is the direct result of the attempt to reduce the VMT or VHT. Most of those studies share the features of the generalized traveling salesman problem (GTSP) or the shortest covering path problem (SCPP). The GTSP typically assumes a real-world location to correspond to multiple vertices on the graph, with each vertex belongs to one cluster or subset (Laporte et al., 1987; Rice and Tsotras, 2013). The SCPP typically assumes only one vertex for each real-world location, and the vertex may belong to different clusters (Current et al., 1984; Niblett and Church, 2014). Both the GTSP and the SCPP do not capture the precedence relation (i.e., picking up before dropping off), time windows, and vehicle capacities that are important considerations in a ridesharing service. However, some recent works (Balardino and Santos, 2016; Zheng et al., 2019) propose variants of either the SCPP or the GTSP formulation that consider one or more of those features in a static setting with demands known in advance. Furthermore, some recent works (Fielbaum et al., 2021; Li et al., 2020; Lyu et al., 2019) study the impacts of ridesharing services with walking trip legs in a dynamic simulation setting, in which demands are revealed over time. Section 2.3 reviews studies with walking legs but that do not explicitly require meeting points.

Lastly, a brief review on the key results of selected literature will be presented in Section 2.4.

## 2.1 Strict Meeting Points Approach: One Pick-up Stop and One Drop-off Stop Along the Way

Aissat and Oulamara (2014) study the selection of best meeting points with known rider-driver assignment. The study assumes that a ride-sharing driver has her own destination but can pick up and drop

off a rider along her route; however, the rider may need to travel to pickup and drop-off points that are not necessarily her origin or destination. Two heuristics and one enumerative solution approach are proposed.

Stiglic et al. (2015) demonstrate the system-level benefits of picking up and dropping off passengers at meeting points. The problem formulation assumes all the riders who are served by a vehicle have to share the same pickup location and drop-off location. The rider-driver assignment is determined by solving a maximum weight bipartite matching problem with side constraints.

Czioska et al. (2017a) propose a GIS-based workflow for identifying, assessing, and reducing the candidate meeting points throughout the study area in an intra-urban travel ride-sharing application. They try to ensure that all the candidate meeting points in the study area are safe and convenient, and they explore the effect of the meeting points' distribution and amount on the performance of the meeting point-based ride-sharing services. The bipartite graph is solved to assign riders to drivers in the simulation study.

Czioska et al. (2017b) propose a workflow for determining the best meeting point in a long-distance travel, single-vehicle ride-sharing application. The problem assumes that a driver who travels through a city is willing to pick up one or more riders in that city and later drop them off in another city along the way. The driver is willing to make only one stop in that city, and riders can use the public transit to arrive at somewhere within the walking range to a meeting point for picking up. The meeting point that entails the smallest maximum travel cost would be selected.

The strict assumption of requiring all riders to be collectively picked up and dropped off at the same location is ideal from a driver's perspective because the assumption requires fewer detours and fewer stops. However, if a vehicle picks up two riders at two different locations, the distance that the riders need to walk is likely to be reduced, entailing a better rider experience. Furthermore, a strict assumption about meeting points confines the process of picking up and dropping off passengers to relatively small regions. The robustness of such a ride-sharing service depends on whether the two regions have a high ride-sharing trip demands between them, and there could be some sharing potential missed by providing such a ride-sharing service.

## 2.2 Relaxed Meeting Points Approach: Multiple Pickup and Drop-off Stops Along the Way

Li et al. (2018) assume every request's origin location and destination location to be potential meeting points, and the vehicle can pick up and drop off riders at multiple meeting points. They propose an ILP formulation that is a combination of the vehicle routing problem and a location-allocation problem. The location-allocation portion of the formulation determines the meeting point locations and which demands are allocated to each location. Their model also explicitly captures the paths of both vehicles and travelers.

Martínez et al. (2014) incorporate meeting points into an express minibus service design problem. The methodology starts with clustering the potential users, and the results of the clustering are two spatial-temporal centroids—one for trip origin and one for trip destination—for each cluster. The next step is to try to aggregate two clusters of users to form cluster aggregates if possible, and this step distinguishes such a methodology with a strict meeting point approach. Finally, a third level of the aggregation—which concatenates the cluster aggregates into routes—is also performed using integer programming.

Miklas-Kalczynska and Kalczynski (2020) incorporate meeting points into carpooling, in which all the travelers are assumed to have the same destination. There are three different carpooling schemes being analyzed: door-to-door, strict meeting points, and a hybrid scheme in which the travelers sharing the same vehicle could be either all served by the door-to-door scheme or all served by the strict meeting points scheme. Such a hybrid scheme differentiates the study with the studies mentioned in Section 2.1., and the result shows that the additional flexibility associated with the hybrid scheme could slightly improve the system-level efficiency.

Czioska et al. (2019) propose a multi-step workflow for selecting the meeting points for each request as well as assigning a vehicle to serve each request. The workflow begins with a Euclidean distance-based clustering algorithm that produces clusters of requests with equal size in an attempt of ensuring the size of each cluster is sufficiently large. The requests in each cluster need to be further divided into as few subgroups as possible, and all the requests in a subgroup must be able to reach at least one common candidate meeting point each in the pickup and drop-off processes. Later, the clustering algorithm would

be re-run to group those subgroups of requests into "equally sized trip clusters with similar itineraries." A neighborhood search algorithm utilizing parallel computing can be performed on each trip cluster to determine (i) the trip-vehicle assignment, (ii) the route of each vehicle, and (iii) the meeting points selection. This vehicle routing heuristic algorithm appears to implicitly allow the existence of more than one pickup or drop-off location along the route.

While all studies that are reviewed in Section 2.1 and Section 2.2 so far are implemented in a static and deterministic environment, there are a few recent studies focusing on consolidating PUDO locations in a stochastic dynamic simulation environment. Gökay et al. (2019) designates a new request's PUDO location as the closest accepted request's PUDO location if such an assignment respects the specified location and time flexibility thresholds. Furthermore, they note that the initial requests' origins/destinations could have a profound influence on the future requests' PUDO locations. They thus apply Ordering Points to Identify the Clustering Structure (OPTICS) on historical request data, and the closest locations on the road network with respect to the clusters' weighted centers would be considered as candidate meeting points in addition to the accepted requests' PUDO locations. Overall, their study determines the PUDO location for a request first and then determine the best vehicle assignment along with the vehicle route.

Araldo et al. (2019) assume a grid network, define a stop spacing (between two adjacent candidate stops), and limit candidate PUDO locations to a subset of the road nodes—called admitted stop locations—using the pre-defined stop spacing. Each new request is assigned with an admitted stop location that is closest to its origin for picking up and an admitted stop location that is closest to its destination for dropping off. With the PUDO locations for a new request determined first, an insertion algorithm can be run for each new incoming request to determine the vehicle for serving the request and the new stop sequence plan.

Similar to Araldo et al. (2019) , Gurumurthy and Kockelman (2020) also limit the candidate PUDO locations to occur only at certain pre-defined locations, but they assume more realistic road network of Bloomington, Illinois. They utilize a hierarchical clustering algorithm to create the set of candidate PUDO locations using all possible origins and destinations in the study area, and a stop spacing parameter is also involved in this step.

Overall, the methods reviewed in Section 2.2. address the shortcoming of strict meeting points approach, namely, that the strict meeting points approach misses opportunities for sharing. In fact, such a relaxed meeting points approach sometimes is associated with demand consolidation or stop aggregation (Araldo et al., 2019; Gurumurthy and Kockelman, 2020).

## 2.3 Non-Meeting Points Approach: Reducing the Number of Intermediate Stops is Not the Key

Zhao et al. (2018) evaluate the benefits of allowing flexible PUDO locations without using a meeting-point approach. Instead, given a request's origin/destination location on a grid transportation network, the nodes that are topologically adjacent to the origin/destination location are candidate PUDO nodes. Such a set of nodes defines the pickup/drop-off space window, and travelers can only be picked up/dropped off at nodes within the space window. The problem is formulated using a space-time network representation, which involves discretizing the time horizon (such as 20-time steps). This formulation introduces a temporal dimension to the decision variables tracking traversed edges, and it introduces decision variables and constraints tracking whether a vehicle picks up/drops off a traveler at a space-time vertex. In contrast, the approach proposed in this thesis does not discretize the time horizon to better reflect the high temporal resolution in the real-world, does not rely on a grid network to make the model applicable to more general networks, and formulates a model that is more aligned with the conventional PDPTW formulation.

Balardino and Santos (2016) consider a multi-vehicle ridesharing problem that allows flexible pickup locations. The problem assumes that all ridesharing participants share the same destination, and a set of drivers may pick up riders at locations not necessarily the riders' origins. The pickup location for a rider must be within a pre-defined radius from the rider's origin, and the driver's traveling distance cannot exceed a pre-defined amount. The objective function minimizes the vehicle driving distances and maximizes the total number of served riders. An ILP program is formulated, and an iterated local search heuristic algorithm is proposed to solve the problem. The formulation considers vehicle capacities. Nevertheless, the pickup/drop-off time and the time window constraints are not considered.

Zheng et al. (2019) formulate an ILP that utilizes the idea of clusters to enable alternative PUDO locations in flex-route transit design, and they develop a memetic algorithm that combines the genetic algorithm with local search algorithm. They consider an un-capacitated problem that involves routing a vehicle thru a set of mandatory checkpoints: two terminal checkpoints and one intermediate checkpoint. The route of the vehicle is flexible such that route deviation and additional stops could be made to accommodate travelers whose origins and/or destinations are not at the checkpoints. The vehicle would dwell at a non-checkpoint stop with a fixed amount of time, but it could dwell at a checkpoint stop with a variable amount of time. The minimum dwelling time and the fixed departure time at checkpoints are given, and those are the only mandatory time constraints. Time windows for picking up and dropping off each request are not modeled.

Lyu et al. (2019) propose a taxi-sharing system that allows travelers to walk to/from road junctions that are within acceptable walking distances for picking up/dropping off. This system could be implemented in a dynamic and real-time setting. For a given new request and a waiting queue consisting of requests that have not yet found companions to share a taxi, the system would try to identify the companions in the waiting queue to share the ride with the given new request, the PUDO locations of each rider, and the shortest route for serving those riders. One important component in their proposed system is companion candidate search, which uses the waiting time constraint and the detour constraint to quickly narrow down the companion candidates in the waiting queue. Another key component in their proposed system—called scheduling algorithm—searches for valid subsets of candidate companions, solves the generalized traveling salesman problem using dynamic programming, assigns the closest empty vehicle to the first pickup location, checks for the constraints, and eventually returns the best sharing plan. Their results show that such a flexible taxi-sharing service outperforms both the door-to-door service and the strict meeting point service.

Li et al. (2020) introduce the alternative PUDO locations to the public vehicle service problem, which is an on-demand ridesharing service with large (20 seats or more) vehicle capacities and flexible routes. Unlike Lyu et al. (2019) who assume that the route and schedule of a shared vehicle would be fixed after

the riders-vehicle assignment is made, Li et al. (2020) propose an algorithm that inserts each new request into the route of one vehicle in the fleet. Additionally, the problem that is considered by Li et al. (2020) enforces that the traveler must arrive at the pickup location by the time when the vehicle arrives, and such an assumption does not appear in Lyu et al. (2019). Li et al. (2020) propose a concept called border points, which are the locations forming the border of a traveler's potential walking area. There are two disjoint situations being identified. First, the vehicle does not need to enter the walking area to pick up or drop off a given traveler, and the traveler needs to walk to/from one of the border points. Second, the optimal PUDO location is inside the walking area, and the vehicle would pass by one border point for entering the walking area. Based on this idea, the authors develop a three-stage algorithm. The first stage inserts the new request into a vehicle's route stop sequence by only testing the border points. The second stage determines the new route segments through the best border point and then selects the optimal PUDO location on the new route segments. The last stage tries to merge the PUDO locations of two or more requests if a common candidate location exists. A check is also performed on whether the pickup location would enable traveler to arrive before the vehicle arrives. If such a requirement is not satisfied, a pickup location that is closer to the request origin and satisfies the requirement would be re-selected. The algorithm is tested using the network and taxi data of Shanghai. The authors compare the proposed service with the fixed-route bus service, and they compare the service performance in the light traffic with the heavy traffic condition.

Fielbaum et al. (2021) extend the dynamic ridesharing algorithm proposed by Alonso-Mora et al. (2017) to consider the possibility of walking. The core of Alonso-Mora et al. (2017) is solving an assignment problem that aims to assign each trip (defined as a group of requests) to a vehicle if possible. This assignment problem requires identifying a set of feasible trip-vehicle pairs, and the identification of trip-vehicle pairs really involves repeatedly solving single-vehicle DARPs that determine the best stop sequence plans. To incorporate walking into the decision space, Fielbaum et al. (2021) not only need to determine the stop sequence plan but also the PUDO locations for each request in a trip-vehicle pair. The authors utilize an insertion algorithm with several speed-up heuristics: they terminate the search for the optimal insertion positions as soon as the total cost does not decrease, they only evaluate the PUDO

locations adjacent to a PUDO location that decreases the total cost, and they discard feasible trip-vehicle pairs that entail a comparatively high cost (thus no need to further evaluate trips that are supersets of this discarded trip).

A non-meeting points approach tends to be flexible in assigning PUDO locations for a traveler. In a relaxed meeting points approach, the traveler tends to be assigned to a cluster centroid or the closest candidate PUDO location. In a non-meeting points approach, the optimal PUDO locations for a traveler may depend on the interaction among the traveler's origin/destination location, the current vehicle route (in a dynamic setting), and other unassigned travelers' origins/destinations. Such a complex interaction brings a high-level flexibility in PUDO location determination, but it also makes the problem much harder to tackle from a computation perspective.

## 2.4 Key Results from the Literature Review

The existing literature generally shows that enabling walking can reduce vehicle travel distance or travel time. This finding is consistent regardless of whether the studies assume a static (Czioska et al., 2019; Stiglic et al., 2015) or dynamic (Fielbaum et al., 2021; Gurumurthy and Kockelman, 2020; Li et al., 2020; Lyu et al., 2019) setting, and it holds even if part or all of the optimization framework does not focus on minimizing vehicle travel distance or time (Czioska et al., 2019; Fielbaum et al., 2021; Lyu et al., 2019). Nevertheless, the amount of reduction in vehicle distance or time traveled as a result of introducing walking trip legs compared to a door-to-door service can vary depending on methods and datasets. Stiglic et al. 2015 report a 2.2% system mileage savings assuming four meeting points per traffic analysis zone (TAZ) in Austin, Texas. Czioska et al. 2019 report that the total VKMT reduces from 110,655 to 79,470 (28.2%) and the total VHT reduces from 3,208 to 2,133 (33.5%) assuming the highest demand scenario in Braunschweig, Germany. Li et al. 2020 report a 21,707 km (33.8%) reduction in system total vehicle travel distance and a 581 hours (39.6%) reduction in system total vehicle driving time assuming a 500-meter maximum allowable walking distance per walking trip leg and a heavy traffic scenario in inner ring road of Shanghai, China.

Enabling walking can lead to an increase in the matching rate (or a decrease in rejection rate) compared to a door-to-door ridesharing service. Stiglic et al. 2015 report a 6.8% increase in the system matching rate assuming four meeting points per TAZ. Araldo et al. 2019 report that the number of rejected requests decrease from slightly more than 24,000 to slightly more than 12,000 assuming a 560-meter stop spacing and 320 requests/hour/km^2. Fielbaum et al. 2021 report that the percent of rejected requests decrease from 4.44% to 0.78% assuming a fleet of 3,000 vehicles with a capacity of 6. Lo and Morseman 2018 report that the Uber Express Pool product has a 3.6% higher matching rate compared with the Uber Pool product.

One important service quality measure would be the total time needed for a traveler to reach her destination (total trip time), which may include walking, curbside waiting, and in-vehicle travel time. The existing literature does not have a consistent conclusion on whether enabling walking can reduce an average rider's total trip time compared to a door-to-door service, although most studies that report traveler's total trip time do show an increase in total trip time as walking is introduced (Araldo et al., 2019; Czioska et al., 2019; Gurumurthy and Kockelman, 2020; Li et al., 2020). Fielbaum et al. 2021 also show that the total trip time (which they report as the average delay) may increase if walking is enabled for toy (hypothetical) networks in three out of four scenarios. On the other hand, Fielbaum et al. 2021 show that the total trip time may decrease if walking is enabled for their Manhattan case study. There are four conditions in Fielbaum et al. 2021 that appear to jointly result in a reduction in total trip time: (i) a method that belongs to a non-meeting points approach to provide adequate flexibility in selecting optimal PUDO locations; (ii) an objective function that gives sufficiently large weights on walking and curbside waiting time throughout the optimization procedure; (iii) a constraint that ensures all travelers have enough time to walk towards their pickup location before the vehicle arrives; and (iv) a scenario in which both origins and destinations of requests tend to be concentrated rather than dispersed throughout the study area. Li et al. 2020 also demonstrate that the total trip time can be reduced if only the pickup walking trip leg is enabled while the drop-off walking trip leg is disabled, even if the objective function only focuses on minimizing VHT.

# 3 Model

## 3.1 List of Notations

### Indices:

| | | |
|---|---|---|
| $i, j, v, w$ | : | Vertex index |
| $k, k'$ | : | Cluster index |
| $n$ | : | Total number of requests |
| $HI$ | : | The index of the last vertex (the vehicle's final destination) added on the graph |
| $P$ | : | The array of indices of pickup clusters (i.e., clusters with indices $\{1,\ldots,n\}$) |
| $D$ | : | The array of indices of drop-off clusters (i.e., clusters with indices $\{n+1,\ldots,2n\}$) |
| $S_k$ | : | The set of vertices belonging to the cluster $k$ |
| $IN(j)$ | : | The set of vertices that are directly connected to the vertex $j$ via the inbound edges ending at $j$ |
| $OUT(i)$ | : | The set of vertices that are directly connected from the vertex $i$ via the outbound edges starting at $i$ |
| $V$ | : | The array of indices of all the vertices on the graph; $\{0, \ldots, HI\}$ |
| $N$ | : | The array of indices of all the clusters; $\{0, \ldots, 2n+1\}$ |
| $A_{k'}$ | : | The set of indices of clusters whose vertices are directly connected with vertices in cluster $k'$ via the outbound edges starting from vertices in cluster $k'$ |

### Model Inputs

| | | |
|---|---|---|
| $t_{ij}$ | : | The vehicle driving time for traversing through the edge $(i, j)$ |
| $t_j'$ | : | The walking time between the vertex $j$ and its corresponding cluster's center (request's origin/destination)[1] |
| $RT_k$ | : | The request time of the request $k$ (if a request is made, the traveler is assumed to be ready for walking or picking up) |
| $t_k^{direct}$ | : | The direct (without ridesharing) travel time from request $k$'s origin to destination |
| $l_k$ | : | The number of travelers to be picked up/dropped off at the cluster $k$. If $k$ is a pickup cluster, the number is positive. If $k$ is a drop-off cluster, the number is negative |

### Model Parameters:

| | | |
|---|---|---|
| $\alpha$ | : | The discount factor (weight) in front of the walking time objective term (which is the secondary objective) |
| $\theta$ | : | The walking time threshold used in the graph construction step; the maximum allowable walking time per leg |
| $MWT$ | : | The maximum waiting time (the pickup walking trip leg is assumed to occur within the waiting time) |
| $d$ | : | The time for traveler getting on or getting off the vehicle |

[1] The walking time is computed by the distance divided by an assumed walking speed, which is 3 feet per second. The assumed walking speed is much less than the average walking speed because the service should be reliable for most (if not all) people instead of an "average" person.

| | | |
|---|---|---|
| *MDT* | : | The amount of time that enables a less restrictive latest arrival time constraint, in addition to the maximum waiting time plus the direct (without ridesharing) travel time |
| *MaxWalkTime* | : | The maximum allowable walking time of both the first walking leg (for picking up) and the last walking leg (for dropping off) combined |
| *C* | : | The maximum number of riders that a vehicle could accommodate at the same time (vehicle capacity) |
| *M* | : | A sufficiently large number |

**Decision Variables:**

| | | |
|---|---|---|
| $X_{ij}$ | : | If the edge (*i, j*) is traversed by the vehicle, 1; otherwise, 0 |
| $T_k$ | : | The continuous decision variables tracking the service time. If *k* is a pickup cluster, it is the time at which the vehicle starts serving request *k*. If *k* is a drop-off cluster, it is the time at which the vehicle ends serving request *k-n* |
| $L_k$ | : | The continuous decision variables tracking the number of passengers in the vehicle upon leaving cluster *k* |

## 3.2 Concept of Operations

In contrast to the classic PDPTW or DARP, travelers' pickup locations and drop-off locations are endogenous in the PDPTWWL. Subject to maximum walking time constraints, the ride-sharing system determines each traveler's PUDO locations, implying that each traveler may need to walk from her origin to the pickup location for the first leg of her trips and/or walk from her drop-off location to her destination for the last leg.

The single-vehicle PDPTWWL is set up as a single-vehicle routing problem, with a pre-specified set of demands (i.e., travelers who want to travel from one place to another) that should be satisfied by the vehicle, unless the time-related constraints forbid a feasible solution. The single vehicle must be empty when it begins its route. A dummy depot (an extra vertex on the graph) that is not meant to correspond to a real-world location may serve as the vehicle's final destination that the vehicle heads to after it serves its final request (Case Study I and Case Study II), and another dummy depot corresponding to the vehicle's initial location may also be introduced (Case Study II). Every vertex connects to the dummy depot with a travel time edge value equal to zero. Presumably, each traveler walks to her pickup location immediately after her request time and walks to her final destination immediately after the vehicle drops her off.

In the PDPTW or DARP, each pickup vertex (the origin of the traveler) is visited by one and only one vehicle, and the drop-off vertex must be visited by the same vehicle. However, the PDPTWWL utilizes the concept of clusters to encapsulate alternative PUDO locations into the model formulation. A cluster consists

of vertices corresponding to locations that are within the walking range of the request's origin or destination. Each request's origin or destination has a cluster of vertices, and a vertex only belongs to one cluster. Note that the same real-world location (e.g., road intersection) can correspond to multiple vertices each in their own cluster. If the vehicle traverses through any vertex within the cluster, then the cluster can be considered as served (the traveler is either picked up or dropped off).

The walking trip legs need to be properly captured by the model formulation. The walking time should be properly accounted in both the objective function and the constraints tracking the time when a traveler can be picked up by a vehicle. In this proposed model, if any of the inbound edges connecting to vertex *j* is traversed by the vehicle, the walking time associated with the vertex *j* is captured by the summation shown in the left of Figure 1. As shown in Figure 1, a request's origin or destination and other locations within the walking time threshold relative to it are abstracted as vertices, and those vertices (blue square and green circles) form a cluster. The walking time between the request's origin/destination vertex (referred as the cluster's center) and a vertex with an index *j* in the same cluster is denoted as $t'_j$. If a directed edge starting at vertex *i* in another cluster (yellow circle) and ending at the vertex *j* is traversed by the vehicle (i.e., $X_{ij}$=1), the walking time (either in picking up or dropping off) of the request would be $t'_j X_{ij}$. There is a set of inbound edges connecting to vertex *j*, so traversing through any of those edges would take the walking time associated with vertex *j* into account. An outer summation that sums all the vertices in the cluster needs to be added so that the walking time associated with the cluster can be captured, regardless of which location is selected to pick up or drop off a request. Figure 1 illustrates the idea by summing the inbound edges but summing the outbound edges can be an alternative option. Nevertheless, a nested summation that sums all the vertices in a cluster and all the inbound or outbound edges of each vertex would require special attention when coding the model in a mathematical programming language[2].

[2] A conventional summation method such as sum or quicksum (in Gurobipy) would be inefficient in constructing the expressions that involve such nested summations and different index sets. I recommend the use of the most efficient expression constructor available in a mathematical programming language, such as Gurobi's LinExpr method. Otherwise, a substantial amount of the script runtime would spend on setting up the model instead of actually solving the model.

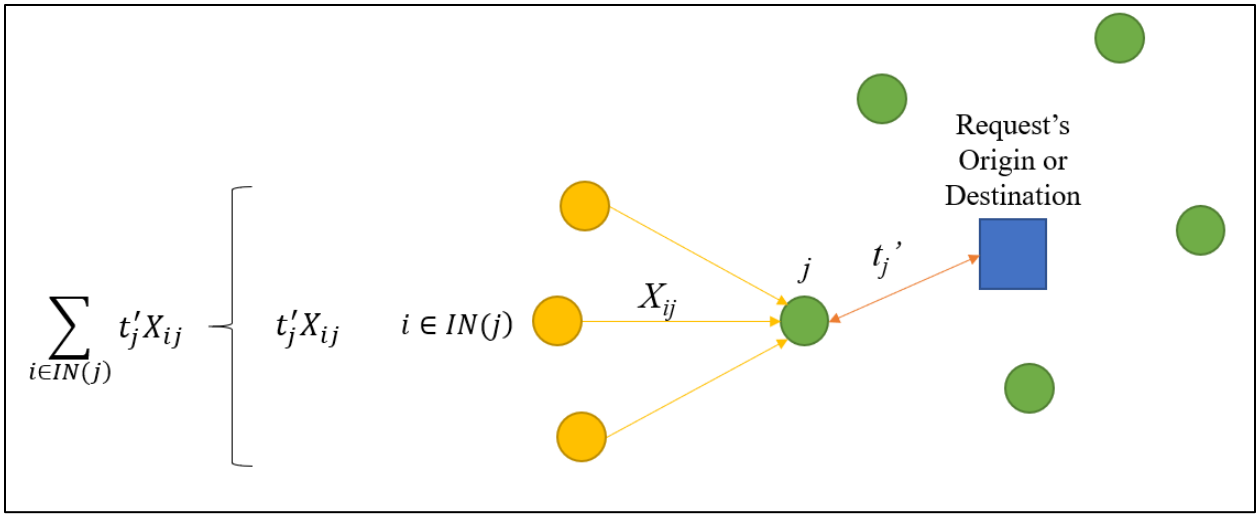


**Figure 1 Walking time calculation and diagram**

The model objective function has two components that reflect a hierarchy of goals. The primary goal is to avoid vehicle detours in the process of picking up or dropping off travelers; the vehicle detour time can be reflected in the total vehicle driving time (not including the time that the vehicle waits for travelers). Therefore, the primary term of the objective function involves minimizing the total vehicle driving time. However, this component alone is likely to prescribe PUDO locations that lead to unnecessary walking. For example, the vehicle route indicated by the $X_{ij}$variables may pass by a real-world road node that is within the walking range and takes less walking time from a request's origin compared with the pickup location prescribed by the $X_{ij}$ variables. In other words, the graph vertex traversed by the vehicle is not guaranteed to correspond to the most efficient pickup location on the vehicle route, and a candidate pickup location that is on the vehicle route and entails a shorter walking time for the traveler may simply not be selected in the model solution. Therefore, the objective function includes a second component that minimizes the combined walking time of all legs and all requests. This term ensures there is no unnecessary walking for travelers. In other words, it aims to identify the minimum walking path for each request to reach the shortest vehicle route that is determined by the primary objective. To reflect the primary and secondary objectives of driving time and walking time, respectively, the walking time objective term is multiplied by the discount factor $\alpha$.

The PDPTWWL formulation includes maximum walking time constraints, time-window constraints, precedence constraints (picking up before dropping off), and vehicle capacity constraints.

## 3.3 Graph Construction

In this study, a graph consisting of edges and vertices is created. The initial vehicle's location is vertex 0. For each request's origin or destination, a cluster of vertices is created on the graph. The total number of requests is denoted as *n*. The vertices representing requests' origin locations are indexed from 1 to *n*; whereas, the vertices representing the destination locations of requests are indexed from *n+1* to *2n*. If a road intersection (Case Study I) or a road centerline midpoint (Case Study II) can be reached within a walking time threshold, $\theta$, from a specific request's origin or destination, a vertex is created, and the vertex is considered a member of the cluster corresponding to the request's origin or destination. More than one vertex is created for a road node if the road node is within $\theta$ from multiple requests' origins or destinations. For example, if a road node is within the six minute-walking time from both request 1 and 2's request origin locations, two vertices need to be created—a vertex for request 1 (a member of cluster 1) and a vertex for request 2 (a member of cluster 2). In this situation, the same real-world location is represented as two separate vertices (though the travel time between the two vertices is zero). For each vertex in a cluster, the walking time ($t_j'$) between the cluster's center (representing either the request's origin or destination location) and the real-world location that corresponds to the vertex needs to be stored as an attribute of the vertex. The vertex representing the dummy depot (or the vehicle's final destination) is added last, with an index of *HI*.

Clusters also have indices. The vertex representing the vehicle's initial location forms a cluster with index 0. The pickup clusters are indexed from 1 to *n*, and the drop-off clusters are indexed from *n+1* to 2*n*. Request 1 corresponds to cluster 1 and cluster *n+1*. The dummy depot vertex forms a cluster with index *2n+1*. This study uses *P* to denote the array of pickup cluster indices ({1,…,n}) and *D* to denote the array of drop-off cluster indices ({n+1,…,2n}).

There are six situations in which directed graph edges need to be created for the vertices between a pair of clusters: (1) the initial vehicle location cluster to a pickup cluster; (2) a pickup cluster to another pickup cluster besides itself; (3) a pickup cluster to a drop-off cluster; (4) a drop-off cluster to another drop-off cluster besides itself; (5) a drop-off cluster to a pickup cluster besides the pickup cluster corresponding to it (same request); (6) a drop-off cluster to the dummy depot cluster. In other words, for a pickup cluster, all its vertices need to have outbound edges connected to all the vertices in drop-off clusters as well as vertices in pickup clusters besides itself. For a drop-off cluster, all its vertices need to have outbound edges connected to all the vertices in drop-off clusters besides itself, in pickup clusters besides the one corresponding to the same request, and in the dummy depot cluster. Two vertices belonging to the same cluster would not have edges between them. To capture the clusters (identifiable by their indices) that are directly connected with a given cluster $k'$(i.e., directed edges exist from the vertices in cluster $k'$ to the vertices in cluster *k*), I introduce the sets $A_{k'}$. The graph edges have vehicle driving time as weights.

## 3.4 Mathematical Formulation

$$Min \sum_{i \in V} \sum_{j \in OUT(i)} t_{ij} X_{ij} + \sum_{j \in V} \sum_{i \in IN(j)} \alpha t'_j X_{ij} \tag{1}$$

$$\sum_{i \in S_k} \sum_{j \in OUT(i)} X_{ij} = 1 \quad \forall\, k \in P \tag{2}$$

$$\sum_{j \in S_k} \sum_{i \in IN(j)} X_{ij} = 1 \quad \forall\, k \in D \tag{3}$$

$$\sum_{j \in OUT(0)} X_{0j} = 1 \tag{4}$$

$$\sum_{i \in IN(HI)} X_{i,HI} = 1 \tag{5}$$

$$\sum_{j \in IN(i)} X_{ji} - \sum_{j \in OUT(i)} X_{ij} = 0 \quad \forall\, i \in V \backslash \{0, HI\} \tag{6}$$

$$\sum_{j \in S_k} \sum_{i \in IN(j)} X_{ij} - \sum_{i \in S_k} \sum_{j \in OUT(i)} X_{ij} = 0 \quad \forall\, k \in P \cup D \tag{7}$$

$$-M + \sum_{i \in S_{k'}} \sum_{j \in S_k} M X_{ij} + T_{k'} + d + \sum_{i \in S_{k'}} \sum_{j \in S_k} t_{ij} X_{ij} - T_k \leq 0 \quad \forall\ k' \in N, k \in A_{k'} \tag{8}$$

$$T_k + d \leq T_{k+n} \quad \forall\, k \in P \tag{9}$$

$$T_k \geq RT_k + \sum_{j \in S_k} \sum_{i \in IN(j)} t'_j X_{ij} \quad \forall\, k \in P \tag{10}$$

$$RT_k + MWT \geq T_k \quad \forall\, k \in P \tag{11}$$

$$\sum_{i \in S_k} \sum_{j \in OUT(i)} t'_i X_{ij} + \sum_{w \in S_{k+n}} \sum_{v \in IN(w)} t'_w X_{vw} \leq MaxWalkTime \quad \forall\, k \in P \tag{12}$$

$$\sum_{w \in S_{k+n}} \sum_{v \in IN(w)} t'_w X_{vw} + d + T_{k+n} \leq RT_k + MWT + t_k^{direct} + 2d + MDT \quad \forall\, k \in P \tag{13}$$

$$-M + \sum_{i \in S_{k'}} \sum_{j \in S_k} MX_{ij} + L_{k'} + l_k - L_k \leq 0 \quad \forall\, k' \in N, k \in A_{k'} \tag{14}$$

$$M - \sum_{i \in S_{k'}} \sum_{j \in S_k} MX_{ij} + L_{k'} + l_k - L_k \geq 0 \quad \forall\, k' \in N, k \in A_{k'} \tag{15}$$

$$L_k \leq C \quad \forall\, k \in P \tag{16}$$

$$L_k \geq l_k \quad \forall\, k \in P \tag{17}$$

$$L_{k+n} \leq C - l_k \quad \forall\, k \in P \tag{18}$$

$$X_{ij} = \{0,1\} \quad \forall\, i \in V, j \in OUT(i) \tag{19}$$

$$T_k \geq 0 \quad \forall\, k \in N \tag{20}$$

$$L_k \geq 0 \ \forall\, k \in N \tag{21}$$

The primary objective in objective function (1) minimizes the vehicle driving time (which encapsulates detours) while the secondary objective minimizes customer walking time.

The constraints in (2) ensure that every request must be picked up. For each pickup cluster, the sum of all its vertices' outbound edges needs to be one. Similarly, the constraints in (3) ensure that every request must be dropped off. For each drop-off cluster, the sum of all its vertices' inbound edges needs to be one.

Constraints (4), (5), and (6) are the flow conservation constraints. Constraint (4) states that the vehicle route must start at the initial location. The sum of the outbound edges from the initial location must be one. Constraint (5) states that the vehicle route must end at the final vehicle destination (e.g., dummy depot) vertex whose index is *HI;* i.e., the highest index number among vertices. The sum of the inbound edges into that vertex must be one. The constraints in (6) ensure that for all other vertices on the graph, the inflow equals outflow.

The flow conservation constraints in (6) are designed at the vertex level. Additionally, the total flow entering a cluster must equal the total flow exiting the same cluster. The constraints in (7) are the flow conservation constraints for clusters. This set of constraints, by trimming the integer solution space, tends to decrease computation time.

The constraints in (8) track the time that the vehicle may serve each cluster. If the vehicle traverses through any of the edges going from cluster *k'* to cluster *k* (Figure 2), the time in which the cluster *k* may be served must be no less than the sum of the following three terms: (i) the time in which the cluster *k'* is served ($T_{k'}$), (ii) the time for a traveler to get into (or out of) the vehicle (*d*), and (iii) the vehicle travel time ($t_{ij}$) between vertex $i$ and $j$ if that specific edge is traversed. Those constraints do not aim to enforce equality since a vehicle may stay at one place in waiting for the pickup time window to come or waiting for the traveler to arrive at the pickup location. Additionally, when *k'* is zero (initial vehicle location cluster), $T_0$ needs to be replaced with a known value (e.g., the time when the vehicle will be ready to serve the next set of travelers). The constraints in (8) also act as the subtour elimination constraints in this model, and a detailed discussion on subtour elimination constraints will be presented in Section 3.7.1. It should be noted that the selection of the parameter value *M*—a sufficiently large number—requires special attention, and a discussion on the value of *M* will also be presented in Section 3.7.2.

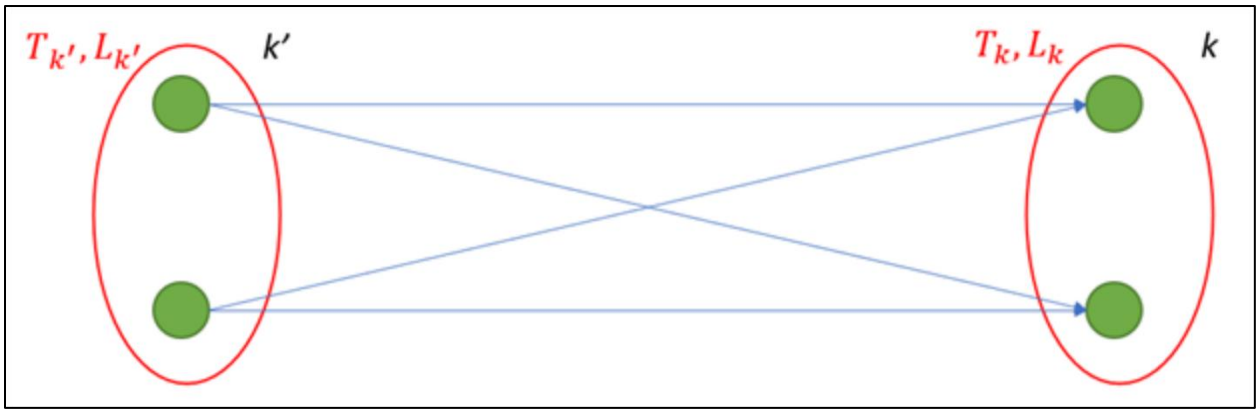


**Figure 2 Constraints (8), (14), (15) are applied when a cluster is connected to another cluster via edges**

The constraints in (9) mandate that a request must be picked up before it is dropped off. It is possible to add a term denoting the minimum travel time between the pickup cluster $k$ and its corresponding drop-off cluster $k + n$ to the left-hand side of Constraints (9) so that the drop-off time windows can be tighter. However, adding this extra minimum travel time term does not bring obvious reductions in the computational time given the case studies' setups and the solution approach (commercial MIP solver) used in this thesis research.

The constraints in (10) ensure that a request can only be picked up after its request time and after the traveler walks to the pickup location (assuming the walk trip starts at the request time). The traveler's walk time ($t_j'$) is accounted by the sum of all inbound edges to every vertex in a pickup cluster.

The constraints in (11) mandate that a request must be picked up before their maximum waiting time is exceeded. The waiting time in this formulation implicitly includes the walking time from the traveler's origin to her pickup location.

The constraints in (12) impose an upper limit on the total walking time, which is the sum of origin to pickup location walking leg and drop-off location to destination walking leg. Each pickup cluster and its corresponding drop-off cluster (they are related to the same request) form a constraint. The maximum walking time per leg constraint is already captured during the graph construction process (i.e., only locations within $\theta$ are considered in the cluster).

The constraints in (13) are the latest arrival time constraints for the travelers. The left-hand side of the constraints track the latest arrival time of each request and consist of three terms: the walking time of the drop-off location to destination walking leg, the time for the traveler to get out of the vehicle, and the time in which the vehicle arrives at the drop-off location. The right-hand side of the constraints define the upper limit on the latest arrival time of each request and consist of five terms: the request time, the maximum time that a traveler is willing to wait before being picked up, the direct vehicle travel time from the request's origin to destination, the time for getting in and out the vehicle, and the additional amount of delay time (as a result of sharing the ride with others and possible walking) that is tolerable by the travelers.

The constraints in (14) and (15) track the number of passengers in the vehicle upon leaving each cluster. Those constraints enforce strict equality, and $l_k$ could be positive or negative depending on whether $k$ is a pickup or drop-off cluster. When $k$ is $2n+1$ (the final vehicle destination or dummy depot cluster), the term $l_k$ could be omitted in the constraints or set equal to zero (no passenger will be getting on/off the vehicle). When $k'$ is 0 (the vehicle's initial location cluster), the term $L_{k'}$ could be omitted in the constraints or set equal to zero (the vehicle initially has no passenger). Constraints (14) and (15) may also act as subtour elimination constraints in some special circumstances, which I will discuss in Section 3.7.1.

The constraints in (16) enforce the capacity of the vehicle. The constraints in (17) define the lower bound of the vehicle occupancy upon leaving a pickup cluster. The constraints in (18) define the upper bound of the vehicle occupancy upon leaving a drop-off cluster.

The constraints in (19), (20), and (21) are the binary or non-negativity constraints of the decision variables.

## 3.5 Pre-processing

An important issue that arises when introducing walking trip legs into the ridesharing vehicle route is that a trip might have the potential of being accommodated solely by walking, and an explicit discussion on this issue is lacking in the existing literature. For example, given that a traveler is willing to walk a maximum of six minutes from her origin to the pickup location and another six minutes from the drop-off location to her destination, the trip could be completed solely by walking if the entire trip requires no more than twelve minutes of walking. In an extreme case involving such a trip and its associated request, a vehicle may visit the request's pickup cluster and then immediately visit the drop-off cluster, and the vertices being visited may correspond to the same real-world location. That is, the vehicle is forced to pick up and drop off the request, but the traveler does not actually travel along with the vehicle and eventually walks from the "pickup" location to her destination. Theoretically, this extreme case could lead to higher travel time for passengers already onboard who would need to wait for the traveler to get on and off the vehicle and possibly making unnecessary detours. The traveler who makes such a request might have to wait for the

vehicle to arrive at the pickup location before continuing walking to her destination, and such waiting obviously does not make sense.

Nevertheless, such a trip that can be completed solely by walking sometimes could benefit from using a walking-enabled ridesharing service. For example, the traveler's origin and destination are already located on the vehicle route, and the vehicle may arrive at the traveler's origin in a few minutes. In such an extreme case, serving this traveler would relieve the burden of walking for this traveler with negligible impact on the experience of riders already onboard. To my best knowledge, studies on meeting points or flexible PUDO locations do not explicitly discuss this issue and simply simulate demands that have origins and destinations sufficiently apart. In this thesis, Case Study I does not have any such a request being generated, but Case Study II has a few instances involving such a request being generated in the initial simulation step. Those instances are effectively treated as invalid and removed from the analysis. Although I have not proposed any alternative way to treat such requests, I think it is important to raise awareness of this issue.

## 3.6 Post-processing

After solving the ILP, it is important to post-process the results to obtain corrected service times. Basically, the service time-related constraints (8)-(13) do not guarantee that a pick-up occurs immediately when both the vehicle and traveler arrive at the pickup location, nor do they guarantee a drop-off occurs at the time when the vehicle arrives at the request's drop-off location. Post-processing of the $T_k$ decision variables is needed to overcome those two shortcomings of the ILP formulation. Post-processing should ensure that the pickup/drop-off times are as "compact" as possible, which is essential for a proper analysis of request waiting times and in-vehicle times. This thesis also analyzes the time for a traveler to reach her destination, which is defined as the difference between the time when the traveler arrives at her destination and the time when the traveler makes a request. The time to reach destination has four components: (i) the time before the request is picked up (walking time to the pickup location and waiting time at the pickup location), (ii) the in-vehicle travel time (from start boarding the vehicle to start getting-off the vehicle), (iii)

the time for a traveler to get off the vehicle, and (iv) the time to walk from the drop-off location to destination.

## 3.7 Additional Considerations

This section provides theoretical discussion on the subtour elimination constraints, the value of *M*, and the formulation of in-vehicle travel time constraints that limit the time a traveler may be onboard the vehicle.

### *3.7.1 Elimination of Subtours*

The classic subtour elimination constraints were first proposed by Dantzig et al. (1954) in a formulation of the traveling salesman problem. Such constraints can be written as follow:

$$\sum_{i \in Q} \sum_{j \in OUT_i \cap Q} X_{ij} \leq |Q| - 1 \quad \forall\, Q \subset V \;\&\; 2 \leq |Q| \leq \frac{|V|}{2} \tag{22}$$

$Q$ denotes a subset of vertices on the graph, and $|Q|$ gives the number of vertices in the subset $Q$. Any subset of vertices on a graph has the potential to form a subtour (i.e., a loop that is disconnected with the rest of the route). Constraints (22) requires enumerating all possible subsets that have a size within the given range. It is unnecessary to enumerate subsets that have a size greater than one half of the total number of vertices on the graph. For example, on a graph with six vertices, if we eliminate all the subtours involving two vertices, subtours that involve four vertices would be eliminated simultaneously since the occurrence of a subtour involving four vertices would create another subtour involving two vertices. If $|V|$ is an even number, it is also unnecessary to enumerate all subsets with a size equal to $|V|/2$, and only half of such combinations is sufficient. In the example of a graph with six vertices, if we create a subtour elimination constraint involving the first three vertices, the subtour involving the last three vertices would be eliminated simultaneously without adding an explicit constraint.

The constraints in (22) can be extended to problems involving alternative stops such as the problem addressed in this thesis:

$$\sum_{k' \in Q} \sum_{i \in S_{k'}} \sum_{k \in A_{k'} \cap Q} \sum_{j \in S_k} X_{ij} \leq |Q| - 1 \quad \forall\, Q \subset P \cup D \;\&\; 2 \leq |Q| \leq \frac{|P \cup D|}{2} \tag{23}$$

*Q* now denotes a subset of clusters instead of vertices. Formulation that is similar to Constraints (23) has appeared in the integer programming formulation of the GTSP (Laporte et al., 1987). The Table 1 shows the number of Constraints (23) that might be needed in the single-vehicle PDPTWWL problem.

**Table 1 Dantzig-Fulkerson-Johnson Subtour Elimination Constraints in the Context of Alternative Stops**

| # of Requests | # of Pickup Clusters | # of Drop-off Clusters | # of PUDO Clusters | # of Subtour Elimination Constraints |
|---|---|---|---|---|
| **2** | 2 | 2 | 4 | $0.5*{}_4C_2$ = 3 |
| **3** | 3 | 3 | 6 | ${}_6C_2+0.5*{}_6C_3$ = 25 |
| **4** | 4 | 4 | 8 | ${}_8C_2+{}_8C_3+0.5*{}_8C_4$ = 119 |

Adding those explicit subtour elimination constraints to the formulation presented in Section 3.4. could dramatically reduce the computational time in problem instances involving four requests (see Section 5.4.). However, as the number of requests involved in a problem increases, the number of such dedicated subtour elimination constraints increases quickly. One heuristic way to address the subtour problem is to use only those clusters that constitute a subtour appearing in the solution of a relaxed formulation (without enumerating all possible subtours) to form a subtour elimination constraint and re-solve the model with the newly added constraint.

Another well-known way of formulating subtour elimination constraints is first presented by Miller et al. (1960). Their formulation requires introducing continuous decision variables $u_i$ for each vertex *i*. Using the index scheme described in Section 3.3. (without introducing alternative stops), Miller-Tucker-Zemlin formulation of subtour elimination constraints can be written as follow:

$$u_0 = 1 \tag{24}$$

$$2 \leq u_i \leq |V| \quad \forall\, i \in \{1, \dots, HI\} \tag{25}$$

$$u_i + 1 - |V|\left(1 - X_{ij}\right) - u_j \leq 0 \quad \forall\, i \in V, j \in OUT_i \backslash \{0\} \tag{26}$$

The value of $u_i$ after the model is solved effectively represents the position of vertex *i* in the tour. Constraint (24) sets the vertex corresponding to the initial location as the position 1 in the tour. Constraints

(25) reflect the requirement that any other vertex would be visited as position 2 or up to the last ($|V|$ which is the total number of vertices) position in the tour.

If the edge starting at vertex *i* and ending at vertex *j* is traversed, the position of vertex *j* in the tour must be one position higher compared with the position of vertex *i* in the tour. This requirement is partly mandated by Constraints (26). Constraints (26) do not enforce equality, but their interactions with Constraints (25) would ensure that the proper track of position in the tour. Yet, Constraints (26) alone would be sufficient to produce a strictly increasing pattern of the positions of vertices in the tour, and such a strictly increasing pattern of the positions in the tour is the reason why such constraints could prevent the occurrence of subtours.

The Miller-Tucker-Zemlin formulation of subtour elimination constraints can be extended to problems involving alternative stops such the problem that is addressed by the current paper. This would require introducing continues decision variables $u'_k$ for each cluster *k*.

$$u'_0 = 1 \tag{27}$$

$$2 \leq u'_k \leq |N| \quad \forall\, k \in \{1, \dots, 2n+1\} \tag{28}$$

$$u'_{k'} + 1 - |N|\left(1 - \sum_{i \in S_{k'}} \sum_{j \in S_k} X_{ij}\right) - u'_k \leq 0 \quad \forall\, k' \in N, k \in A_{k'} \backslash \{0\} \tag{29}$$

The revised Miller-Tucker-Zemlin formulation is suitable in the context of alternative stops, and the number of constraints is more tractable compared with Constraints (23), as approximately ${}_{|N|}C_2$ additional constraints are needed.

As briefly mentioned in Section 3.4., Constraints (8) also act like subtour elimination constraints, and its capability of eliminating subtours comes from a similar fashion as Constraints (29). Given that parameter *d* is set to a positive number, the term $d + \sum_{i \in S_{k'}} \sum_{j \in S_k} t_{ij} X_{ij}$ in Constraints (8) should lead to an increasing pattern of the values of the decision variables $T_k$ as the vehicle moves from one cluster to the next cluster, just like the plus 1 term in the left-hand-side of Constraints (29). However, if the vehicle driving time from one cluster to the next cluster is zero (i.e. the two stops at the clusters correspond to the same real world

location) and the time to getting on/off the vehicle ($d$) is zero, the term $d + \sum_{i \in S_{k'}} \sum_{j \in S_k} t_{ij} X_{ij}$ would be equal to zero rather than a positive number. In such a case, the service time at the preceding cluster ($T_{k'}$) could be the same as the service time at the following cluster ($T_k$) without violating Constraints (8). A subtour consisting of the two vertices—each in one of the two clusters—could form. Therefore, if the parameter value $d$ is set to zero in the proposed model, Constraints (8) may fail to prevent subtours, and Constraints (23) will be necessary to ensure subtours always get eliminated; otherwise, Constraints (8) with a properly set $M$ value will be sufficient to eliminate all possible subtours.

Constraints (14) and (15) sometimes can also act like subtour elimination constraints. For example, if a vehicle moves from a pickup cluster $k'$ to another pickup cluster $k$, $L_k$ must be equal to $L_{k'} + l_k$ where $l_k$ is positive according to Constraints (14) and (15). A subtour involving the two vertices—each in one of the two pickup clusters—would require $L_k$ be equal to $L_{k'} + l_k$ while $L_{k'}$ be equal to $L_k + l_{k'}$. Those equalities could not be satisfied simultaneously given that for all clusters $k$ involving in the subtour have all positive (or conversely, negative) $l_k$. Subtours involving only two pickup clusters are eventually eliminated by Constraints (14) and (15). In a similar fashion, subtours involving more than two pickup clusters (no drop-off cluster) are also prohibited, and subtours involving two or more drop-off clusters (no pickup cluster)—where both $l_{k'}$ and $l_k$ are given as negative—are eliminated. However, if a subtour involves both pickup and drop-off clusters, Constraints (14) and (15) do not always prevent the occurrence of the subtour. In a simple example of a subtour involving only one pickup and one drop-off clusters, the subtour cannot be eliminated by Constraints (14) and (15) if the number of travelers being picked up at the pickup cluster equals to the number of travelers being dropped off at the drop-off cluster, while the subtour can be eliminated if the number of travelers being picked up at the pickup cluster does not equal to the number of travelers being dropped off at the drop-off cluster. The takeaway is that Constraints (14) and (15) have some subtour elimination capabilities, but those constraints are insufficient to eliminate all subtours.

### *3.7.2 Selection of the Large Number M*

Constraints (8), (14), and (15) involve the use of a parameter denoted as *M*, which is a generic term symbolizing a sufficiently large number. The large number *M* acts as a component of translating a binary condition into a linear constraint. If the vehicle traverses through any edge connecting one cluster with the other, one positive *M* and one negative *M* terms would cancel out. If the vehicle does not traverse through any edge connecting one cluster with the other, the positive or negative *M* (depending on the sign of the constraints) should be large enough to not violate the constraint(s) regardless of the values for the continuous decision variables in the constraints. It might be convenient to arbitrarily choose a really large number as the value of *M*, but a too large number might cause a constraint that is supposed to be violated becomes un-violated. Basically, mixed-integer program (MIP) solver may result in fractional values for binary decision variables because the solver decides on whether to make a cut on the binary decision variables based on the solver setting known as the integer feasibility tolerance (or integrality tolerance). If the fractional value of a binary decision variable falls within the integer feasibility tolerance, the solver would not make a cut on that decision variable. The product between a too large *M* and a fractional binary decision variable may make an impact on whether the constraint is satisfied or not. Considering the following example of an extreme case in which a simplified version of Constraints (8) fails to eliminate a subtour due to the use of a too large value of *M*.

$$-M\left(1 - X_{46,115}\right) + T_2 + t_{46,115} + d - T_4 \leq 0$$

$$-M\left(1 - X_{115,46}\right) + T_4 + t_{115,46} + d - T_2 \leq 0$$

Let $M$=10,000,000, $t_{115,46}$=$t_{46,115}$=7, $d$=10, $X_{46,115}$=1, $X_{115,46}$=0.9999980, $T_2$=118, the first inequality would require that $T_4$ to be no less than 135. Practically, the second inequality requires that $T_2$ to be no less than $T_4 + t_{115,46} + d$, and such an inequality certainly cannot hold true given that the first inequality is true. Numerically, $-M\left(1 - X_{115,46}\right) = -20$ and effectively makes the second inequality true while the first inequality is true. In contrast, assuming a smaller number of *M* such as 10,000 would make $-M\left(1 - X_{115,46}\right)$ equal to -0.02 and have negligible impact on the feasibility of the constraint.

A too large value of *M* may have some other problems such as leading to weak relaxations. In fact, rather than saying that *M* is a large number, *M* perhaps is better to be described as an adequately large number that is as small as possible. In this thesis, *M* is set as 7,200 in Constraints (8) since the total time for serving all the travelers is not expected to exceed 7,200 seconds (2 hours). *M* is equal to the maximum vehicle capacity in Constraints (14) and (15).

### *3.7.3 Maximum Detour Constraints*

An important aspect of a ridesharing service's quality is the amount of detour time experienced by each traveler. One strategy of ensuring service quality is to impose the latest arrival time constraints. The latest arrival time that is tolerable by a traveler should be no less than the direct vehicle travel time from the traveler's origin to destination, and there should be some extra waiting time that enables the vehicle to arrive at the traveler's origin. In the context of ridesharing, even more extra time that enables the vehicle to serve other travelers along the way can also be introduced to extend the latest allowable arrival time. Nevertheless, if the extra waiting time is long enough, it is possible for a vehicle to pick up a traveler early but make a lot of detours in serving other travelers without violating the traveler's latest arrival time constraints. The user experience of the traveler is deteriorated. Hence, the latest arrival time constraints along might be insufficient to ensure the service quality.

In the conventional door-to-door ridesharing services, it is common to add constraints that restrict the absolute amount of detour time for each traveler and/or impose a detour limit based on the ratio of the detour time to the direct travel time. Those maximum detour constraints have two major roles. The first role is to ensure that a traveler would not spend an excessive amount of time traveling on the route that mainly services other travelers. In this case, a more direct, non-detour route is preferred, and the actual amount of time that a traveler spends in the vehicle matters the most. The second role is to give more assurance on the time when a traveler could reach her destination, and this second role can be considered as a complement to the latest arrival time constraints. In this case, the travel time of the entire journey matters the most. The distinction in the emphasis is blurred in the conventional door-to-door ridesharing

services, besides the fact that the more constraints there are the better assurance on the service qualities. In contrast, a model that incorporates walking trip legs should make a clear distinction on those two roles. While the latest arrival time constraints already provide some assurance on the travel time of the entire journey, the assurance on the amount of time that the vehicle may make detours to pick up or drop off other travelers could be modeled by in-vehicle travel time constraints.

In-vehicle travel time constraints should correctly track the actual travel time of each traveler from the pickup location to the drop-off location, and the additional flexibility in the amount of detour—either defined as an absolute amount or a ratio—could be based on the direct travel time from the pickup location to the drop-off location. Alternatively, the additional flexibility of in-vehicle detour could be defined in terms of the direct travel time from the origin to destination. Although the second alternative is computationally simpler, the comparison between the actual in-vehicle travel time and the theoretical maximum allowable in-vehicle travel time is not on the same basis. A tradeoff between the computational efficiency and the theoretical soundness of the comparison might need to be made. Nevertheless, regardless of which alternative is chosen, those constraints ensure the amount of time the travelers would spend in the vehicle does not exceed a maximum amount, so their vehicle trip would not make various turns and detours just for picking up or dropping off other passengers.

This thesis presents a way of formulating in-vehicle travel time constraints that define the additional detour flexibility based on the direct travel time from the pickup location to the drop-off location.

$$(1 + IVDR)t_{iw} + M\left(1 - \sum_{j \in OUT(i)} X_{ij}\right) + M\left(1 - \sum_{v \in IN(w)} X_{vw}\right) \geq T_{k+n} - (T_k + d) \quad (30)$$

$$\forall\, k \in P, i \in S_k, w \in S_{k+n}$$

For any pair of vertices that one vertex belongs to a pickup cluster and the other vertex belongs to the corresponding drop-off cluster, a constraint can be added. Here $t_{iw}$ denotes the direct vehicle travel time between the pair of vertices, and *IVDR* is a user-defined ratio. The in-vehicle travel time for any passenger

must be no more than $1 + IVDR$ times the shortest travel time between the request's pickup and drop-off location.

Constraints (30) can be aggregated by summing all the vertices in the drop-off cluster or alternatively summing all the vertices in the pickup cluster to reduce the number of constraints.

$$\sum_{w \in S_{k+n}} \sum_{v \in IN(w)} (1 + IVDR) t_{iw} X_{vw} + M \left( 1 - \sum_{j \in OUT(i)} X_{ij} \right) \geq T_{k+n} - (T_k + d) \quad (31)$$

$$\forall\, k \in P, i \in S_k$$

$$\sum_{i \in S_{k'-n}} \sum_{j \in OUT(i)} (1 + IVDR) t_{iw} X_{ij} + M \left( 1 - \sum_{v \in IN(w)} X_{vw} \right) \geq T_{k'} - (T_{k'-n} + d) \quad (32)$$

$$\forall\, k' \in D, w \in S_{k'}$$

Although adding Constraints (31) and (32) to the proposed model can significantly reduce the computational time compared with adding Constraints (30) to the proposed model, the model after adding such in-vehicle travel time constraints is only practically solvable in two-request cases. Therefore, those constraints are not implemented in Case Study I and Case Study II.

# 4 Case Study I: Isla Vista

## 4.1 Data and Parameter Settings

Isla Vista (lower portion of Figure 3) is a small community adjacent to University of California, Santa Barbara. Its land use is characterized by a mix of residential use and commercial use. A shopping mall is located northwest of Isla Vista—seen in the top-left portion of Figure 3.

The roads in Figure 3 are classified into three facility types: major, local, and destination road. The driving speeds are 51, 22, and 10 feet per second on the major, local, and destination roads, respectively. The walking speed is 3 feet per second regardless of facility type. Both driving travel time and walking travel time are computed for all pairs of road nodes in the network.

The next section includes illustrative examples consisting of two, three, and four requests. All requests are assumed to be made at the same time (time 0), and each request includes only one traveler. To provide further insights and to verify that the findings in the illustrative examples hold in general, the study randomly generates fifty sets of 2-request instances, fifty sets of 3-request instances, and fifty sets of 4-request instances. For each set of requests, the origin locations of requests are randomly sampled among the network nodes in Isla Vista. The initial vehicle location is the same as the illustrative examples, and the request destinations also remain unchanged compared with the illustrative examples. The experimental results based on those 150 sets of randomly generated request instances are shown in Section 4.3., and the computational complexity results are shown in Section 4.4.

**Table 2 Model Parameter Settings for Case Study I**

| Parameters | Values |
|---|---|
| $\alpha$ | 0.001 |
| *MWT* | 600 seconds (10 minutes) |
| *d* | 10 seconds |
| *MDT* | 600 seconds (10 minutes) |
| *MaxWalkTime* | $2\theta$ |
| *C* | 5 |
| *Relative MIP Gap* | 1e-7 |
| *Integer Feasibility Tolerance* | 1e-7 |

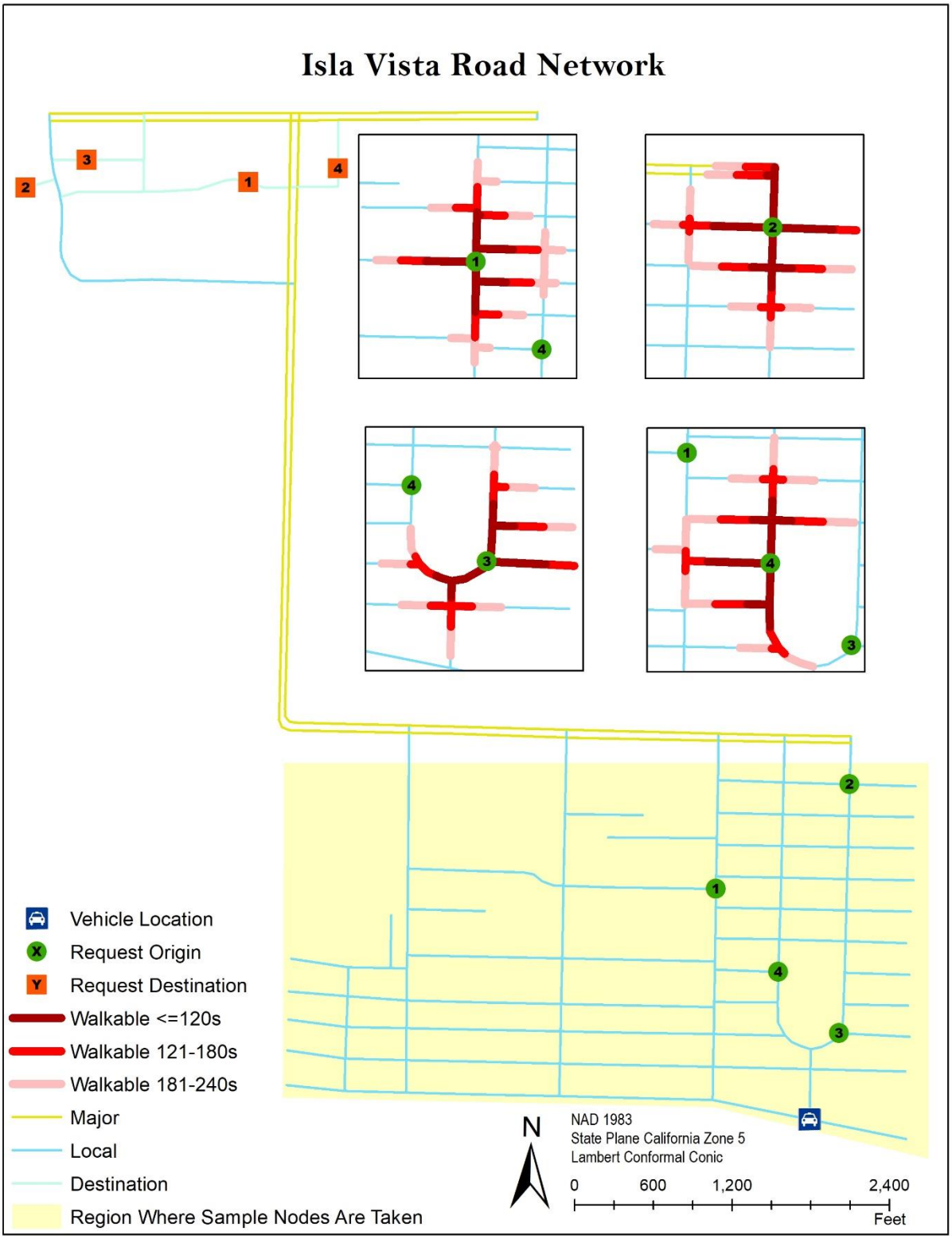


**Figure 3 Map of Isla Vista Road Network**

## 4.2 Illustrative Examples

Figures 4 and 5 display the results of the illustrative examples. Figure 4 displays the pickup results in the residential portion of the network; whereas, Figure 5 displays the drop-off results in the shopping district. The three rows in both Figure 4 and Figure 5 display the cases with two, three, and four requests, respectively. The columns in Figures 4 and 5 illustrate how the vehicle routes and walking legs change as $\theta$ changes. Table 3 contains summary model output statistics for the illustrative examples.

Figures 4 and 5 indicates and Table 3 confirms that increasing $\theta$ can significantly decrease the vehicle route time. Compared with the maps in the first and second column, the maps in the last column ($\theta$ is 240 seconds) clearly show that having requests walk for a few minutes can transform optimal vehicle routes from circuitous to direct routes that resemble a fixed bus route service.

The last column of Figure 5 indicates that with 2, 3, and 4 requests, the vehicle serves each set of requests with the same vehicle route and thus the same vehicle driving time. Hence, the vehicle route efficiency benefits of having requests walk a short distance seem to increase as the number of requests, the capacity of vehicles, and the service design allow more sharing.

However, looking at the last column of Figure 5 and particularly the last column of Table 2, it appears that the reduction in vehicle driving time is actually more than offset by the increase in total walking time across the requests. For example, in the 2-request case with $\theta = 240$ seconds, the reduction in vehicle driving time is 196 seconds while the increase in the system walking time is 718 seconds. Ultimately, this may translate to an increase in the traveler's time needed to reach his/her final destination (total trip time).

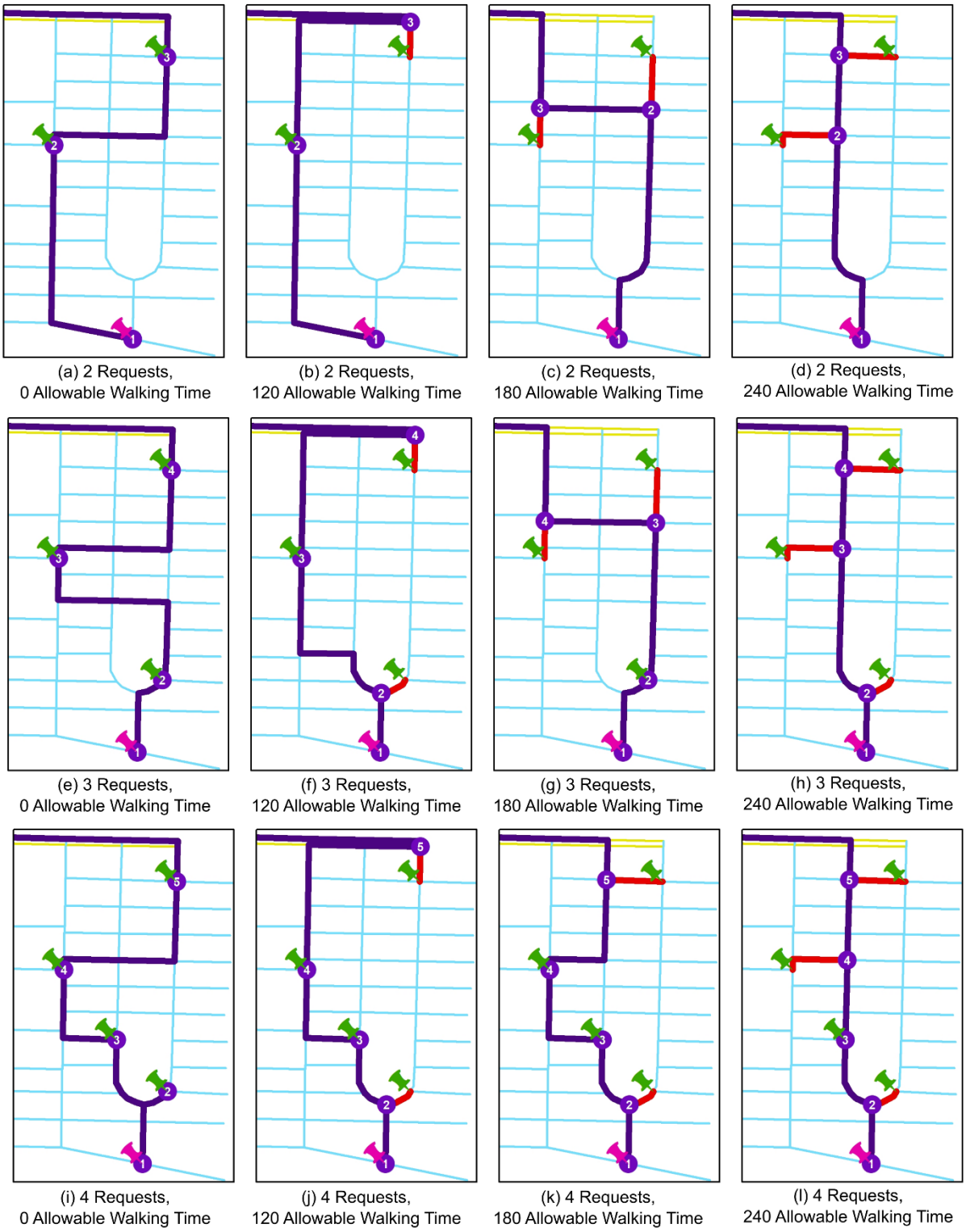


**Figure 4 Vehicle routes (purple) and walking legs (red) illustrative example results for request origin locations (green marker)**

**Table 3 Key Performance Measures of the Illustrative Examples**

| | | Maximum Allowable Walking Time, $\theta$, (s) | | | |
|---|---|---|---|---|---|
| | | **0** | **120** | **180** | **240** |
| **Number of Requests** | **2** | D: 540<br>W: 0/0<br>F: [436, 580]<br>C: 0.0019953 | D: 451 (-89)<br>W: 105/185<br>F: [485, 570] (+39)<br>C: 0.0069799 | D: 394 (-146)<br>W: 273/253<br>F: [543, 621] (+148)<br>C: 0.0149603 | D: 344 (-196)<br>W: 365/353<br>F: [555, 719] (+258)<br>C: 0.0209425 |
| | **3** | D: 618<br>W: 0/0<br>F: [610, 470, 678]<br>C: 0.0060198 | D: 452 (-166)<br>W: 192/274<br>F: [647, 559, 654] (+102)<br>C: 0.0249333 | D: 394 (-224)<br>W: 273/342<br>F: [582, 543, 631] (-2)<br>C: 0.0428848 | D: 344 (-274)<br>W: 452/542<br>F: [680, 555, 729] (+206)<br>C: 0.0877671 |
| | **4** | D: 709<br>W: 0/0<br>F: [721, 488, 581, 789]<br>C: 0.0109413 | D: 542 (-167)<br>W: 192/274<br>F: [757, 548, 569, 764] (+59)<br>C: 0.0309184 | D: 401(-308)<br>W: 255/492<br>F: [623, 599, 584, 672] (-101)<br>C: 0.2234051 | D: 344 (-365)<br>W: 452/692<br>F: [690, 590, 555, 739] (-5)<br>C: 0.2363663 |

D: Vehicle driving time (reduction in vehicle driving time compared with no walking allowed)
W: Walking time of all requests' pickup legs/Walking time of all requests' drop-off legs
F: [Request 1's total trip time, Request 2's total trip time, …] (change in the sum of total trip time of all requests compared with no walking allowed)
C: Computational time

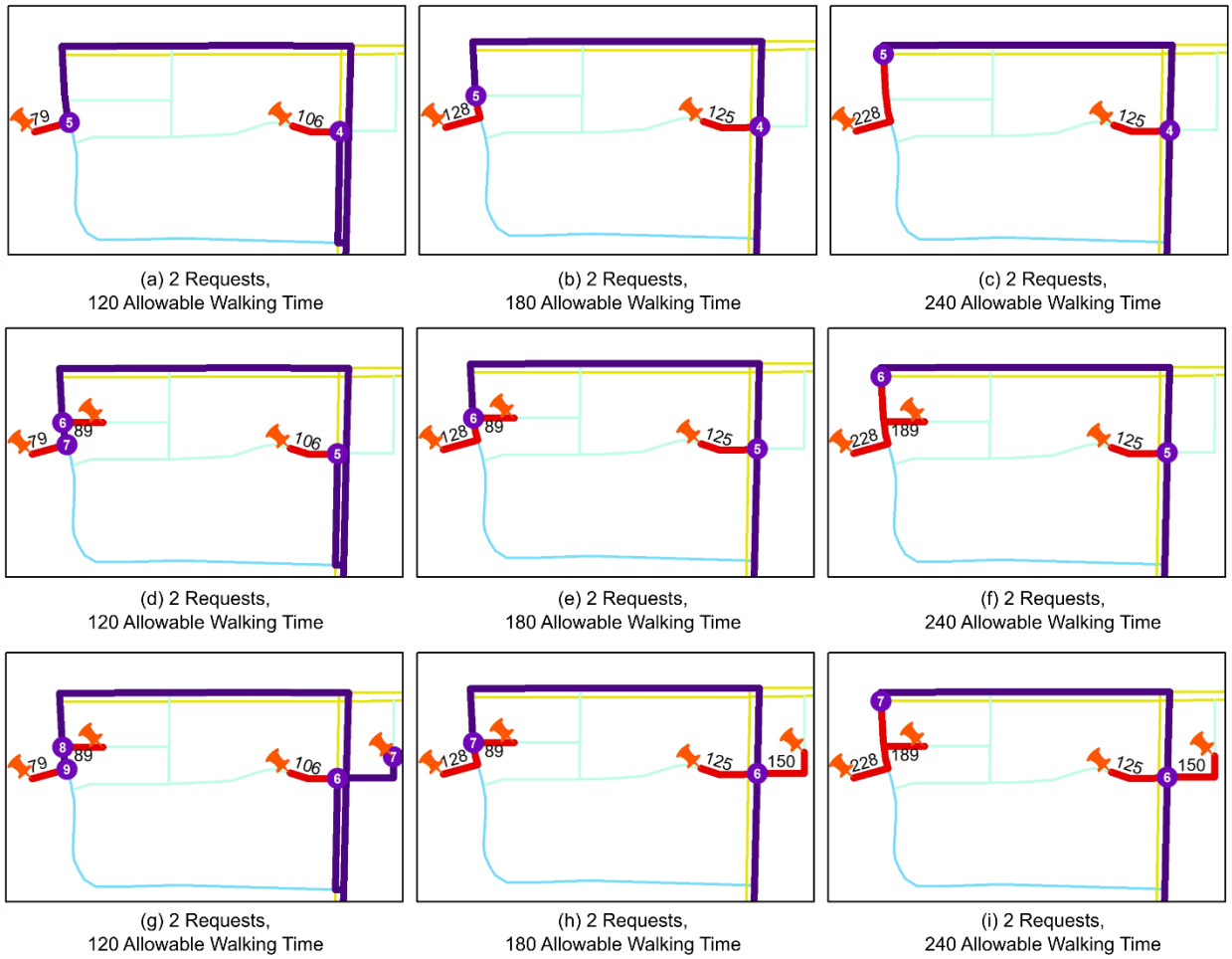


**Figure 5 Vehicle routes (purple) and walking legs (red) illustrative example results for request destination locations (orange marker)**

## 4.3 Experimental Results

This section provides plots of the summary statistics across a variety of service performance metrics.

Some performance metrics focus on the system overall performance. These metrics include the:

- vehicle driving time—the primary objective function term;
- the driving time reduction compared with a door-to-door service;
- the system walking time—the walking time of both pickup and drop-off walking legs across all requests;
- and the driving time reduction per second of walking (DTRPSW).

These performance metrics are shown in Figures 6 and 7. Each dot on the line of a sub-plot in Figure 6 and 7 corresponds to either the median or mean of 50 problem instances.

Some performance metrics focus on the riders' experience. The summary statistics of those metrics are calculated on an agent (individual) basis rather than on a problem instance basis. For example, a two-request problem instance implies two riders, and fifty 2-request problem instances imply a total of 100 agents (riders). These performance metrics include:

- the pickup walking time—the walking time of the pickup walking leg alone;
- the curbside waiting time;
- the total waiting time which includes both the curbside waiting time and the pickup walking time;
- the in-vehicle travel time—the time in which the rider is actually traveling along with the vehicle, not including the time for getting on/off the vehicle;
- the drop-off walking time;
- and the total time needed to reach destination—i.e., total trip time.

Those performance metrics are shown in Figures 8 and 9. Each dot on the line of a sub-plot in Figures 8 and 9 corresponds to either median or mean of 100, 150, or 200 riders (for 2-request, 3-request, 4-request, respectively). The x-axis is the maximum allowable walking time per leg ($\theta$) ranging from 0 second to 360 seconds with a 60-second increment.

Some performance metrics focus on the driver's experience. These metrics include the vehicle service time (from the initial time that the vehicle is ready to serve any traveler to the time that the last rider is dropped-off) and the vehicle waiting time (a vehicle may wait for a request's pickup time window to come or wait for a traveler walking to the designated pickup location; it includes the time for riders to get on/off the vehicle). Figures 10 and 11 show those performance metrics. Since each problem instance is associated with a single vehicle, each dot on the line of a sub-plot corresponds to either median or mean of 50 problem instances. The x-axis is the maximum allowable walking time per leg ($\theta$) ranging from 0 second to 360 seconds with a 60-second increment.

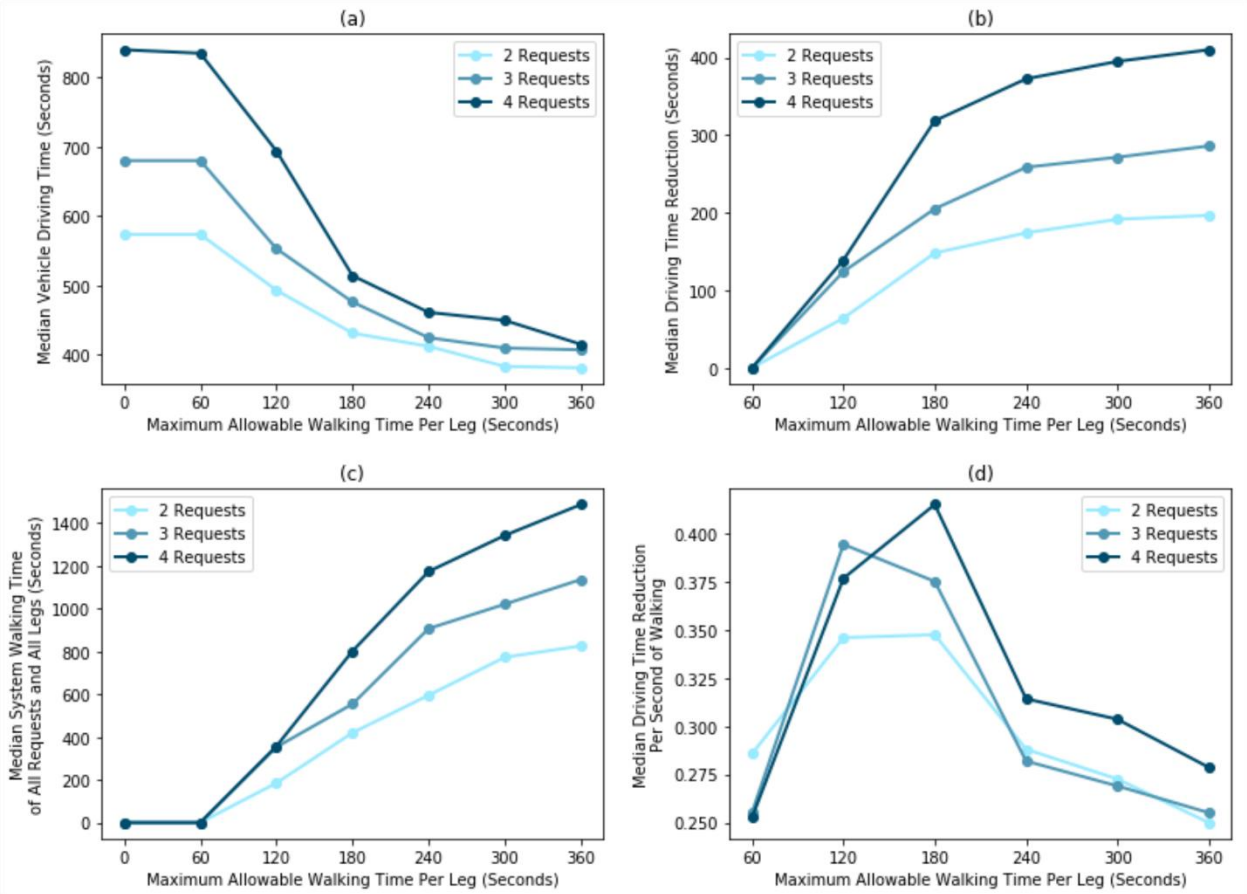


**Figure 6 Case Study I system performance measures. (a) Median vehicle driving time, (b) Median driving time reduction, (c) Median system walking time, (d) Median driving time reduction per second of walking.**

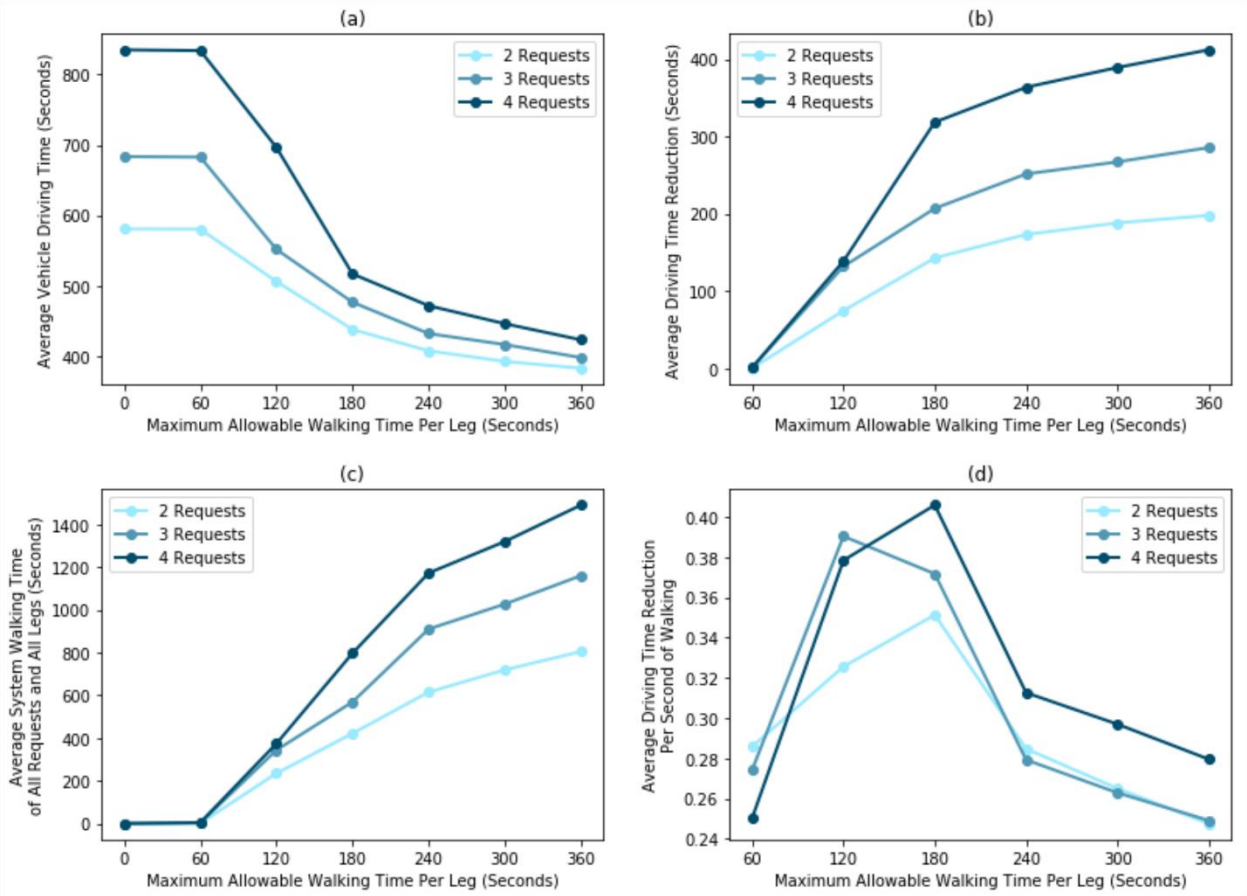


**Figure 7 Case Study I system performance measures. (a) Mean vehicle driving time, (b) Mean driving time reduction, (c) Mean system walking time, (d) Mean driving time reduction per second of walking.**

Figures 6(a)(b) and 7(a)(b) suggest that the vehicle driving time reduces rapidly when $\theta$ increases from one to three minutes, and the vehicle driving time does not decrease much when $\theta$ is greater than three minutes. This is likely because the transportation network is relatively small; a near-direct (bus-like) route can be obtained by just allowing two to three minutes of walking. Such a convex functional relation between the allowable walking time and the vehicle driving time might not be easily observed in a larger transportation network.

Figures 6(a)(b) and 7(a)(b) also show that the more requests, the larger the reduction in vehicle driving time associated with enabling walking. The benefit of introducing walking trip legs into the ridesharing vehicle routes is more prominent if more requests are pooled into one vehicle. For example, when $\theta$ equals to three minutes, the driving time reduction is somewhat more than 100 seconds for a typical 2-request

problem instance. In contrast, the driving time reduction is more than 300 seconds for a typical 4-request problem instance.

As shown in Figures 6(a) and 7(a), the vehicle driving time varies greatly among 2-request, 3-request, and 4-request problem instances when no or little walking is allowed ($\theta$ is less than two minutes). However, when a sufficient among of walking is allowed (e.g., six minutes), the vehicle driving time of different number of requests converges to about 400 seconds. Such a convergence in vehicle driving time might be caused by the direct, bus-like route associated with requiring walking trips, as shown in the illustrative example. This converged vehicle driving time is likely to reflect the approximated travel time from the origin region (Isla Vista) to the destination region (the shopping mall). The setup of this case study might greatly contribute to this observation: requests' origins and destinations are always confined to a relatively small geographic area, and the vehicle's initial location is fixed at the corner of the study area.

Figures 6(c) and 7(c) show that the combined traveler walking time unsurprisingly increases with the number of requests to serve, and $\theta$. In this specific case study, the increase in combined traveler walking time with respect to changes in $\theta$ tends to slow down when $\theta$ is greater than four minutes. One possible explanation is the relatively small size of the transportation network, which might lead to a quick convergence to a near-direct route. When $\theta$ is four minutes, the model typically assigns travelers (in each problem instance) to walk a combined 600 seconds (2-request), 900 seconds (3-request), or 1200 seconds (4-request).

Notably, the increase in walking time for customers is over three times higher than the decrease in vehicle driving time shown in Figures 6(b) and 7(b). In other words, when the time is measured in the same unit, solutions to the PDPTWWL increase total walking time substantially more than they reduce vehicle driving time. Therefore, the driving time reduced per second walking (DTRPSW) is less than one in all instances—see Figures 6(d) and 7(d). Figures 6(d) and 7(d) also show that the highest DTRPSW occurs when $\theta$ equals to two or three minutes, and the DTRPSW does not monotonically increase as $\theta$ increases.

Figures 8 and 9 display the passenger-related metrics. Figures 8(a) and 9(a) show that the pickup walking time generally increases as $\theta$ increases. Figures 8(b) and 9(b) show that the curbside waiting time

tends to decrease as $\theta$ increases. However, such a reduction in the curbside waiting time is likely caused by the fact that some time is used for walking. For example, when $\theta$ is six minutes, 2-request problem instances tend to have zero curbside waiting time, and this does not suggest that the vehicle picks up the request immediately after the request is made; rather, the vehicle might arrive at the pickup location earlier than the traveler and must wait for the traveler. The total waiting time that encompasses both the pickup walking time and the curbside waiting time is shown in Figures 8(c) and 9(c). The total waiting time shows a steadily increasing pattern when $\theta$ is greater than two minutes. In other words, allowing travelers to walk to their designated pickup locations can delay the time when a traveler can get on the vehicle (if we allow the vehicle to wait for the traveler).

Figures 8(d) and 9(d) show the in-vehicle travel time, which generally decreases as $\theta$ increases. In addition, the more requests there are, the larger reduction in the in-vehicle travel time associated with walking tends to be. Those trends are consistent with the vehicle driving time. Theoretically, the reduction in the in-vehicle travel time is caused by a more direct (or less detoured) route, and some portion of the door-to-door trip is now completed by walking.

Figures 8(e) and 9(e) show the drop-off walking time, which generally increases as $\theta$ increases. However, when $\theta$ is between four to six minutes, the drop-off walking time appears to not change too much. This observation is likely due to the transportation network structure, in which the number of road nodes within the drop-off region is sparse, so this suddenly stopped increasing pattern in the drop-off walking time might not be applicable to other networks.

Figures 8(f) and 9(f) show the time needed to reach destination (total trip time). The fewer number of requests being pooled in one vehicle, the larger increase in the total trip time. For example, when $\theta$ is six minutes, the total trip time in a typical 2-request instance increases about 200 seconds, while it only increases about 75 seconds in a typical 4-request instance. Additionally, for a typical 2-request instance, only a small walking amount (i.e., $\theta$ is two minutes) leads to a comparatively large increase in the total trip time. On the other hand, a typical 3-request or 4-request instance tends to have a comparatively small

increase (or even slightly decrease) in the total trip time. Interestingly, for a typical 4-request instance, the total trip time has a prominent drop when $\theta$ is three minutes. It is unclear whether such a drop in the total trip time is generally applicable to other networks (though later I shall show that a similar pattern can be observed in the Chicago downtown case study). It is also unclear whether a more prominent drop in the total trip time can be observed if the number of requests being pooled in one vehicle is even larger (e.g., five or six requests).

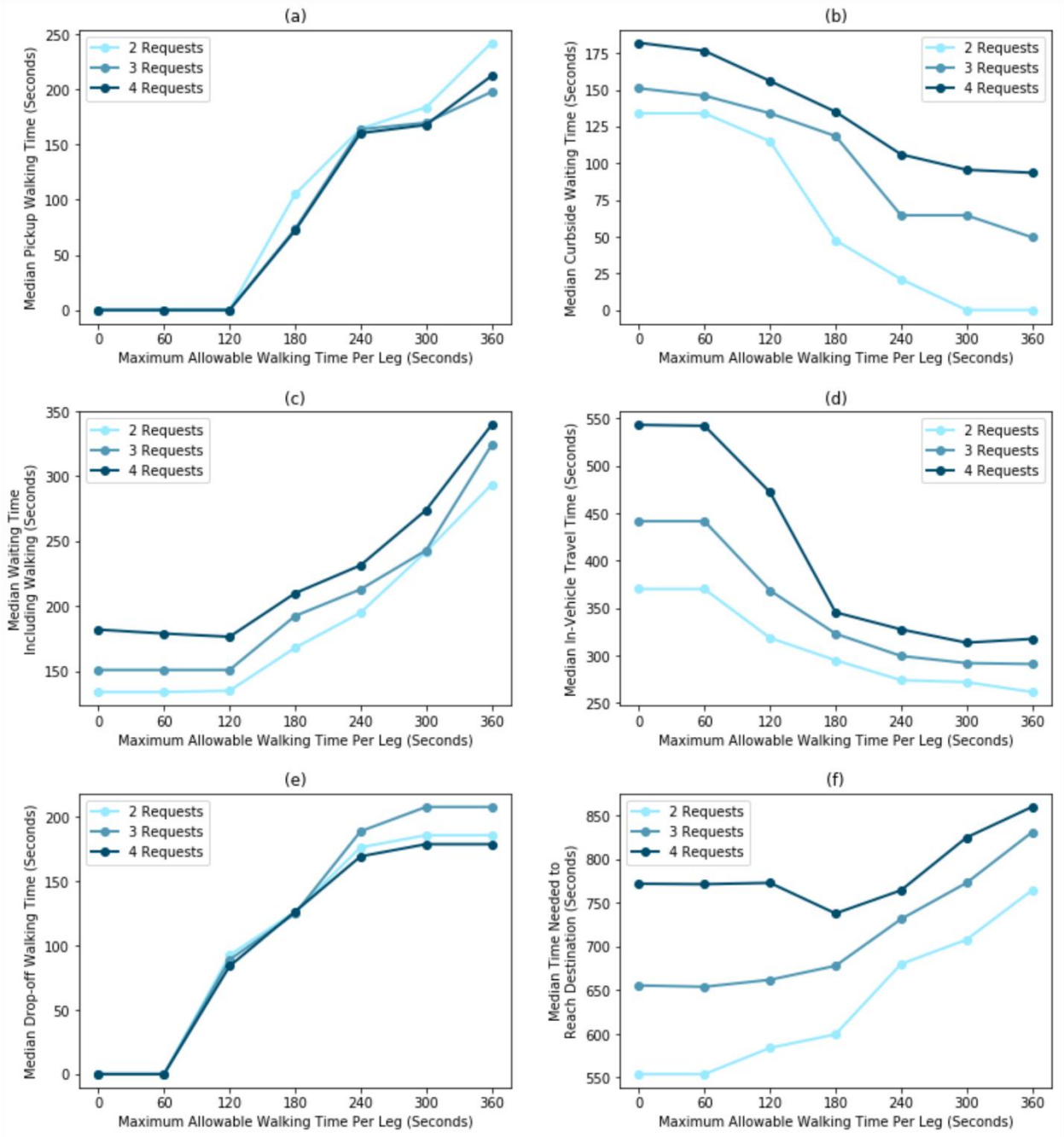


**Figure 8 Case Study I rider experience performance measures. (a) Median pickup walking time, (b) Median curbside waiting time, (c) Median waiting time including walking time, (d) Median in-vehicle travel time, (e) Median drop-off walking time, (f) Median time needed to reach destination.**

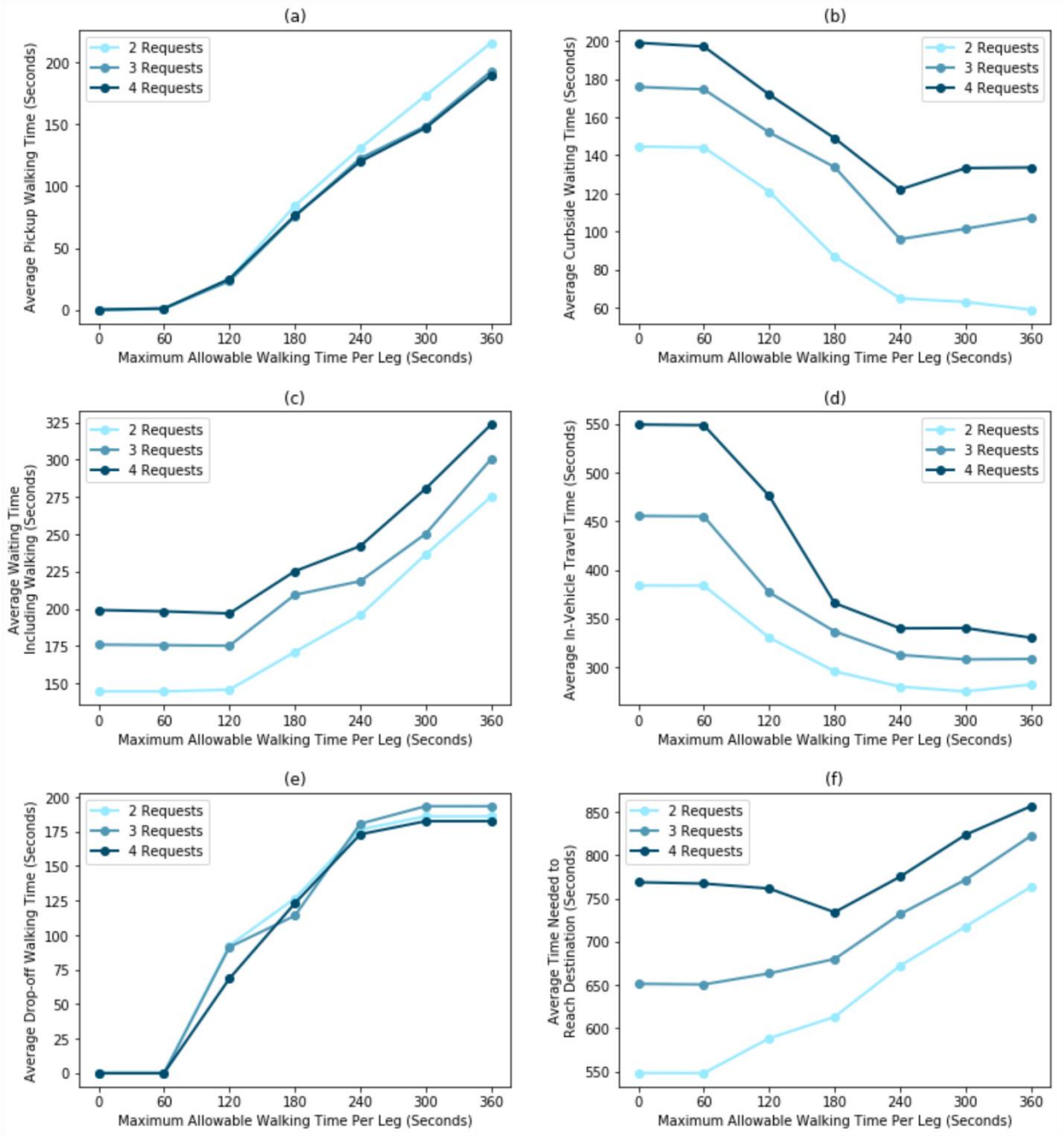


**Figure 9 Case Study I rider experience performance measures. (a) Mean pickup walking time, (b) Mean curbside waiting time, (c) Mean waiting time including walking time, (d) Mean in-vehicle travel time, (e) Mean drop-off walking time, (f) Mean time needed to reach destination.**

Figures 10 and 11 display the vehicle-related metrics. Figures 10(a) and 11(a) show that the vehicle service time reaches a minimum when $\theta$ is around three to four minutes. Before this interval of maximum allowable walking time, the vehicle service time tends to decrease as more walking is allowed; after this interval, the vehicle service time starts increasing as more walking is allowed. Such a convex relationship

between $\theta$ and the vehicle service time might be related to the steady increase in the vehicle waiting time (Figures 10(b) and 11(b)) and the traveler waiting time (Figures 8(c) and 9(c)) as $\theta$ gets larger. It might also be related to the slow-down in the vehicle driving time reduction (due to the relatively small size of the network) when $\theta$ is greater than four minutes (Figures 6(b) and 7(b)). If the vehicle spends too much time in waiting for certain travelers to walk to the designated pickup locations, the benefits (i.e., vehicle driving time reduction) associated with a more direct route might not be enough to compensate the additional vehicle waiting time resulted by allowing travelers to walk longer. Therefore, this Isla Vista case study shows that the vehicle service time might increase as $\theta$ increases in certain experimental setups, but there is no guarantee that such an increase would always be observed (see the Chicago downtown case study).

Figures 10(b) and 11(b) show the vehicle waiting time. When $\theta$ is zero, the vehicle waiting time is 40, 60, and 80 seconds for 2-request, 3-request, and 4-request respectively, and those values correspond to the total time for all travelers sharing a vehicle to get on and get off the vehicle. When $\theta$ is greater than two minutes, the vehicle waiting time steadily increases as $\theta$ increases. This pattern reflects the fact that the vehicle may arrive at the pickup location earlier than the traveler. It seems that, for every second that is allowed for the travelers to walk, there could be slightly more than 0.8 second increase in the vehicle waiting time. However, there is no guarantee that the increase in the vehicle waiting time per second of maximum allowable walking time is always less than one. For example, consider a walking-enabled ridesharing service that dynamically inserts new requests into the existing vehicle route, the vehicle might wait for the first traveler at one location for four minutes, and later it receives a second request and waits for the second traveler at another location for four minutes, while the maximum allowable walking time is five minutes. In this scenario, for every second that is allowed for walking, there could be 1.6 seconds increase in the vehicle waiting time.

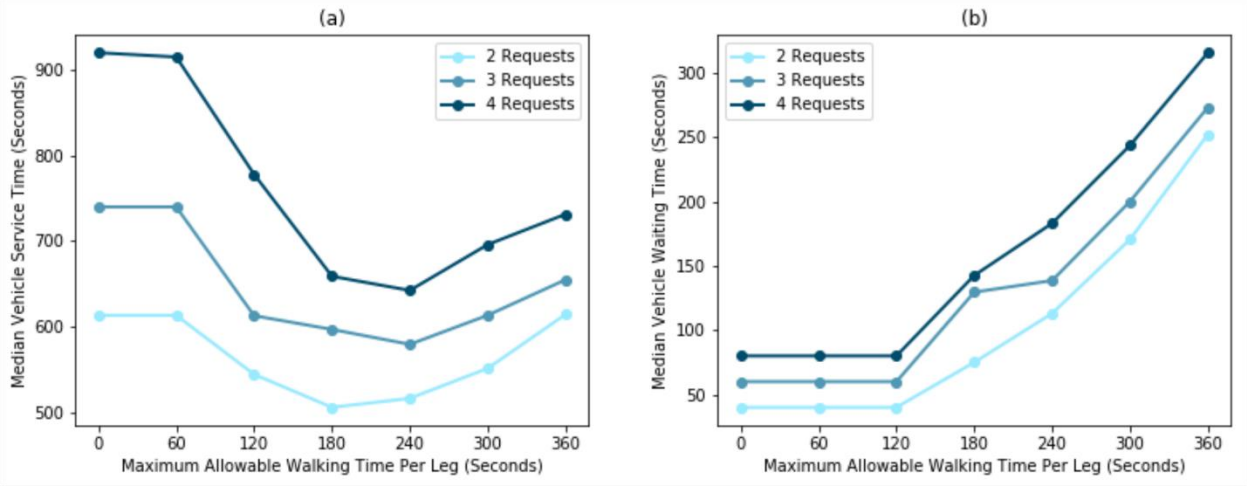


**Figure 10 Case Study I driver experience performance measures. (a) Median vehicle service time, (b) Median vehicle waiting time.**

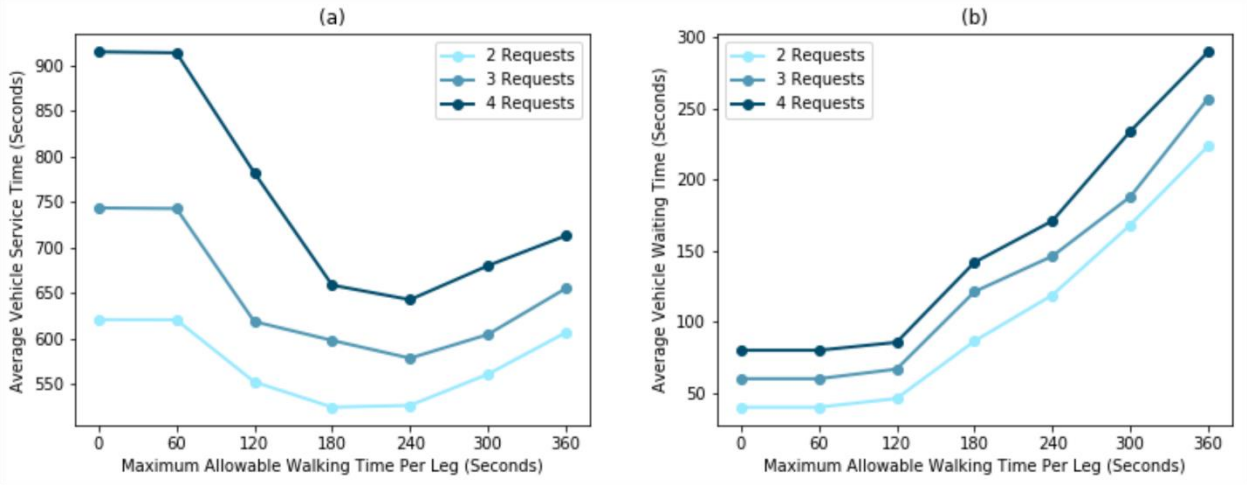


**Figure 11 Case Study I driver experience performance measures. (a) Mean vehicle service time, (b) Mean vehicle waiting time.**

## 4.4 Computational Complexity Results

This section presents the summary statistics of problem size and computational time, which are shown in Table 4. For each number of requests being pooled in one vehicle (2, 3, or 4) and each value of maximum allowable walking time ($\theta$), there are a total of 50 problem instances being solved, and the summary statistics of those 50 problem instances form a row in Table 4. This thesis quantifies the problem size by analyzing the size of the PUDO clusters (the number of vertices in each cluster). For each problem instance, there are four to eight pickup and drop-off clusters depending on the number requests being pooled in one vehicle, and it is possible to compute the average, minimum, median, and maximum cluster size as well as

the total number of vertices for that specific problem instance. An average value over fifty problem instances can be computed to obtain the average cluster size per problem instance, with the cluster size (for each problem instance) measured in average, minimum, median, or maximum number of vertices.

The ILP is solved using Gurobi 9.0.2 on a PC with 2.30 GHz CPU and 8 GB memory.

When the number of requests is only two (i.e. two requests are pooled in one vehicle), the single-vehicle PDPTWWL problem can be easily solved to optimality. The worst case takes only 0.320 second to solve, while on average 0.106 second is needed to solve a problem with two requests and 360 seconds of maximum allowable walking time.

When the number of requests is three, the computational time is not terrible, considering that the model is solved exactly. When $\theta$ is six minutes, it only takes 0.683 second to solve the model on average, while the worst case can be solved within two seconds. When $\theta$ is no more than four minutes, the worst case only takes 0.197 second.

When the number of requests is four, the model becomes quite difficult to solve. The worst case (when $\theta$ is six minutes) takes 51.5 seconds to solve. Even when $\theta$ is four minutes, the computational time of the worst case takes 4.65 seconds, which is still too long for a dynamic setting.

This case study uses a small network that does not have a large cluster size, so a more realistic application of the PDPTWWL model may entail longer computational time, such as in the Chicago downtown case study which is presented in the next chapter.

**Table 4 Case Study I problem size and computational time summary statistics**

| # of Requests | Max Allowable Walking Time Per Leg, $\theta$, (s) | Average Pickup/Drop-off Cluster Size Per Problem Instance | | | | | Computational Time (s) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Average | Min. | Median | Max. | Total # of Vertices | Average | Median | 0.9 Quantile | Max. |
| 2 | **0** | 1.0 | 1.0 | 1.0 | 1.0 | 4.0 | 0.002 | 0.002 | 0.003 | 0.004 |
| | **60** | 1.1 | 1.0 | 1.0 | 1.3 | 4.3 | 0.002 | 0.002 | 0.003 | 0.004 |
| | **120** | 2.2 | 1.4 | 2.1 | 3.2 | 8.7 | 0.006 | 0.006 | 0.010 | 0.014 |
| | **180** | 3.9 | 2.0 | 3.6 | 6.3 | 15.6 | 0.017 | 0.016 | 0.027 | 0.041 |
| | **240** | 5.4 | 2.6 | 5.2 | 8.7 | 21.7 | 0.023 | 0.021 | 0.032 | 0.040 |
| | **300** | 8.8 | 5.8 | 8.3 | 12.5 | 35.0 | 0.039 | 0.036 | 0.049 | 0.129 |
| | **360** | 11.8 | 7.3 | 11.2 | 17.4 | 47.1 | 0.106 | 0.091 | 0.185 | 0.320 |
| 3 | **0** | 1.0 | 1.0 | 1.0 | 1.0 | 6.0 | 0.007 | 0.007 | 0.008 | 0.009 |
| | **60** | 1.1 | 1.0 | 1.0 | 1.5 | 6.6 | 0.007 | 0.007 | 0.009 | 0.018 |
| | **120** | 2.2 | 1.2 | 2.0 | 3.7 | 13.4 | 0.020 | 0.018 | 0.028 | 0.033 |
| | **180** | 4.0 | 1.7 | 3.8 | 6.9 | 24.3 | 0.038 | 0.032 | 0.063 | 0.080 |
| | **240** | 6.2 | 2.6 | 6.2 | 10.1 | 37.4 | 0.115 | 0.115 | 0.166 | 0.197 |
| | **300** | 9.5 | 5.1 | 9.2 | 14.3 | 56.9 | 0.320 | 0.283 | 0.483 | 0.646 |
| | **360** | 12.4 | 6.8 | 11.7 | 19.4 | 74.1 | 0.683 | 0.600 | 1.353 | 1.950 |
| 4 | **0** | 1.0 | 1.0 | 1.0 | 1.0 | 8.0 | 0.013 | 0.012 | 0.014 | 0.044 |
| | **60** | 1.1 | 1.0 | 1.0 | 1.7 | 9.0 | 0.021 | 0.014 | 0.023 | 0.224 |
| | **120** | 2.1 | 1.0 | 2.0 | 3.9 | 17.1 | 0.051 | 0.037 | 0.073 | 0.376 |
| | **180** | 4.2 | 1.6 | 4.0 | 7.3 | 33.5 | 0.152 | 0.117 | 0.222 | 1.471 |
| | **240** | 6.3 | 2.4 | 6.1 | 10.5 | 50.1 | 0.403 | 0.274 | 0.523 | 4.651 |
| | **300** | 9.7 | 4.6 | 9.3 | 14.7 | 77.4 | 1.109 | 0.842 | 1.964 | 6.517 |
| | **360** | 12.3 | 6.1 | 11.4 | 20.2 | 98.3 | 3.514 | 1.741 | 4.560 | 51.495 |

# 5 Case Study II: Chicago Downtown

## 5.1 Data and Parameter Settings

In this Chapter, the proposed ILP model is applied to a larger, more realistic urban network—Chicago downtown network. This case study mimics a morning commuting application, in which travelers need to go from somewhere north of the Chicago River to the downtown Chicago. Some highlights of this case study include the consideration of one-way restrictions, the use of street centerline midpoints as request origins and destinations, and the use of a dummy depot as the vehicle's initial location.

The street center lines data was downloaded from the City of Chicago's open data portal (City of Chicago: Data Portal, 2020). This dataset has attributes including one-way and z-level (for tiered roads), which enable a relatively realistic modeling of the real-world transportation network. Only a small portion of the downloaded street centerlines is used to create the network dataset in ArcMap: those street center lines within the community area 32 and its adjacent community areas and have a class code not corresponding to extent, river, sidewalk, and unclassified. Based on Google Maps, a manual review and editing process was also conducted on this small portion of the downloaded street center lines dataset. A few missing road segments were digitized, and a few road segments that appear to not exist were removed. The direction of travel (one-way) attribute of road segments was also reviewed extensively, and a few edits were made as needed. The quality of the data of Lower Wacker Dr. is not guaranteed since its validation would require additional high-quality data sources and time for reviewing. However, the generation of the requests' origins and destinations as well as the candidate alternative pickup and drop-off locations were setup in a way that the vehicle routes should utilize Lower Wacker Dr. sparsely. The walking speed is assumed to be 3 feet per second, and the driving speed on alleys is also set to 3 feet per second to discourage the use of alleys. The driving speed of all other drivable road segments is set to 13 feet per second (8.8 mph).

The request origins are first generated at a zonal (census tract) level based on derived sample probabilities (Table 5), and the road center lines' midpoints within each census tract are sampled assuming a uniform distribution. The zonal sample probabilities are derived from the Transportation Network

Providers Trips dataset available on the City of Chicago's open data portal (City of Chicago: Data Portal, 2019). The trips that have the following characteristics are eventually used to derive the probabilities: (i) drop-off timestamp is between 7:30 A.M. and 8:30 A.M. on Tuesday October 8, 2019; (ii) the drop-off census tract has a geoid of 17031839100 (downtown Chicago); (iii) the pickup census tracts are the eleven census tracts north of Chicago River (census tract geoids are shown in the second column of Table 5). The total trip miles of the filtered TNC trips sum to 415 miles, and the total trip hours of the filtered TNC trips sum to approximately 47 hours. The average speed is thus 8.8 mph (the basis for the assumed driving speed).

**Table 5 Case Study II zonal sample probabilities**

| Pickup Zone ID | Census Tract Geoid | Hourly Trips to Downtown | Probability |
|---|---|---|---|
| 1 | 17031080400 | 5 | 0.017985612 |
| 2 | 17031080300 | 15 | 0.053956835 |
| 3 | 17031080201 | 13 | 0.04676259 |
| 4 | 17031080202 | 20 | 0.071942446 |
| 5 | 17031838300 | 5 | 0.017985612 |
| 6 | 17031081000 | 43 | 0.154676259 |
| 7 | 17031081100 | 17 | 0.061151079 |
| 8 | 17031081900 | 6 | 0.021582734 |
| 9 | 17031081800 | 54 | 0.194244604 |
| 10 | 17031081700 | 67 | 0.241007194 |
| 11 | 17031081600 | 33 | 0.118705036 |
| **Total** | NA | **278** | **1** |

All requests are assumed to have a destination in the downtown Chicago, specifically, the census tract with a geoid of 17031839100. The study area consisting of 11 pickup zones and one drop-off zone is shown in Figures 12 and 13. An Arcpy script is used create midpoints on both sides (with a six-foot offset) of the road segments in those census tracts, and a manual editing procedure is performed to remove midpoints that are too clustered or along the Wacker Drive. Those midpoints are overlaid with the census tracts so that each midpoint would correspond to one and only one census tract, and the midpoints within a census tract can be sampled assuming an equal probability to be selected.

The request origins, destinations, and candidate PUDO locations are the street centerline midpoints. Such an approach better reflects the fact that real-world requests are generated along the curbside of the streets instead of at road junctions. However, census tract boundaries sometimes are along street centerlines, and this pattern can lead to arbitrariness in assigning a midpoint that is not generated using an offset to a census tract. Using midpoints with an offset addresses the two problems above, but it also entails shortcomings. Under a realistic setting, the vehicle generally could not make a U-turn in the middle of a street, and it should continue to the intersection and then make the U-turn. While this case study assumes midpoints to be candidate PUDO locations (vehicle stops), there is no mechanism in the optimization model to ensure a realistic U-turn behavior in the prescribed vehicle route, and an underestimate of the vehicle driving time might occur in an arbitrarily way, regardless of whether walking is allowed.

Because each road segment has two midpoints (one to the left and one to the right) being created and used for demand generation, the number of PUDO locations would be too large and thus make the ILP model extremely difficult to solve. The number of midpoints must be reduced prior to running the optimization model. This study arbitrarily keeps one of the two midpoints corresponding to the same road segment. The generated demand information (request origins and destinations) is updated to reflect this consolidation of midpoints, and the travel time matrices are computed for the midpoints that remain after the consolidation.

Initially, two hundred sets of 2-request instances, two hundred sets of 3-request instances, and two hundred sets of 4-request instances are generated. However, some generated demands may have origins and destinations within 12 minutes walking time, and those demands thus could be fully accommodated by walking. Any set of requests that contain at least one such request were removed from the analysis. Therefore, only 197 two-request instances, 198 three-request instances, and 195 four-request instances are used to compute summary statistics that are presented in the experimental results and computational complexities results sections.

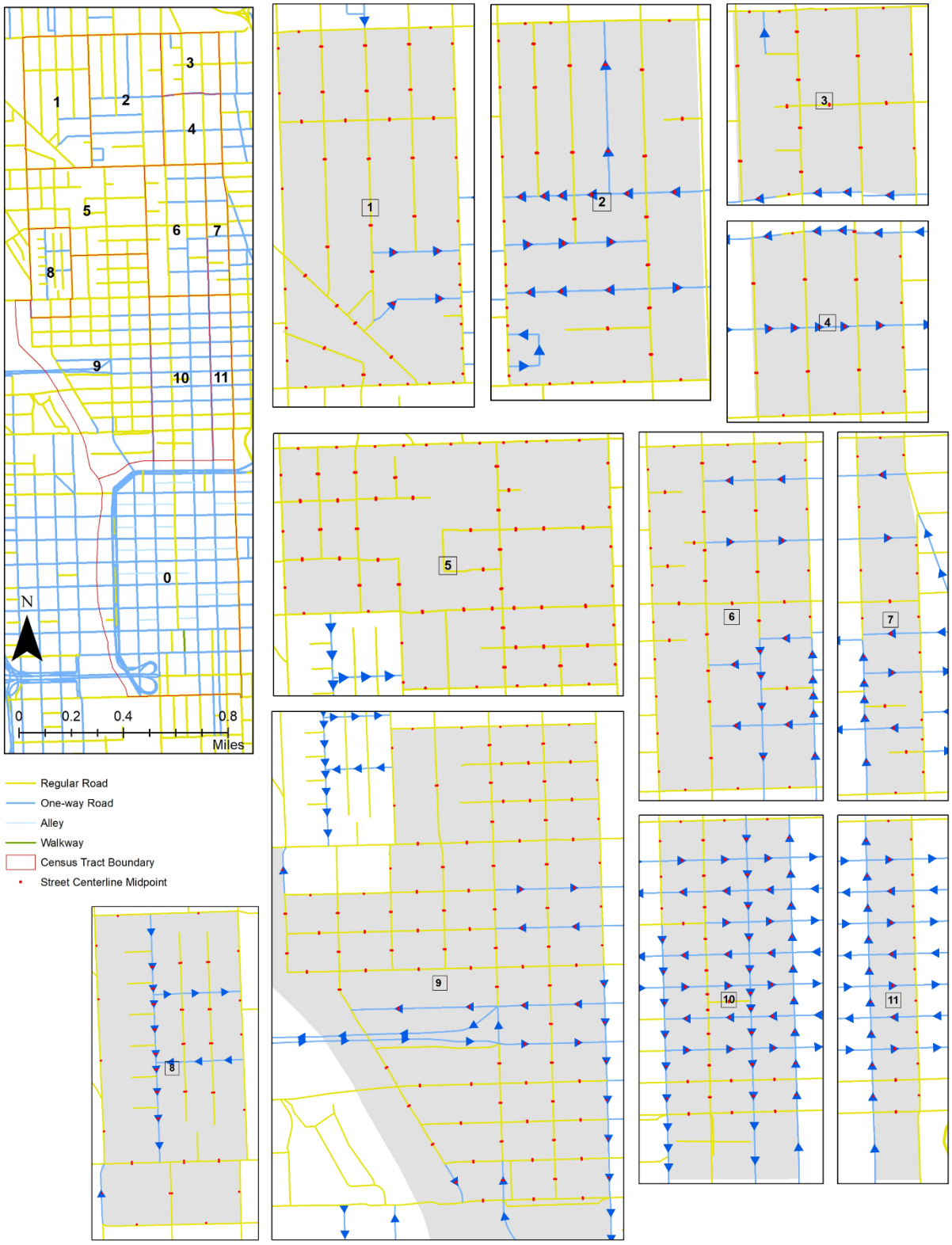


**Figure 12 The Chicago downtown network map: the entire study area and each of the 11 pickup zones.**

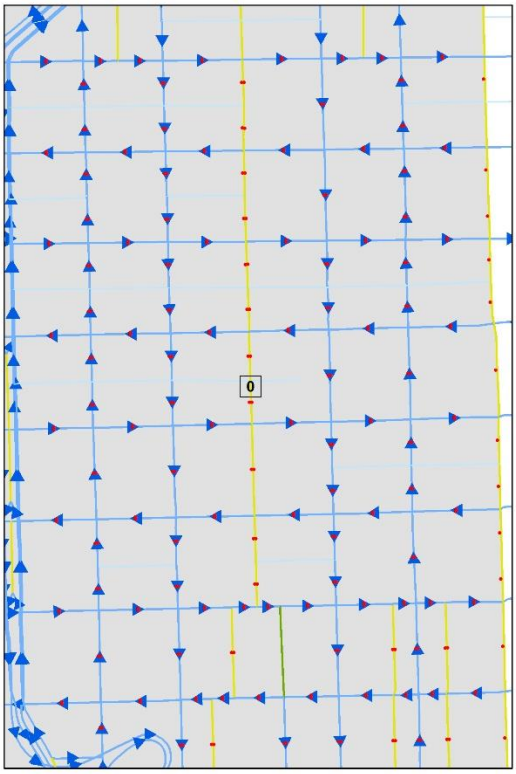


**Figure 13 The Chicago downtown network map: the drop-off zone.**

Similar to the previous Isla Vista case study, all requests are assumed to be made at the same time (time 0), and each request includes only one traveler. The dummy depot idea is also used to model the final vehicle destination.

Unlike the previous Isla Vista case study that assumes a fixed initial vehicle location for all generated instances, the Chicago downtown case study utilizes the idea of dummy depot that is represented by a vertex not corresponding to any real-world location to model the vehicle's initial location. Effectively, the edges connecting the initial vehicle location vertex with all vertices in all pickup clusters would have a zero weight (driving time). Assuming that the vehicle departs from its initial location immediately, the vehicle would immediately appear at the pickup location of the first request that the vehicle would pick up, and the vehicle would wait at that location until the traveler arrives at the pickup location. Certain service performance metrics are sensitive to this assumption, and a detailed exploration of this assumption's implications is presented in Section 5.5.

**Table 6 Model Parameter Settings for Case Study II**

| Parameters | Values |
|---|---|
| $\alpha$ | 0.001 |
| *MWT* | 1800 seconds (30 minutes) |
| *d* | 10 seconds |
| *MDT* | 300 seconds (5 minutes) |
| *MaxWalkTime* | $2\theta$ |
| *C* | 5 |
| *Relative MIP Gap* | 1e-7 |
| *Integer Feasibility Tolerance* | 1e-7 |

## 5.2 Illustrative Examples

This section presents two illustrative examples for the 4-request case. Both examples are arbitrarily selected among the 195 four-request instances.

Figures 14 and 15 give a visualization of the vehicle routes of Example 1 and 2, respectively.

Figure 14(a) shows that the request origins of Example 1 are in a relatively confined geographic area while the request destinations are located throughout a relatively larger area. Figures 14(b)(c)(d) demonstrate the effect of one-way streets, evident by the loops in the vehicle route for picking up request 1 and 3 or dropping off request 4. As $\theta$ increases, such loops become smaller or eventually eliminated. Figure 14(e) shows one effect of the using dummy depot as the vehicle's initial location. In all other cases, the vehicle picks up request 2 first, but when $\theta$ is four minutes, the vehicle picks up request 3 first. Figure 14(f) illustrates that $\theta$ of five minutes can transform a highly circuitous route into a direct, bus-like vehicle route for this specific set of requests. Figures 14(e)(f) show that a location where multiple travelers are collectively picked up can be spontaneously formed. In Figure 14(e), Requests 1 and 4 are required to walk to the same location (the second vehicle stop) for pickup. Similarly, in Figure 14(f), Requests 1, 3, and 4 are designated to be picked up at the same location. The proposed model does not group requests based on certain criteria and assign a common pickup location to all of them, but those travelers are still picked up at the same location based on the model solution.


**Figure 14 Vehicle routes (purple) and walking legs (red) of Case Study II illustrative example 1 results for request origin locations (green marker) and request destination (orange marker)**

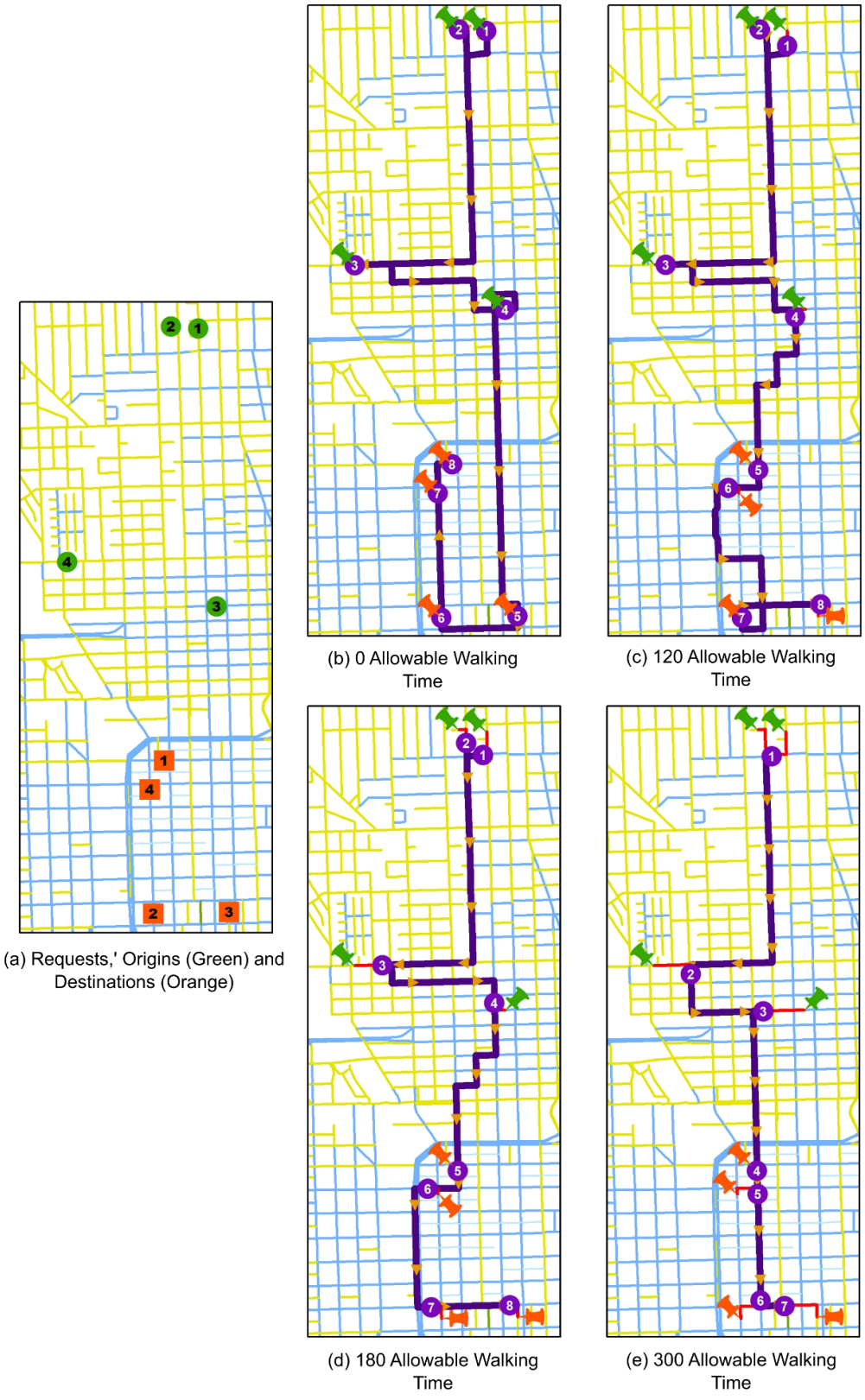


**Figure 15 Vehicle routes (purple) and walking legs (red) of Case Study II illustrative example 2 results for request origin locations (green marker) and request destination (orange marker)**

Figure 15(a) shows that the request origins of Example 2 are largely scattered throughout the area north of the Chicago River while the request destinations form a "L" shape. Figure 15(b) shows one interesting effect of one-way streets. The road where requests 2 and 4 are located is northbound, but the request destinations where the vehicle heads to are to the south of request origins. The vehicle route without walking would require the vehicle to travel southward all the way near the southern boundary of the study area and then drive northward all the way near the Chicago River. As walking is allowed, such an interesting pattern (which also leads to significant increase in VMT/VHT) disappears.

Table 7 shows some key performance measures of the two illustrative examples with five different values of the maximum allowable walking time.

**Table 7 Key Performance Measures of the Case Study II Illustrative Examples**

| | | Example 1 | Example 2 |
|---|---|---|---|
| **Maximum Allowable Walking Time, $\theta$, (s)** | **0** | D: 1138<br>W: 0/0<br>F: [874, 984, 1218, 636]<br>C: 0.0099611 | D: 1930<br>W: 0/0<br>F: [2010, 1734, 1576, 1933]<br>C: 0.0100236 |
| | **120** | D: 967 (-171)<br>W: 214/179<br>F: [909, 1119, 1237, 671] (+224)<br>C: 0.1998954 | D: 1740 (-190)<br>W: 279/275<br>F: [1406, 1769, 2014, 1514] (-550)<br>C: 0.0799565 |
| | **180** | D: 805 (-333)<br>W: 249/412<br>F: [770, 980, 1128, 834] (0)<br>C: 0.4597948 | D: 1465 (-465)<br>W: 603/447<br>F: [1351, 1732, 1843, 1459] (-868)<br>C: 0.2198462 |
| | **240** | D: 641 (-497)<br>W: 816/636<br>F: [811, 1123, 1185, 831] (+238)<br>C: 1.5591807 | D: 1391 (-539)<br>W: 670/873<br>F: [1508, 1950, 1815, 1581] (-399)<br>C: 0.5896878 |
| | **300** | D: 560 (-578)<br>W: 620/689<br>F: [756, 1050, 1162, 820] (+76)<br>C: 7.9157338 | D: 1119 (-811)<br>W: 1069/822<br>F: [1269, 1657, 1751, 1448] (-1128)<br>C: 0.6896060 |

D: Vehicle driving time (reduction in vehicle driving time compared with no walking allowed)
W: Walking time of all requests' pickup legs/Walking time of all requests' drop-off legs
F: [Request 1's total trip time, Request 2's total trip time, …] (change in the sum of total trip time of all requests compared with no walking allowed)
C: Computational time

Both illustrative examples show that a substantial reduction in vehicle driving time can be achieved by instructing the travelers to walk for a short distance. When $\theta$ is five minutes, the vehicle driving time reduces 578 seconds or 50.8% compared with the no walking case in Example 1, and the vehicle driving

time reduces 811 seconds or 42.0% compared with the no walking case in Example 2. When $\theta$ is only three minutes, the vehicle driving time reduces 333 seconds or 29.3% in Example 1, and the vehicle driving time reduces 465 seconds or 24.1% in Example 2.

One of the main sacrifices to achieve the reduction in vehicle driving time is walking. For each second of reduction in vehicle driving time, a two to three seconds of system-wide walking time will be needed according to the two illustrative examples. The DTRPSW is the highest when $\theta$ is three minutes in those two illustrative examples, although the experimental results suggest that the DTRPSW is the highest when $\theta$ is two minutes.

As $\theta$ increases, the walking time of all requests combined does not always show a monotonic increasing pattern. In Example 1, the pickup walking time of all requests combined is 816 seconds when $\theta$ is four minutes, but it decreases to 620 seconds when $\theta$ is five minutes. When $\theta$ is five minutes, request 2 and request 3 need to walk about 252 seconds to their pickup locations (20 seconds more than the case with $\theta$ of four minutes). On the other hand, request 1 needs to walk for only 115 seconds instead of 230 seconds by allowing one extra minute of walking, and request 4 does not need to walk at all when $\theta$ is five minutes, in contrast to the 115-second of walking when $\theta$ is four minutes .

The total trip time for all requests being pooled in a vehicle is non-monotonic with respect to changes in $\theta$. In Example 1, the system total trip time starts with 3712 seconds when $\theta$ is zero minute, increases to 3936 seconds when $\theta$ is two minutes, drops to 3712 seconds (coincidentally, the same value as the case when no walking is allowed) when $\theta$ is three minutes, increases again to 3950 seconds when $\theta$ is four minutes, and drops again to 3788 seconds when $\theta$ is five minutes. In Example 2, the system total trip time shows a decreasing pattern until $\theta$ is three minutes, and it increases (though is still lower than the case when no walking is allowed) when $\theta$ is four minutes, and it drops again to reach the global minimum (among all $\theta$ being tested) when $\theta$ is five minutes. However, in both illustrative examples shown in this case study, the system total trip time does not increase compared with the no walking case when $\theta$ is three minutes.

This observation confirms that the total trip time does not always increase as more walking is allowed, particularly if the number of requests being pooled in one vehicle is sufficiently large.

## 5.3 Experimental Results

This section provides the summary statistics plots for a variety of service performance metrics. The metrics that focus on the system overall performance are shown in Figures 16 and 17. Each dot on the line of a sub-plot in Figures 16 and 17 corresponds to either median or mean of 197, 198, and 195 problem instances for 2-request, 3-request, and 4-request, respectively. The metrics that focus on the riders' experience are shown in Figures 18 and 19. Each dot on the line of sub-plot in Figures 18 and 19 corresponds to either median or mean of 394, 594, and 780 riders for 2-request, 3-request, and 4-request, respectively. The metrics that focus on the drivers' experience are shown in Figures 20 and 21. Each dot on the line of sub-plot in Figures 20 and 21 corresponds to either median or mean of 197, 198, and 195 problem instances for 2-request, 3-request, and 4-request, respectively.

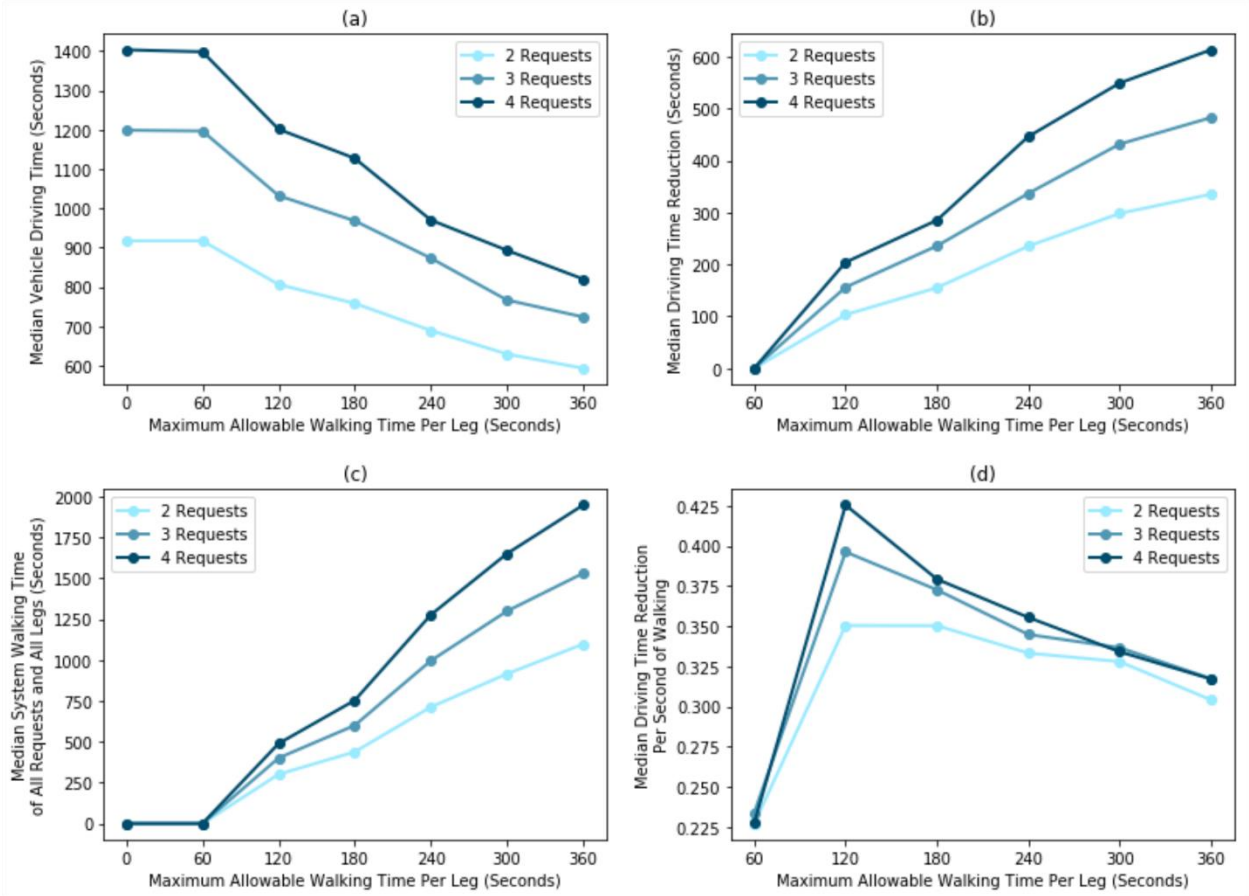


**Figure 16 Case Study II system performance measures. (a) Median vehicle driving time, (b) Median driving time reduction, (c) Median system walking time, (d) Median driving time reduction per second of walking.**

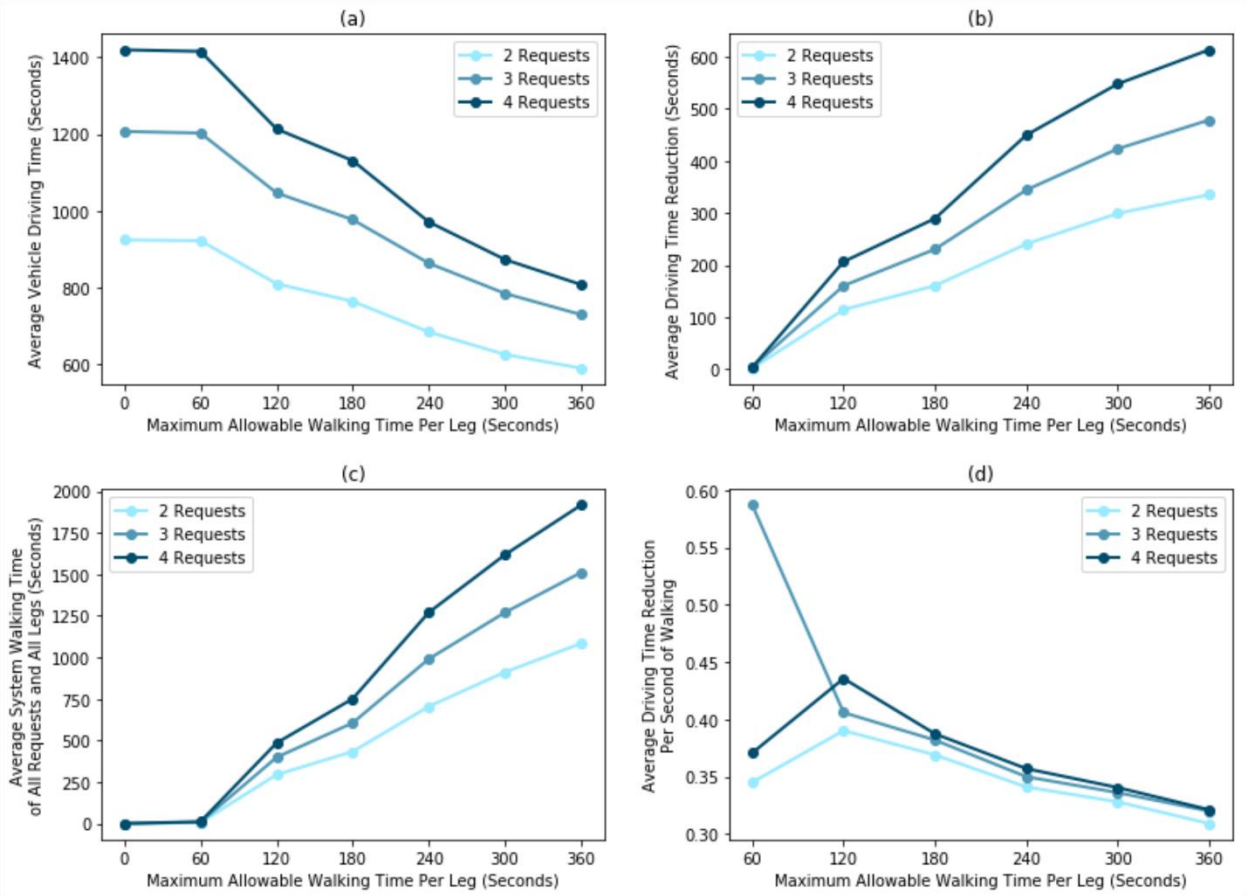


**Figure 17 Case Study II system performance measures. (a) Mean vehicle driving time, (b) Mean driving time reduction, (c) Mean system walking time, (d) Mean driving time reduction per second of walking.**

Figures 16(a)(b) and 17(a)(b) suggest that the vehicle driving time continuously decreases as $\theta$ increases from one to six minutes. This is slightly different from Case Study I, and the reason is that the Isla Vista network is too small and has a faster convergence towards a direct, bus-like route.

Figures 16(a)(b) and 17(a)(b) also show that the more requests being pooled into one vehicle, the larger the reduction in vehicle driving time can be observed if sufficient walking is enabled. Such a general observation is consistent with Case Study I.

In Figures 6(a) and 7(a), the vehicle driving time of different number of requests eventually converges as sufficient walking time is allowed. Although this pattern is not observed in Figures 16(a) and 17(a), the difference between a typical 2-request vehicle driving time and a typical 4-request vehicle driving time drops from 500 seconds (no walking is allowed) to 300 seconds (six-minute walking per leg is allowed). Additionally, enabling enough walking time (e.g., six minutes) can reduce the typical vehicle driving time

of a 4-request instance to a level that is below the typical door-to-door vehicle driving time of a 2-request instance.

Figures 16(c) and 17(c) show that the combined traveler walking time increases with the number of requests to serve, and $\theta$. When $\theta$ is four minutes, the model typically assigns travelers (in each problem instance) to walk a combined 750 seconds (2-request), 1000 seconds (3-request), or 1250 seconds (4-request). This walking time of all travelers increases by about 3 seconds for every one second decrease in vehicle driving time shown in Figures 16(b) and 17(b). This observation is confirmed by Figures 16(d) and 17(d) which show a DTRPSW of around 0.35. Figures 16(d) and 17(d) also show that the highest DTRPSW typically occurs when $\theta$ equals to two minutes, and the DTRPSW generally decreases as $\theta$ is greater than two minutes.

Compared to Case Study I, all four system performance metrics tend to have a higher value in Case Study II. The typical vehicle driving time is longer (which is not surprised as the network is larger), the typical driving time reduction is larger (perhaps because of the consideration of one-way restriction and a denser network), the typical system walking time is longer (perhaps because of a denser network), and the typical DTRPSW tends to have a higher value as well.

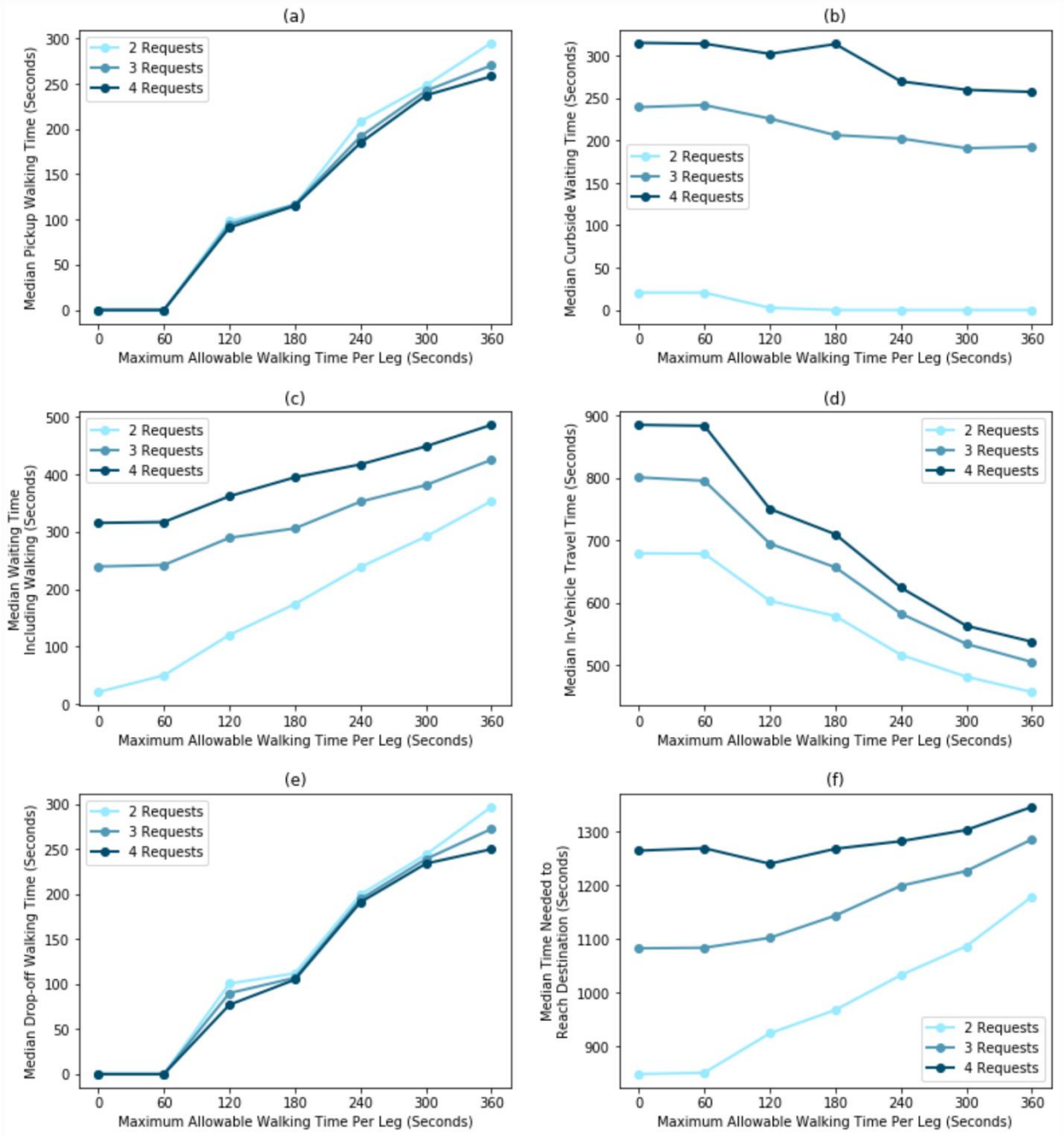


**Figure 18 Case Study II rider experience performance measures. (a) Median pickup walking time, (b) Median curbside waiting time, (c) Median waiting time including walking time, (d) Median in-vehicle travel time, (e) Median drop-off walking time, (f) Median time needed to reach destination.**

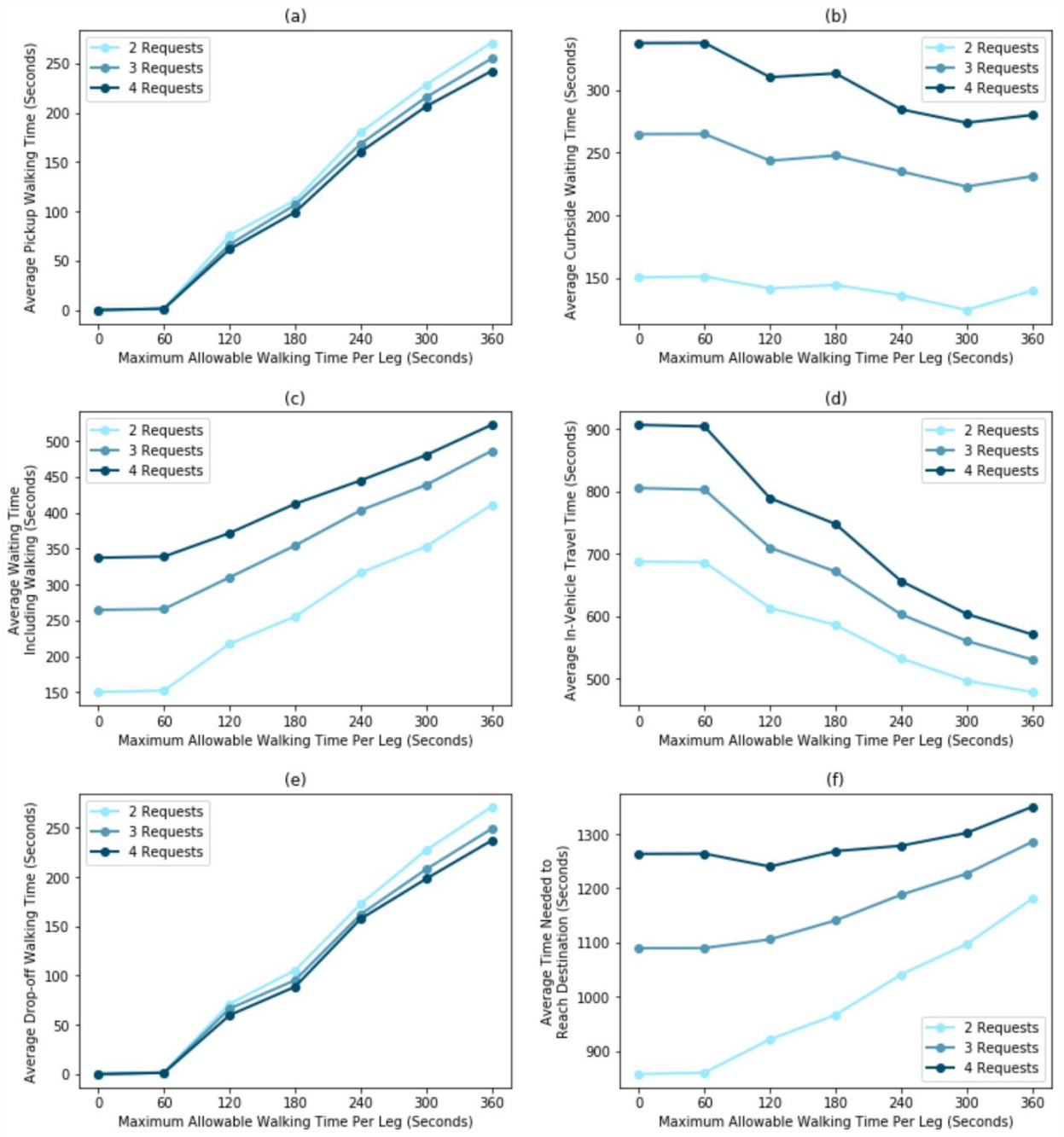


**Figure 19 Case Study II rider experience performance measures. (a) Mean pickup walking time, (b) Mean curbside waiting time, (c) Mean waiting time including walking time, (d) Mean in-vehicle travel time, (e) Mean drop-off walking time, (f) Mean time needed to reach destination.**

Figures 18(a) and 19(a) unsurprisingly show that the pickup walking time generally increases as $\theta$ increases. Figures 18(b) and 19(b) show that the curbside waiting time tends to decrease as $\theta$ increases, although the decreasing pattern is not monotonic and is not rapid. Interestingly, the median curbside waiting time of 2-request instances drops to zero second when $\theta$ is at least 180 seconds. The use of dummy depot

as the initial vehicle location perhaps is the main cause of this observation. The vehicle would always wait for the first traveler to arrive at the pickup location. Given that those instances have only two requests (travelers), half of those travelers would guarantee to have a curbside waiting time of zero second. If any one of those instances have the vehicle arrives at the second pickup location before the second traveler does, the median curbside waiting time will be zero. In contrast, taking average would take the effect of extreme large values into account, and the plot of mean shifts the curve up about 150 seconds compared with the median plot.

The total waiting time that encompasses both the pickup walking time and the curbside waiting time is shown in Figures 18(c) and 19(c). The total waiting time (for each traveler) shows a steadily increasing pattern when $\theta$ is greater than one minute (slightly earlier than Case Study I). When $\theta$ is six minutes, the median waiting time increases by 333 seconds, 186 seconds, and 171 seconds for 2-request, 3-request, and 4-request, respectively. The average waiting time increases by 260 seconds, 221 seconds, and 185 seconds for 2-request, 3-request, and 4-request, respectively. In general, the larger the number of requests being pooled in a vehicle, the increase in total waiting time resulted by enabling walking would be less sensitive to change in the maximum allowable walking time. It is worth noting that those results are heavily influenced by the use of dummy depot as initial vehicle location.

Figures 18(d) and 19(d) show the in-vehicle travel time, which generally decreases as $\theta$ increases. When $\theta$ is six minutes, the median in-vehicle travel time reduces by 222 seconds, 295 seconds, and 347 seconds for 2-request, 3-request, and 4-request, respectively. The average in-vehicle travel time reduces by 209 seconds, 274 seconds, and 335 seconds for 2-request, 3-request, and 4-request, respectively. In general, the larger the number of requests being pooled in a vehicle, the reduction in in-vehicle travel time resulted by enabling walking would be more prominent.

Figures 18(e) and 19(e) show the drop-off walking time, which generally increases as $\theta$ increases.

Figures 18(f) and 19(f) show the total trip time. The total trip time is sensitivity (increases rapidly) to the increase in the maximum allowable walking time in a typical 2-request instance (an average of 322 seconds increase when $\theta$ is six minutes). On the other hand, the total trip time of a typical 4-request instance

tends to increase gradually as $\theta$ increases (an average of 87 seconds increase when $\theta$ is six minutes), and a drop (an average of 24 seconds reduction) can even be observed when $\theta$ is around two minutes.

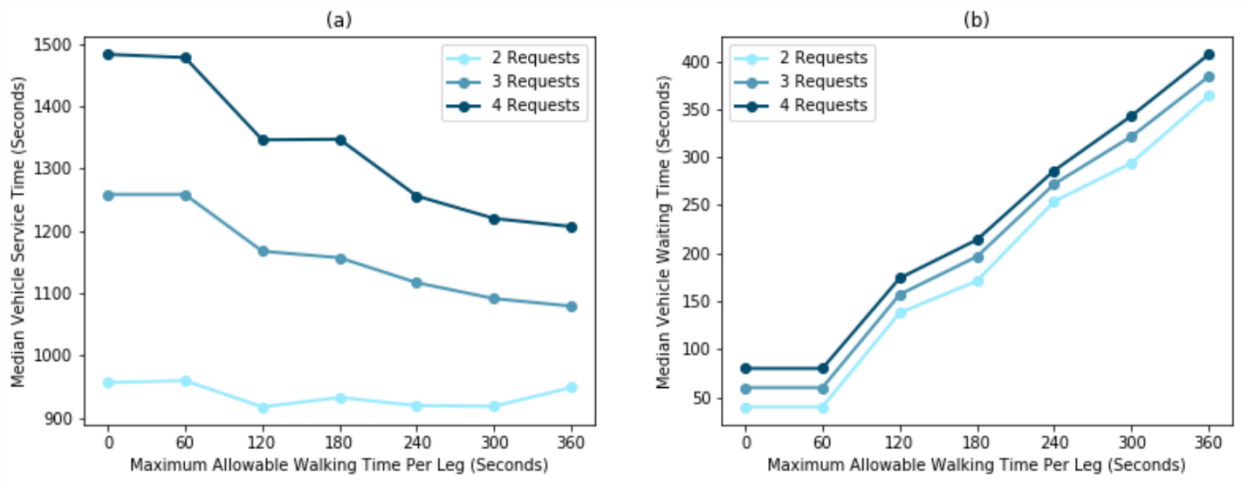


**Figure 20 Case Study II driver experience performance measures. (a) Median vehicle service time, (b) Median vehicle waiting time.**

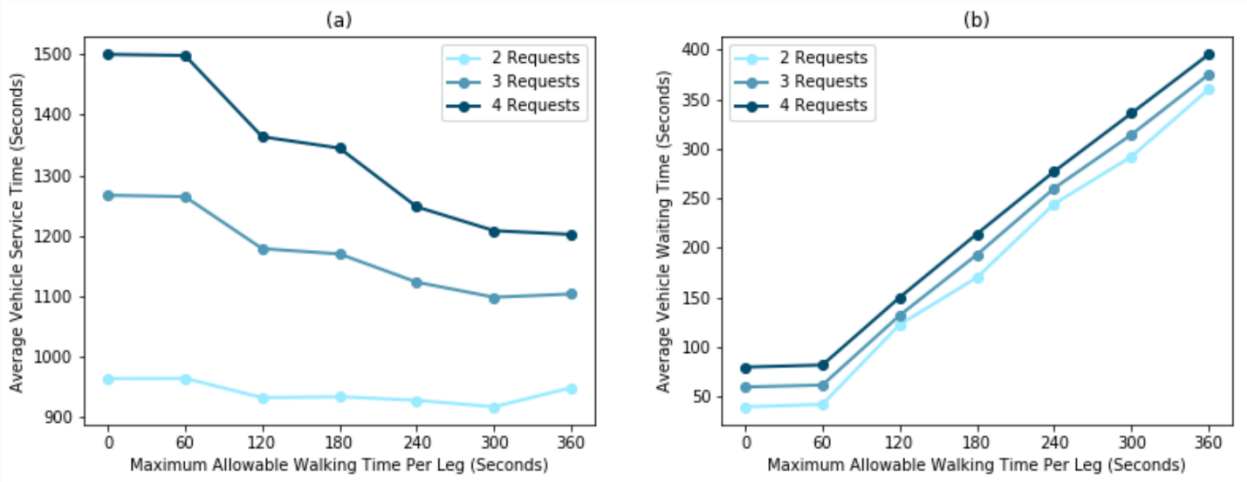


**Figure 21 Case Study II driver experience performance measures. (a) Mean vehicle service time, (b) Mean vehicle waiting time.**

Figures 20(a) and 21(a) show that the vehicle service time tends to decrease as $\theta$ increases especially for 3-request and 4-request. This pattern is not exactly the same as in Case Study I which shows a convex relation between $\theta$ and the vehicle service time. The convex relation can be attributed to the slow-down in the vehicle driving time reduction (due to the relatively small size of the network in Case Study I). The vehicle driving time reduction in Case Study II, in contrast, does not show a strong sign of slow-down in increasing as $\theta$ increases. Nevertheless, there is a possibility that the vehicle driving time reduction would eventually slow down (at some larger values of $\theta$) so that a clear convex relation can be observed for this

Chicago downtown network. Particularly, the average vehicle service time reaches the local minimum when $\theta$ is five minutes for both 2-request (47 seconds reduction) and 3-request (169 seconds reduction). When $\theta$ is six minutes, the vehicle service time could reduce by 15 seconds, 163 seconds, and 298 seconds on average for 2-request, 3-request, and 4-request, respectively.

Figures 20(b) and 21(b) show the vehicle waiting time. When $\theta$ is zero minute, the vehicle waiting time is 40, 60, and 80 seconds for 2-request, 3-request, and 4-request respectively, and those values correspond to the total time for all travelers sharing a vehicle to get on and get off the vehicle. When $\theta$ is greater than one minutes (slightly earlier than Case Study I), the vehicle waiting time steadily increases as $\theta$ increases. As mentioned in Case Study I, this pattern reflects the fact that the vehicle may arrive at the pickup location earlier than the traveler. Interestingly, the vehicle waiting time increases linearly as $\theta$ increases, especially when average is used, and the lines representing 2-request, 3-request, and 4-request are also close to each other. One possible explanation is that the vehicle's initial location is too close to (or located exactly at) the first pickup location, in both case studies, and the vehicle has to wait for the traveler to arrive.

## 5.4 Computational Complexity Results

This section presents the summary statistics of problem size and computational time, which are shown in Table 8.

When the number of requests is only two (i.e., two requests are pooled in one vehicle), the single-vehicle PDPTWWL problem can be easily solved to optimality. The worst case takes only 0.730 second to solve, while on average 0.264 second is needed to solve a problem with two requests and 360 seconds of maximum allowable walking time.

When the number of requests is three, the model may take a few seconds to solve in some extreme cases. When $\theta$ is six minutes, it takes 1.582 seconds to solve the model on average, while the worst case needs 8.366 seconds to solve. When $\theta$ is no more than three minutes, the worst case takes 0.300 second.

When the number of requests is four, the model becomes extremely difficult to solve. The worst case (when $\theta$ is six minutes) takes 1799.9 seconds to solve. On average, it takes 69.6 seconds to solve a problem with $\theta$ of six minutes. When $\theta$ is three minutes, the computational time of the worst case takes 2.62 seconds. However, if Constraints (23)—the cluster-variant of Dantzig-Fulkerson-Johnson subtour elimination constraints—are added to the model formulation, the computational time can reduce dramatically for 4-request instances. On average, it takes 11.2 seconds to a solve a problem with $\theta$ of six minutes, and the worst case takes 175.9 seconds to solve.

**Table 8 Case Study II problem size and computational time summary statistics**

| # of Requests | Max Allowable Walking Time Per Leg, $\theta$, (s) | Pickup/Drop-off Cluster Size Per Problem Instance | | | | | Computational Time (s) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Average | Min. | Median | Max. | Total # of Vertices | Average | Median | 0.9 Quantile | Max. |
| 2 | 0 | 1.0 | 1.0 | 1.0 | 1.0 | 4.0 | 0.002 | 0.000 | 0.010 | 0.010 |
| | 60 | 1.1 | 1.0 | 1.0 | 1.3 | 4.4 | 0.002 | 0.000 | 0.010 | 0.010 |
| | 120 | 4.1 | 2.6 | 4.0 | 5.6 | 16.2 | 0.012 | 0.010 | 0.020 | 0.040 |
| | 180 | 6.8 | 5.3 | 6.8 | 8.5 | 27.4 | 0.022 | 0.020 | 0.030 | 0.070 |
| | 240 | 12.5 | 8.7 | 12.3 | 16.4 | 49.8 | 0.058 | 0.050 | 0.094 | 0.210 |
| | 300 | 19.6 | 14.2 | 19.9 | 24.4 | 78.3 | 0.131 | 0.130 | 0.214 | 0.330 |
| | 360 | 26.9 | 19.8 | 27.1 | 33.8 | 107.7 | 0.264 | 0.240 | 0.420 | 0.730 |
| 3 | 0 | 1.0 | 1.0 | 1.0 | 1.0 | 6.0 | 0.004 | 0.000 | 0.010 | 0.010 |
| | 60 | 1.1 | 1.0 | 1.0 | 1.4 | 6.5 | 0.005 | 0.010 | 0.010 | 0.020 |
| | 120 | 4.1 | 2.1 | 4.1 | 5.9 | 24.3 | 0.034 | 0.030 | 0.050 | 0.100 |
| | 180 | 6.7 | 4.7 | 6.7 | 8.8 | 40.4 | 0.087 | 0.080 | 0.150 | 0.300 |
| | 240 | 12.5 | 7.7 | 12.4 | 17.5 | 75.0 | 0.260 | 0.240 | 0.430 | 0.740 |
| | 300 | 19.4 | 12.9 | 19.7 | 25.1 | 116.6 | 0.699 | 0.545 | 1.277 | 3.358 |
| | 360 | 26.8 | 17.9 | 26.9 | 35.2 | 160.6 | 1.582 | 1.284 | 2.732 | 8.366 |
| 4 | 0 | 1.0 | 1.0 | 1.0 | 1.0 | 8.0 | 0.013 | 0.010 | 0.020 | 0.060 |
| | 60 | 1.1 | 1.0 | 1.0 | 1.6 | 8.8 | 0.017 | 0.010 | 0.030 | 0.110 |
| | 120 | 4.1 | 1.9 | 4.1 | 6.2 | 32.7 | 0.135 | 0.090 | 0.266 | 0.979 |
| | 180 | 6.8 | 4.5 | 6.8 | 9.2 | 54.3 | 0.500 | 0.310 | 1.123 | 2.619 |
| | 240 | 12.7 | 7.4 | 12.6 | 18.3 | 101.3 | 2.826 | 1.119 | 4.815 | 62.037 |
| | 300 | 19.6 | 12.3 | 19.9 | 26.0 | 156.8 | 15.356 | 3.998 | 32.790 | 224.810 |
| | 360 | 26.9 | 17.2 | 27.1 | 36.8 | 215.5 | 69.587 | 13.233 | 167.906 | 1799.887 |

**Table 9 Case Study II computational time for 4-request instances with the cluster-variant of Dantzig-Fulkerson-Johnson subtour elimination constraints**

| # of Requests | Max Allowable Walking Time Per Leg, $\theta$, (s) | Computational Time (s) | | | |
|---|---|---|---|---|---|
| | | Average | Median | 0.9 Quantile | Max. |
| 4 | **0** | 0.009 | 0.010 | 0.010 | 0.020 |
| | **60** | 0.010 | 0.010 | 0.010 | 0.030 |
| | **120** | 0.095 | 0.080 | 0.150 | 0.400 |
| | **180** | 0.297 | 0.250 | 0.466 | 1.009 |
| | **240** | 1.064 | 0.920 | 1.583 | 3.648 |
| | **300** | 3.347 | 2.789 | 5.505 | 18.680 |
| | **360** | 11.153 | 7.216 | 17.635 | 175.936 |

## 5.5 Sensitivity Analysis on the Vehicle Driving Time to Reach the First Pickup Location

Section 5.1. mentions that a dummy depot is used as the vehicle's initial location, and the driving time from the vehicle's initial location to any candidate pickup location is assumed to be zero. Effectively, a parameter that governs how long it will take for the vehicle to go from its initial location to the first pickup location on the vehicle route is implicitly involved, and this parameter is referred to as the time for the vehicle to reach the first pickup location (TVRFPL). The experimental results that are presented in Section 5.3. show that the traveler waiting time and total trip time (Figures 18 and 19) tend to increase as the maximum allowable walking time increases. However, those observations assume that the TVRFPL parameter is zero second. In this Section of the thesis, a sensitivity analysis is performed on the TVRFPL parameter, which demonstrates a profound influence on riders' and drivers' experience of using a walking-enabled ridesharing service.

Figure 22 shows the sensitivity analysis results based on the average of 191 sets of 3-request instances. The maximum allowable walking time per leg, $\theta$, is varied from zero second to 420 seconds (seven minutes), and the TVRFPL parameter is varied from zero minute to five minutes.

Figure 22(a) shows that the average vehicle driving time is longer as the TVRFPL parameter is larger. The larger TVRFPL value implies that the vehicle is further away from the first pickup location, so it

unsurprisingly takes more time to reach the first pickup location and to complete the entire trip. The general observation that more allowable walking reduces more VHT remains unchanged.

Figure 22(b) shows that the total waiting time of a typical rider tends to decrease as more walking is allowed until the TVRFPL parameter has the same value as $\theta$. If the value of the TVRFPL is higher than the value of $\theta$, the vehicle is always expected to arrive at the first pickup location later than the traveler, so the total waiting time would not be affected by how much walking is allowed. Conversely, if the value of the TVRFPL is lower than the value of $\theta$, the vehicle is likely to wait for the traveler, whose arrival time at the pickup location is dependent on how much walking is allowed. The later rider arrival time at the pickup location for a specific traveler implies a longer total waiting time of that traveler as well as any traveler who would be picked up subsequently. In general, if the initial vehicle location is sufficiently away from the first pickup location, a walking-enabled ridesharing service can decrease the total waiting time of travelers.

Figure 22(c) shows that the in-vehicle travel time of a typical rider remains the same as the value of the TVRFPL varies.

Figure 22(d) shows the total trip time for a typical rider tends to decrease as more walking is allowed until the TVRFPL parameter has the same value as $\theta$. This trend is related to the trend observed in Figure 22(b). Without the decreasing pattern of the total waiting time, the decreasing in-vehicle travel time along often cannot offset the extra travel time associated with the walking to the destination and the potential increase in the total waiting time. In general, if the initial vehicle location is further away from the first pickup location, a walking-enabled ridesharing service is more likely to reduce the total trip time compared with a door-to-door service.

Figure 22(e) shows the average vehicle service times corresponding to the different TVRFPL values are equally spaced when $\theta$ is zero second, but as more walking is allowed, the average vehicle service times show a converging pattern. This converging pattern is related to whether the TVRFPL parameter value is smaller or larger than the value of $\theta$; in other words, it is related to whether the vehicle is expected to arrive

at the first pickup location before or after the traveler arrives. When $\theta$ is 120 seconds, only the line corresponding to a TVRFPL value of 60 seconds almost collapses with the line corresponding to a TVRFPL value of 0 second; the lines corresponding to TVRFPL values between 180 seconds and 300 seconds are still evenly spaced. In this case, the prominent converging pattern is only observed between two lines which correspond to TVRFPL values smaller than the value of $\theta$, indicating that the vehicle could arrive at the first pickup location earlier than the traveler. When $\theta$ is 360 seconds, all lines, with TVRFPL values up to 300 seconds, nearly collapse to a single point, and the vehicle tends to wait for the traveler at the first pickup location. In general, enabling walking in a ridesharing service reduces the vehicle service time compared to a door-to-door service, and the magnitude of the reduction can be more prominent if the vehicle's initial location is further away from the first pickup location.

Figure 22(f) shows the average vehicle waiting time which has the lowest possible value of 2**n***d*, where *n* is the total number of requests being pooled, and *d* is the time for a traveler to getting on or off the vehicle. It confirms that the vehicle arrives at the first pickup location after the traveler arrives if the TVRFPL value is equal to or greater than $\theta$, since under such a condition the vehicle waiting time always remains at the lowest possible value. Additionally, if the vehicle's initial location is sufficiently away from the first pickup location, the increase in vehicle waiting time due to enabling walking can be small or even zero.

Overall, this sensitivity analysis demonstrates some benefits of a walking-enabled ridesharing service which are not shown in the previous experimental results. Those benefits include the potential to reduce the waiting time (including pickup walking) as well as the total trip time of travelers compared to a door-to-door service, without explicitly minimizing those metrics. To achieve those benefits, the vehicle's initial location needs to be sufficiently away from the first pickup location so that the vehicle does not need to wait for the traveler. However, assigning a vehicle that is further away to serve a request may entail a longer waiting time and total trip time compared with assigning a vehicle that is closer to the request's origin. Therefore, if none of the vehicles in the fleet are very close to a request's origin, a walking-enabled service is likely to be better than a door-to-door service; otherwise, the vehicle that is very close to the request's

origin may have to make more detour to achieve a low waiting/total trip time (door-to-door), or it may have to retain a low detour while prolonging the total trip time of the new request (enabling walking).

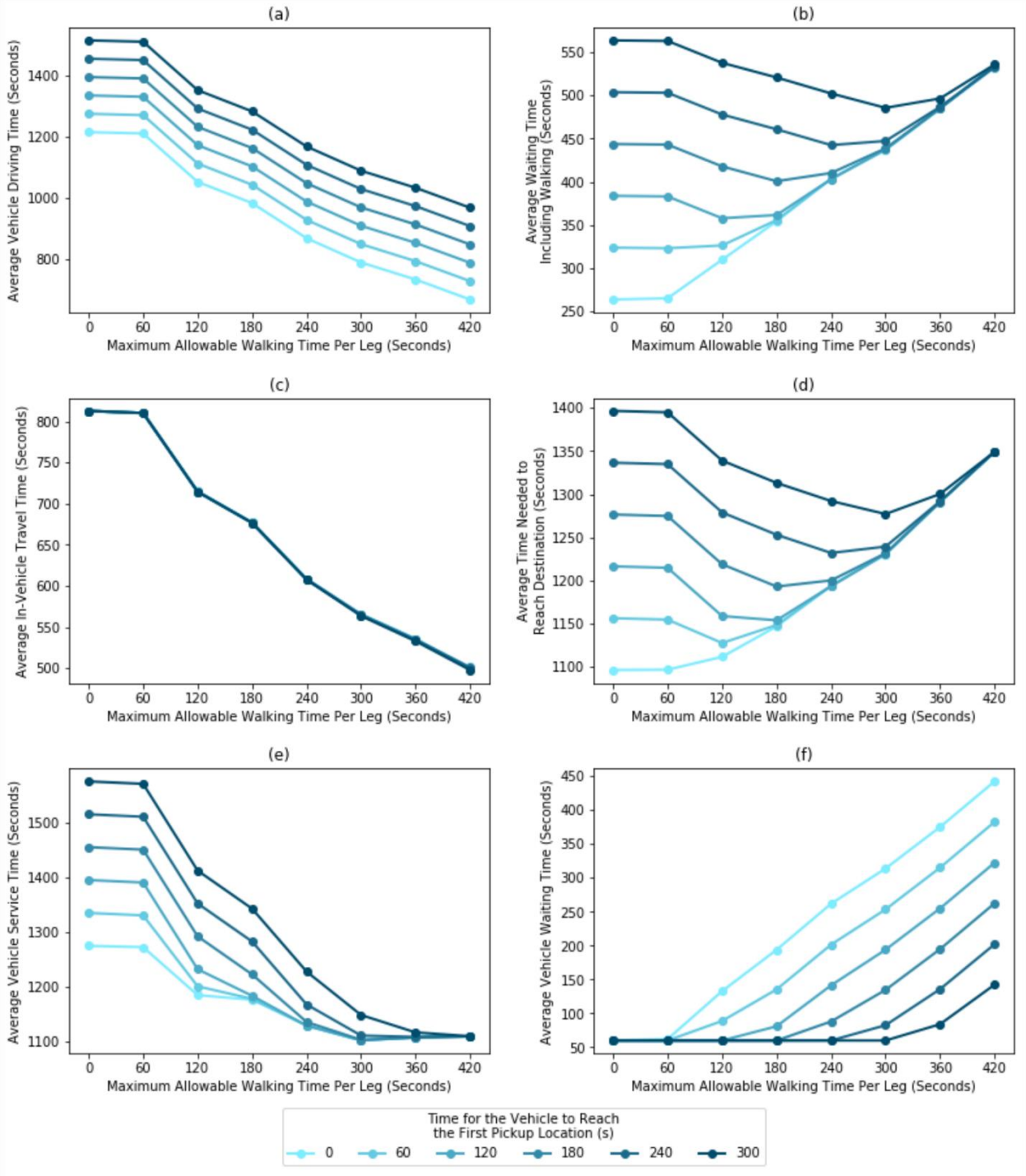


**Figure 22 Case Study II sensitivity analysis on the parameter that governs the time for the vehicle to reach the first pickup location.**

## 5.6 Enabling Only the Pickup Walking Trip Leg or the Drop-off Walking Trip Leg

The results that are presented in previous sections assume a ridesharing service that enables both the pickup walking trip leg and the drop-off walking trip leg. This section analyzes the service performance of enabling only one walking trip leg. To make this analysis more realistic, two scenarios are considered. Both scenarios aim to address the downtown rush-hour congestion by enabling walking in the downtown area only. The first scenario—which mimics morning commute—disables the pickup walking trip leg but keeps the drop-off walking trip leg that should occur in the downtown. The second scenario—which mimics evening commute—disables the drop-off walking trip leg but keeps the pickup walking trip leg that now occurs in the downtown; the origin and destination are reversed for each request, compared to the morning commute scenario. The results of enabling both pickup and drop-off walking trip legs (Base Case) are also presented for comparison purposes. In recognition of the importance of the time for the vehicle to reach the first pickup location (TVRFPL) parameter which is discussed in Section 5.5., the results of two TVRFPL values—0 second and 240 seconds—are presented in Figures 23 and 24, respectively. All results are based on the average of 191 sets of 3-request instances.

Figures 23(a) and 24(a) show that disabling one walking trip leg nearly halve the potential reduction in the vehicle driving time.

Figures 23(b) and 24(b) show that the waiting time for a typical rider remains constant for morning commute, as more walking is allowed. In the morning commute scenario, the vehicle does not need to wait for the traveler to walk to the pickup location, so a typical rider's waiting time is not directly affected by the value of $\theta$. In contrast, both evening commute and the base case permit pickup walking trip leg which may require the vehicle to wait for the traveler, so the effect of the TVRFPL value comes into play. When TVRFPL is zero second, the waiting time shows a steady increasing pattern as $\theta$ increases. When TVRFPL is 240 seconds, the waiting time shows a decreasing pattern until $\theta$ equals to 240 seconds, where it starts increasing as $\theta$ increases. The line corresponding to the base case is above the line corresponding to the evening commute merely reflects the fact that the demands' workplaces are concentrated in the downtown

area while their homes are dispersed in a larger geographic area; picking up all the requests in a geographically more confined area like downtown (evening commute) can be faster than picking up all the requests in a larger geographic area (base case).

Figures 23(c) and 24(c) show that the potential reduction in the in-vehicle travel time of a typical rider is halved by disabling one of the walking trip leg.

Figures 23(d) and 24(d) show that the total trip time for a typical rider increases as $\theta$ increases in the morning commute scenario. Such an increase is the net result of the increased walking time associated with the drop-off walking trip and the reduced in-vehicle travel time. In contrast, the evening commute and the base case have trends that are dependent on the TVRFPL values. A decreasing pattern can be observed when $\theta$ is smaller than the TVRFPL value, and an increasing pattern tends to occur when $\theta$ is larger than the TVRFPL value. Additionally, the evening commute scenario appears to be an attractive option from the riders' perspective. The evening commute scenario shares the base case's advantage of reduced waiting time assuming a sufficiently large TVRFPL value, and it shares the morning commute's advantage of halving the walking time which is a major contributor to the total trip time. The net effect is a larger reduction in the total trip time as more walking is allowed, compared to the base case.

Figures 23(e) and 24(e) show that the evening commute scenario is characterized by increasing vehicle service time for at least a portion of values of $\theta$, particularly when the $\theta$ becomes larger and the TVRFPL value is small. This observation is reasonably. In the evening commute scenario, the vehicle may have to wait for the traveler to walk to the pickup location especially when the TVRFPL value is small, and the waiting may lengthen the vehicle service time. The vehicle also has to offer a door-to-door delivery service which makes the benefit associated with walking on drivers impossible to be realized for the drop-off process.

Figures 23(f) and 24(f) show that the base case and the evening commute case are characterized by increasing vehicle waiting time when $\theta$ is greater than the value of TVRFPL. This increasing pattern confirms that the vehicle may have to wait for the traveler to walk to the pickup location.

Overall, the base case, the morning commute scenario, and the evening commute scenario all demonstrate the capability to reduce the vehicle driving time in the congested downtown area. Assuming three riders are pooled in one vehicle, approximately 200 seconds vehicle driving time (2600 feet) reduction can be achieved with a walking time (distance) of 240 seconds (720 feet) in the downtown area. Enabling extra walking in non-downtown areas (the base case) may further reduce the vehicle driving time by approximately 150 seconds (1950 feet). Nevertheless, more than half of driving time and distance reduction occurs in the downtown, which has a denser network and one-way restriction. Therefore, even if walking is enabled solely in the downtown area for the ridesharing service, the goal of reducing downtown traffic congestion can still be adequately achieved, and the potential reduction in VHT or VMT would not be severely compromised.

A ridesharing service that enables both walking trip legs has many advantages for commuting trips from non-downtown to downtown area. The morning commute scenario (only drop-off walking is enabled) is often associated with an increased total trip time compared to a door-to-door service, but the base case (both walking trip legs are enabled) demonstrates the potential to reduce the total trip time of riders compared to a door-to-door service, assuming that the vehicle's initial location is a few minutes away from the first pickup location. Additionally, a larger reduction in vehicle driving time and vehicle service time can be achieved by enabling both walking trip legs instead of just one walking trip leg. The potential increase in vehicle waiting time in the base case—which requires vehicles to use the curbside—might not cause serious curbside management problems since the extra vehicle waiting would occur in the non-downtown area. The only significant drawback of promoting a ridesharing service that enables both walking trip legs compared with just one in the morning commute is that the travelers may have to walk for a few more minutes.

A ridesharing service that enables only the pickup walking trip leg is rider-friendly for the commuting trips from downtown to non-downtown (evening commute scenario). It not only reduces the burden of walking for travelers but also has the potential to reduce the total trip time of riders compared to a door-to-door service. Even if such a service slightly increases the total trip time compared to a door-to-door service,

the amount of increase is likely to be less than the amount of increase associated with the ridesharing service that enables both walking trip legs. However, such a service has two potential disadvantages compared to the ridesharing service with both walking trip legs enabled, including less reduction or even increase in the vehicle service time (driver-unfriendly) and less reduction in VHT or VMT. Additionally, promoting a walking-enabled ridesharing service in downtown area may require extra policy guidance on curbside management since vehicles may have to wait for travelers along the curbside.

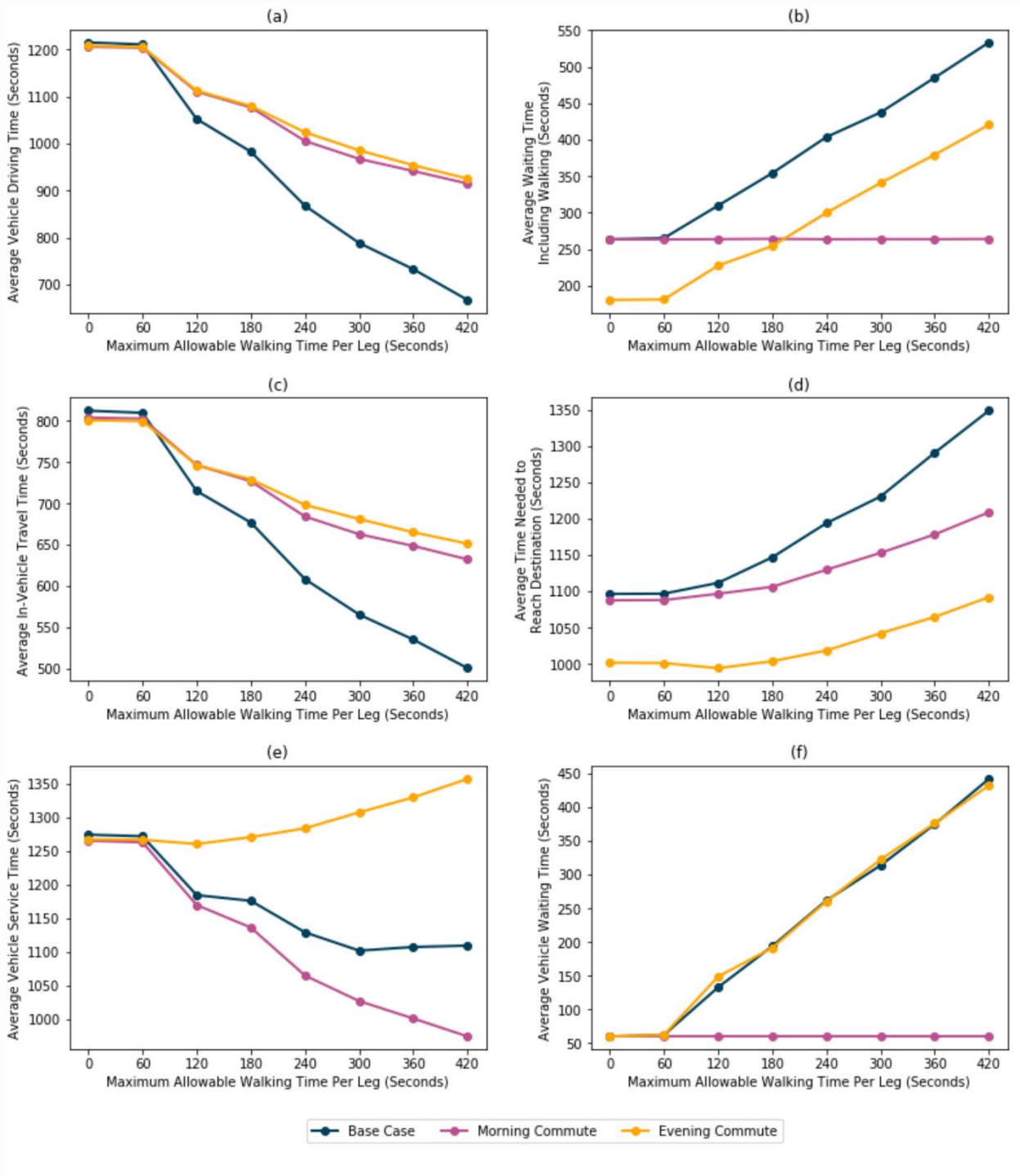


**Figure 23 Results of enabling both walking trip legs, only drop-off walking trip leg, and only pickup walking trip leg (with origin and destination reversed) in a ridesharing service; TVRFPL=0 s.**

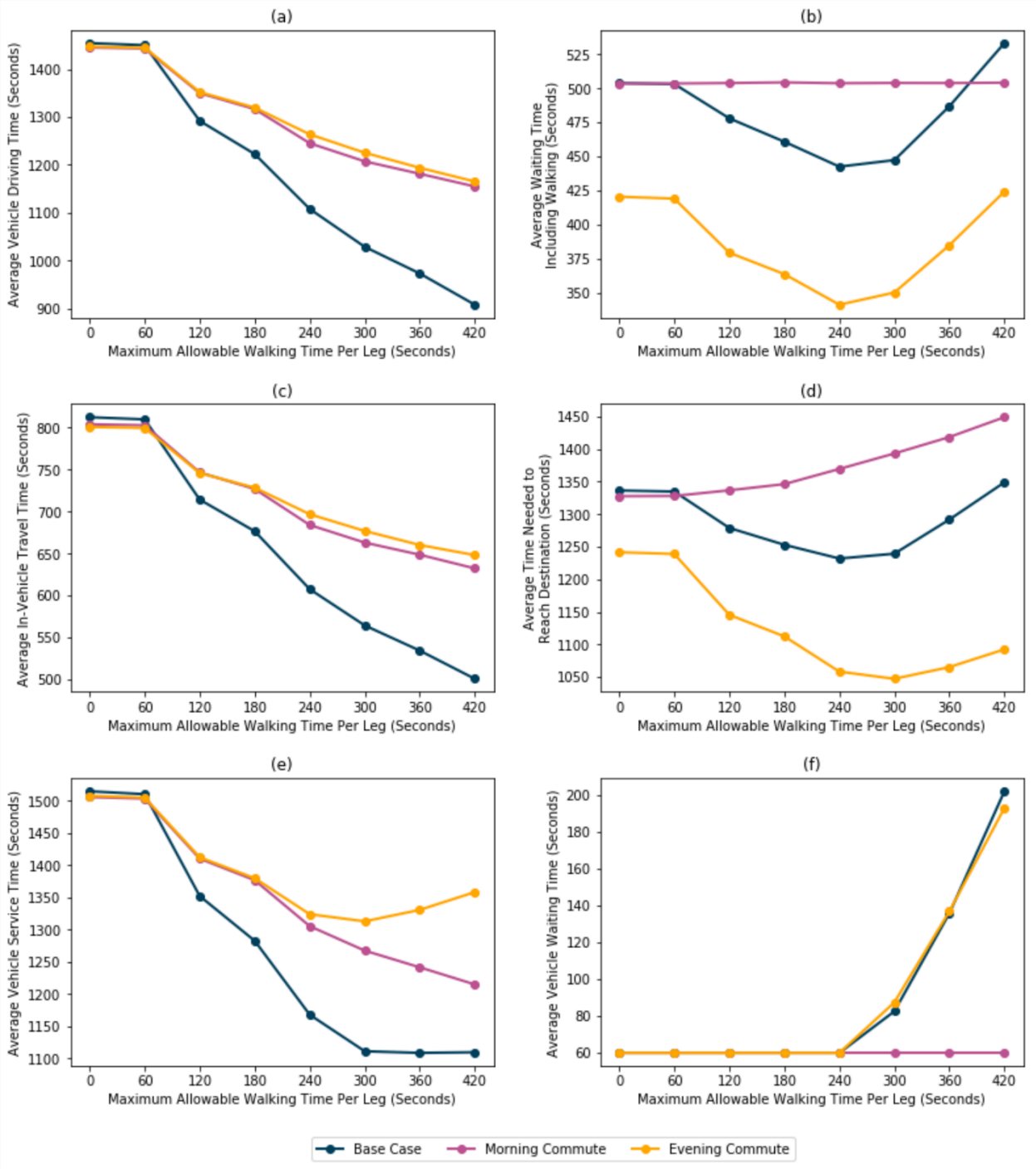


**Figure 24 Results of enabling both walking trip legs, only drop-off walking trip leg, and only pickup walking trip leg (with origin and destination reversed) in a ridesharing service; TVRFPL=240 s.**

# 6 Discussion

Since the primary objective of the proposed ILP minimizes the vehicle driving time, the results guarantee that the vehicle driving time—which can be translated into VHT—shows a monotonic decreasing pattern as more walking is allowed. This decreasing VHT is related to a transformation of the vehicle route, typically from more circuitous to more direct. As more walking is allowed, the vehicle route tends to involve much fewer left and right turns to different streets and less travel on those streets; travelers' walking trip legs tend to be on the streets that are perpendicular to the street that the vehicle is on, assuming a grid network. On a network with one-way streets, a vehicle may need to go around a city block to pick up or drop off a traveler in a conventional door-to-door ridesharing service, but a walking-enabled ridesharing service can eliminate the occurrence of such an undesired vehicle detour. Figures 4, 5, 14, and 15 illustrate how vehicle routes may be transformed as more walking is allowed.

The reductions in vehicle driving time are more prominent when more travelers are pooled into one vehicle. Given the same value of maximum allowable walking time, a larger reduction in the vehicle driving time can be observed for a typical 4-request instance than a typical 3-request or 2-request instance. In other words, as the maximum allowable walking time increases, the vehicle driving time decreases much more rapidly for a typical 4-request instance. Such an observation is not surprising. Without enabling walking, the vehicle may have to make more detours as the number of requests being pooled in one vehicle becomes larger. Hence, enabling walking can be crucial for a high-occupancy ridesharing service.

The monotonic decreasing pattern of vehicle driving time often is associated with, but does not guarantee, a simultaneously decreasing pattern of vehicle driving distance. A road network may have different speed limits for different roads, and minimizing vehicle driving time has the potential bias of choosing a road that has a higher speed limit. In some extreme cases, a route that is suggested by a walking-enabled ridesharing service may have a lower vehicle driving time but a higher vehicle driving distance compared to the route suggested by a door-to-door ridesharing service. A good example illustrating this situation is the illustrative example presented in Case Study I, particularly Figures 4(b)(f)(j) and Figures 5(a)(d)(g). Nevertheless, the occurrence of a suggested route with an increased vehicle driving distance

compared to a door-to-door service is rare and often poses no significant impact on the relevant summary statistics shown in Table 10.

**Table 10 The average vehicle driving time and driving distance reduction compared to a door-to-door ridesharing service**

| # Requests | Maximum Allowable Walking Distance/Time Per Walking Trip Leg | | | | | |
|---|---|---|---|---|---|---|
| | 360 feet/111 meters/120 seconds | | 720 feet/219 meters/240 seconds | | 1080 feet/329 meters/360 seconds | |
| | Case Study I | Case Study II | Case Study I | Case Study II | Case Study I | Case Study II |
| 2 | 75 s (-334 ft*) | 115 s (1,450 ft) | 173 s (2,526 ft) | 241 s (3,086 ft) | 198 s (3,264 ft) | 335 s (4,319 ft) |
| 3 | 132 s (335 ft) | 161 s (2,091 ft) | 251 s (3,613 ft) | 344 s (4,471 ft) | 286 s (4,362 ft) | 478 s (6,215 ft) |
| 4 | 139 s (316 ft) | 207 s (2,658 ft) | 364 s (5,003 ft) | 449 s (5,804 ft) | 412 s (6,163 ft) | 613 s (7,930 ft) |

*: The negative sign indicates a driving distance increase.

The most obvious trade-off to achieve the VHT or VMT reduction by a walking-enabled ridesharing service is that travelers may have to walk for a short distance. The total walking time of all travelers being pooled in a vehicle, as well as a typical (indicated by average or median) rider's walking time, tends to show a steady increasing pattern as more walking is allowed. Nevertheless, this does not imply that every traveler being pooled in a vehicle would experience an increase in walking time, as more walking is allowed. For example, Figure 4 of Case Study I illustrates how different travelers may be required to walk as the maximum allowable walking time parameter value changes. Additionally, some riders may be assigned to walk for a shorter distance than others, and the actual amount of walking may be disproportionately borne by a subset of riders sharing a vehicle. Those observations may have implication on how to set the fare for each traveler. For example, the fare for a rider may need to be reduced based on the actual distance that the rider is required to walk.

Although enabling a longer walking distance tends to bring a larger amount of reduction in VHT, the driving time reduction per second of walking (DTRPSW) tends to reach the maximum when the maximum allowable walking time per leg is set to two to three minutes (equivalently, 360 to 540 feet of walking distance). The maximum DTRPSW generally is less than 0.45, which implies one second of actual walking can reduce 0.45 second of vehicle driving time. Nevertheless, the vehicle driving time is highly dependent on the speed on a road segment, and a higher speed can lower the DTRPSW. An additional measure—the average vehicle driving distance reduction per foot of walking—is presented in Figure 25. In general, this additional measure tends to be high when the maximum allowable walking distance per leg is no longer

than 540 feet (three minutes), except for the Isla Vista Case Study (the model prescribes a vehicle route that is faster but longer in the drop-off region when 360 feet of walking is allowed per walking trip leg). When 540 feet of walking is allowed, each foot of actual walking leads to slightly less than 1.75 feet reduction in vehicle driving distance.

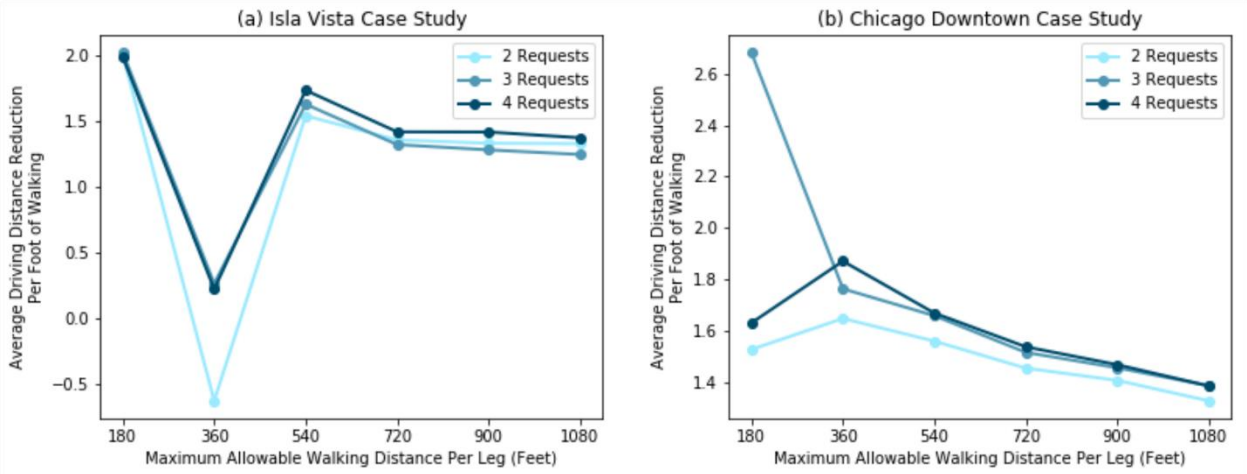


**Figure 25 The average driving distance reduction (compared to a door-to-door service) per foot of actual walking.**

Without explicitly minimizing the travelers' total trip time, travelers using a walking-enabled ridesharing service may still be able to reach their destinations earlier than using a door-to-door service. The outcome of whether the travelers' total trip time would decrease depends on two competing forces. The first force decreases the total trip time, and it mainly involves the in-vehicle travel time, which shows a monotonic decreasing pattern as more walking is allowed for an average rider. The second force increases the total trip time, and it mainly involves the drop-off walking time. Figure 26 shows that the reduction in the in-vehicle travel time may not be able to fully offset the increase in the total trip time caused by the drop-off walking, especially when the number of requests being pooled in one vehicle is only two. Furthermore, if a vehicle that is very close to the request's origin location is assigned to serve the request, an increase in the total trip time relative to a door-to-door service can be expected since the vehicle needs to wait for the traveler to walk to the pickup location.

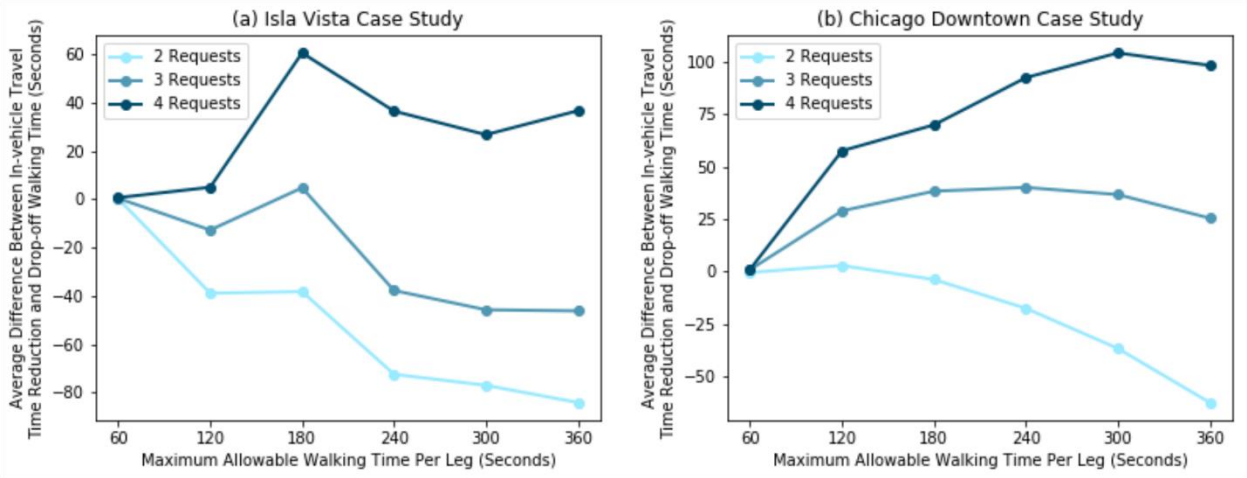


**Figure 26 The average difference between in-vehicle travel time and the drop-off walking time. A non-negative value indicates the reduction of the in-vehicle travel time may offset the increased drop-off walking time**

This study also suggests that enabling both pickup and drop-off walking trip legs would be a preferable option for both travelers and drivers in a morning commute scenario in which trips tend to go from non-downtown areas to the downtown area, compared with enabling only the drop-off walking trip leg. This study also suggests that enabling only pickup walking trip leg would be a preferable option for travelers but not drivers in an evening commute scenario in which trips tend to go from the downtown to non-downtown. Nevertheless, appropriate curbside management strategy might be needed in the downtown especially for the evening commute scenario.

# 7 Conclusion

Allowing travelers to walk for a short distance to be picked up and walk for a short distance to their destinations in a ridesharing service has received a lot of attentions in recent years. Such a service differs from the conventional door-to-door ridesharing service, as it involves introducing alternative PUDO locations into the decision space.

This walking-enabled ridesharing service has perceived benefits of reduction in VMT and increase in matching rate, which are widely confirmed by simulation studies in recent literature. Many studies rely on methods that first group requests with similar itineraries and assign one common PUDO location to this group of requests. Those studies tend to coin the problem with the term meeting points. Some methods used in those literature assume one intermediate pickup and one intermediate drop-off location along the vehicle

route, and some methods relax such an assumption but still heavily focuses on prescribing common PUDO locations. There are several studies that take a different approach to tackle the problem. Assigning a common PUDO location to multiple travelers is not an essential step in such an approach, and the methods associated with such an approach have a root in the GTSP. Using such an approach, it is spontaneous for a location where multiple travelers are simultaneously picked up or dropped off to occur.

This thesis examines the potential for a walking-enabled ridesharing service, using a non-meeting points approach. A novel integer-linear program is used to prescribe the vehicle route incorporating alternative PUDO locations. The proposed model aims to reduce the detours that may occur during the PUDO process by minimizing total vehicle driving time. A secondary objective which minimizes the walking time is also added. The model includes implicit and explicit constraints on the walking time tolerable by travelers, pickup time windows, latest arrival time constraints, and capacity constraints. The model is applied to two case studies.

The two case studies confirm that a walking-enabled ridesharing service with a non-meeting points approach can substantially reduce the vehicle driving time compared to a door-to-door ridesharing service in a static and deterministic simulation environment, and this finding is consistent with the existing literature. The amount of reduction in vehicle driving time due to enabling walking tends to increase as more travelers are pooled in the same vehicle; this observation is intuitive and can be related to the public transit system that also offers a non-door-to-door service. Nevertheless, solely focusing on reducing vehicle driving time may lead to slight delays in travelers' arrival time at their destinations compared with using a door-to-door service, especially if the number of travelers being pooled in one vehicle is only two or three and the drop-off walking trip legs are enabled.

One critical limitation of this study is the use of the dummy depot as the vehicle's final destination and possibly initial location. As a result of using the dummy depot, the vehicle would always drop off the last passenger as soon as it reaches the walking range of the last passenger's destination. Additionally, in the Chicago case study, the first traveler being picked by the vehicle would always have to walk to the boundary of the origin's walking range. Therefore, the walking time of the first traveler being picked up or

the last traveler being dropped off may show a steady increasing pattern as the maximum allowable walking time increases. Indirect consequences may include an under-estimate of the vehicle driving time and traveler in-vehicle time, an over-estimate of the waiting time that includes pickup walking and the total trip time. Nevertheless, without the use of dummy depot, it is difficult to set the initial vehicle location in the Chicago case study, and it is difficult to set the destination of the vehicle after serving all the requests.

Finally, this study addresses a static and deterministic problem, in which traveler's demand is known in advance. However, given the proliferation of on-demand ride-sharing and shared-ride mobility services, an important extension involves incorporating walking into an on-demand ride-sharing service as well as modeling and solving the associated stochastic and dynamic routing problem. One way to do this is through incorporating walking trip legs into a dynamic ride-sharing scheme that takes the advantage of an unimodular constraint matrix in solving a request-vehicle assignment problem, such as those proposed by Hyland and Mahmassani (2020) and Simonetto et al. (2019).

# Reference

Aissat, K., Oulamara, A., 2014. A Priori Approach Of Real-Time Ridesharing Problem With Intermediate Meeting Locations. J. Artif. Intell. Soft Comput. Res. 4. https://doi.org/10.1515/jaiscr-2015-0015

Alisoltani, N., Leclercq, L., Zargayouna, M., 2021. Can dynamic ride-sharing reduce traffic congestion? Transp. Res. Part B Methodol. 145, 212–246. https://doi.org/https://doi.org/10.1016/j.trb.2021.01.004

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., Rus, D., 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proc. Natl. Acad. Sci. 114, 462 LP – 467. https://doi.org/10.1073/pnas.1611675114

Araldo, A., Maria, A.D., Stefano, A.D., Morana, G., 2019. On the Importance of demand Consolidation in Mobility on Demand, in: 2019 IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT). pp. 1–8. https://doi.org/10.1109/DS-RT47707.2019.8958669

Balardino, A.F., Santos, A.G., 2016. Heuristic and Exact Approach for the Close Enough Ridematching Problem BT - Hybrid Intelligent Systems, in: Abraham, A., Han, S.Y., Al-Sharhan, S.A., Liu, H. (Eds.), . Springer International Publishing, Cham, pp. 281–293.

City of Chicago: Data Portal, 2020. Street Center Lines [WWW Document]. URL https://data.cityofchicago.org/Transportation/Street-Center-Lines/6imu-meau

City of Chicago: Data Portal, 2019. Transportation Network Providers - Trips [WWW Document]. URL https://data.cityofchicago.org/Transportation/Transportation-Network-Providers-Trips-2019/iu3g-qa69

Cordeau, J.-F., Laporte, G., 2007. The Dial-a-ride Problem: Models and Algorithms. Ann. Oper. Res. 153, 29–46. https://doi.org/10.1007/s10479-007-0170-8

Current, J., ReVelle, C., Cohon, J., 1984. SYMPOSIUM ON LOCATION PROBLEMS: IN MEMORY OF LEON COOPER. J. Reg. Sci. 24, 161–183. https://doi.org/10.1111/j.1467-9787.1984.tb01030.x

Czioska, P., Kutadinata, R., Trifunović, A., Winter, S., Sester, M., Friedrich, B., 2019. Real-world

Meeting Points for Shared Demand-Responsive Transportation Systems. Public Transp. https://doi.org/10.1007/s12469-019-00207-y

Czioska, P., Mattfeld, D.C., Sester, M., 2017a. GIS-based identification and assessment of suitable meeting point locations for ride-sharing. Transp. Res. Procedia 22, 314–324. https://doi.org/https://doi.org/10.1016/j.trpro.2017.03.038

Czioska, P., Trifunović, A., Dennisen, S., Sester, M., 2017b. Location- and time-dependent meeting point recommendations for shared interurban rides. J. Locat. Based Serv. 11, 181–203. https://doi.org/10.1080/17489725.2017.1421779

Dantzig, G., Fulkerson, R., Johnson, S., 1954. Solution of a Large-Scale Traveling-Salesman Problem. J. Oper. Res. Soc. Am. 2, 393–410.

Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M., Soumis, F., 2002. VRP with Pickup and Delivery, in: The Vehicle Routing Problem. pp. 225–242. https://doi.org/10.1137/1.9780898718515.ch9

Dumas, Y., Desrosiers, J., Soumis, F., 1991. The Pickup and Delivery Problem with Time Windows. Eur. J. Oper. Res. 54, 7–22. https://doi.org/https://doi.org/10.1016/0377-2217(91)90319-Q

Fielbaum, A., Bai, X., Alonso-Mora, J., 2021. On-demand ridesharing with optimized pick-up and drop-off walking locations. Transp. Res. Part C Emerg. Technol. 126, 103061. https://doi.org/https://doi.org/10.1016/j.trc.2021.103061

Gökay, S., Heuvels, A., Krempels, K.H., 2019. On-demand ride-sharing services with meeting points. VEHITS 2019 - Proc. 5th Int. Conf. Veh. Technol. Intell. Transp. Syst. 5, 117–125. https://doi.org/10.5220/0007709101170125

Gurumurthy, K.M., Kockelman, K., 2020. How Much Does Greater Trip Demand and Aggregation at Stops Improve Dynamic Ride- Sharing in Shared Autonomous Vehicle Systems?

Hartman, I.B.A., Keren, D., Dbai, A.A., Cohen, E., Knapen, L., Yasar, A.U.H., Janssens, D., 2014. Theory and Practice in Large Carpooling Problems. Procedia Comput. Sci. 32, 339–347. https://doi.org/10.1016/j.procs.2014.05.433

Hyland, M., Mahmassani, H.S., 2020. Operational benefits and challenges of shared-ride automated mobility-on-demand services. Transp. Res. Part A Policy Pract. 134, 251–270. https://doi.org/https://doi.org/10.1016/j.tra.2020.02.017

Kaan, L., Olinick, E., 2013. The Vanpool Assignment Problem: Optimization models and solution algorithms. Comput. Ind. Eng. 66, 24–40. https://doi.org/10.1016/j.cie.2013.05.020

Laporte, G., Mercure, H., Nobert, Y., 1987. Generalized travelling salesman problem through n sets of nodes: the asymmetrical case. Discret. Appl. Math. 18, 185–197. https://doi.org/https://doi.org/10.1016/0166-218X(87)90020-5

Li, N., Kong, L., Shu, W., Wu, M.Y., 2020. Benefits of Short-Distance Walking and Fast-Route Scheduling in Public Vehicle Service. IEEE Trans. Intell. Transp. Syst. 21, 3706–3717. https://doi.org/10.1109/TITS.2019.2931798

Li, X., Hu, S., Fan, W., Deng, K., 2018. Modeling an Enhanced Ridesharing System with Meet Points and Time Windows. PLoS One 13, 1–19. https://doi.org/10.1371/journal.pone.0195927

Lo, J., Morseman, S., 2018. The Perfect uberPOOL: A Case Study on Trade-Offs. Ethnogr. Prax. Ind. Conf. Proc. 2018, 195–223. https://doi.org/10.1111/1559-8918.2018.01204

Lyu, Y., Lee, V.C.S., Ng, J.K., Lim, B.Y., Liu, K., Chen, C., 2019. Flexi-Sharing: A Flexible and Personalized Taxi-Sharing System. IEEE Trans. Veh. Technol. 68, 9399–9413. https://doi.org/10.1109/TVT.2019.2932869

Martínez, L.M., Viegas, J.M., Eiró, T., 2014. Formulating a New Express Minibus Service Design Problem as a Clustering Problem. Transp. Sci. 49, 85–98. https://doi.org/10.1287/trsc.2013.0497

McCubbin, D.R., Delucchi, M.A., 1999. The Health Costs of Motor-Vehicle-Related Air Pollution. J. Transp. Econ. Policy 33, 253–286.

Miklas-Kalczynska, M., Kalczynski, P., 2020. Self-organized carpools with meeting points. Int. J. Sustain. Transp. 1–12. https://doi.org/10.1080/15568318.2019.1711468

Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer Programming Formulation of Traveling Salesman Problems. J. ACM 7, 326–329. https://doi.org/10.1145/321043.321046

Mourad, A., Puchinger, J., Chu, C., 2019. A Survey of Models and Algorithms for Optimizing Shared Mobility. Transp. Res. Part B Methodol. 123, 323–346. https://doi.org/10.1016/j.trb.2019.02.003

Niblett, T., Church, R., 2014. The Shortest Covering Path Problem: A New Perspective and Model. Int. Reg. Sci. Rev. 39. https://doi.org/10.1177/0160017614550082

Rice, M.N., Tsotras, V.J., 2013. Parameterized Algorithms for Generalized Traveling Salesman Problems in Road Networks, in: Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL'13. Association for Computing Machinery, New York, NY, USA, pp. 114–123. https://doi.org/10.1145/2525314.2525342

Simonetto, A., Monteil, J., Gambella, C., 2019. Real-time city-scale ridesharing via linear assignment problems. Transp. Res. Part C Emerg. Technol. 101, 208–232. https://doi.org/https://doi.org/10.1016/j.trc.2019.01.019

Sol, M., Savelsbergh, M., 1995. The General Pickup and Delivery Problem. Transp. Sci. 29, 17–29.

Stiglic, M., Agatz, N., Savelsbergh, M., Gradisar, M., 2015. The Benefits of Meeting Points in Ride-Sharing Systems. Transp. Res. Part B Methodol. 82, 36–53. https://doi.org/10.1016/j.trb.2015.07.025

Toth, P., Vigo, D., 2002. The Vehicle Routing Problem, Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics. https://doi.org/doi:10.1137/1.9780898718515

Zhang, C., Yang, F., Ke, X., Liu, Z., Yuan, C., 2019. Predictive modeling of energy consumption and greenhouse gas emissions from autonomous electric vehicle operations. Appl. Energy 254, 113597. https://doi.org/https://doi.org/10.1016/j.apenergy.2019.113597

Zhang, K., Batterman, S., 2013. Air pollution and health risks due to vehicle traffic. Sci. Total Environ. 450–451, 307–316. https://doi.org/https://doi.org/10.1016/j.scitotenv.2013.01.074

Zhao, M., Yin, J., An, S., Wang, J., Feng, D., 2018. Ridesharing Problem with Flexible Pickup and Delivery Locations for App-Based Transportation Service: Mathematical Modeling and Decomposition Methods. J. Adv. Transp. 2018. https://doi.org/10.1155/2018/6430950

Zheng, Y., Li, W., Qiu, F., Wei, H., 2019. The benefits of introducing meeting points into flex-route

transit services. Transp. Res. Part C Emerg. Technol. 106, 98–112. https://doi.org/https://doi.org/10.1016/j.trc.2019.07.012

Zhou, M., Jin, H., Wang, W., 2016. A review of vehicle fuel consumption models to evaluate eco-driving and eco-routing. Transp. Res. Part D Transp. Environ. 49, 203–218. https://doi.org/https://doi.org/10.1016/j.trd.2016.09.008