UNIVERSITY OF LJUBLJANA
FACULTY OF ECONOMICS

MITJA ŠTIGLIC

MODELS AND SOLUTION ALGORITHMS FOR THE RIDE-MATCHING PROBLEM: FACILITATING THE MATCHING PROCESS IN RIDESHARING SYSTEMS

DOCTORAL DISSERTATION

LJUBLJANA, 2017

AUTHORSHIP STATEMENT

The undersigned Mitja Štiglic, a student at the University of Ljubljana, Faculty of Economics, (hereafter: FELU), declare that I am the author of the doctoral dissertation entitled Models and Solution Algorithms for the Ride-matching Problem: Facilitating the Matching Process in Ridesharing Systems, prepared under the supervision of prof. dr. Mirko Gradišar and co-supervision of Dr. Niels Agatz

DECLARE

1. this doctoral dissertation to be based on the results of my own research;
2. the printed form of this doctoral dissertation to be identical to its electronic form;
3. the text of this doctoral dissertation to be language-edited and technically in adherence with the FELU's Technical Guidelines for Written Works, which means that I cited and/or quoted works and opinions of other authors in this doctoral dissertation in accordance with the FELU's Technical Guidelines for Written Works;
4. to be aware of the fact that plagiarism (in written or graphical form) is a criminal offence and can be prosecuted in accordance with the Criminal Code of the Republic of Slovenia;
5. to be aware of the consequences a proven plagiarism charge based on this doctoral dissertation could have for my status at the FELU in accordance with the relevant FELU Rules;
6. to have obtained all the necessary permissions to use the data and works of other authors which are (in written or graphical form) referred to in this doctoral dissertation and to have clearly marked them;
7. to have acted in accordance with ethical principles during the preparation of this doctoral dissertation and to have, where necessary, obtained permission of the Ethics Committee;
8. my consent to use the electronic form of this doctoral dissertation for the detection of content similarity with other written works, using similarity detection software that is connected with the FELU Study Information System;
9. to transfer to the University of Ljubljana free of charge, non-exclusively, geographically and time-wise unlimited the right of saving this doctoral dissertation in the electronic form, the right of its reproduction, as well as the right of making this doctoral dissertation publicly available on the World Wide Web via the Repository of the University of Ljubljana;
10. to have acquired from publishers, to whom I have previously exclusively transferred material copyrights for articles, all the necessary permissions for the inclusion of articles in the printed and electronic forms of this dissertation. These permissions enable the University of Ljubljana to save this doctoral dissertation in the electronic form, to reproduce it, as well as to make it publicly available on the World Wide Web via the Repository of the University of Ljubljana free of charge, non-exclusively, geographically and time-wise unlimited;
11. my consent to publication of my personal data that are included in this doctoral dissertation and in this declaration, when this doctoral dissertation is published.

Ljubljana, June 8, 2017 Author's signature: ____________________

Acknowledgements

I would first like to thank my wife Andreja for all of the sacrifices she has made for me and our children in the last few years. Work on this thesis and on the individual research papers would often stretch long into the evenings and would almost never leave my thoughts. I am grateful for her patience and unselfish support. The same goes for my parents and my wife's parents: if I am successful, it is, to an important extent, due to them.

I would like to express my gratitude to my supervisor, prof. dr. Mirko Gradisar for the useful comments and suggestions and for helping me to bring this work to a successful conclusion.

I would also like to thank my co-supervisor Dr. Niels Agatz and de facto co-supervisor Dr. Martin Savelsbergh for steering my work on the individual implementations and papers, for the hundreds of detailed e-mails we have exchanged and for the excellent advice they have given. I am very proud of the work we have accomplished together. Our collaboration was excellent.

Moreover, I would also like to express my gratitude to Prof. Dr. Martin Josef Geiger and prof. dr. Peter Trkman for their advice in the initial phase of my doctoral studies and for helping me shape my research proposal.

Many thanks also goes to other colleagues at the Academic Unit for Business Informatics and Logistics of the Faculty of Economics, University of Ljubljana: although my research was quite independent of the work of the rest of the group I have nevertheless received a lot of useful advice on how to do research and publish scientific results in academic journals.

Thanks are also due to my former colleagues at the Transport Economics Centre at the Faculty of Civil Engineering, University of Maribor: in many ways, this research is a continuation of the collaborative logistics work we were doing within the scope of European research projects. It was through our collaboration that I initially developed many of the ideas that were finally materialized in this thesis.

Finally, in addition I would like to thank the Slovenian Research Agency and the Faculty of Economics, University of Ljubljana for funding and supporting my research.

MODELS AND SOLUTION ALGORITHMS FOR THE RIDE-MATCHING PROBLEM: FACILITATING THE MATCHING PROCESS IN RIDESHARING SYSTEMS

ABSTRACT

In ridesharing, individuals with matching itineraries and schedules share a ride in a personal vehicle. The driver and rider(s) typically share the associated costs so that each benefits from the shared ride. In addition, drivers may save time because they are able to use high-occupancy vehicle lanes reserved for the exclusive use of vehicles with two or more occupants, while riders may appreciate that they do not need to drive or even own a vehicle.

Ridesharing can significantly reduce the number of cars needed to satisfy the mobility needs of participants and, thus, reduce congestion and other externalities related to heavy traffic when people rely on individual transportation to satisfy their mobility needs. Such challenges arise in a myriad of urban areas around the world. Ridesharing appears as an interesting measure since it may result in significant effects without large investments.

Ridesharing services on the market range from simple online bulletin boards to complex systems that can be accessed through web and mobile applications offering automated matching, routing, and payment. In this thesis, we focus on systems that offer automated matching of drivers and riders within an urban area. The service provider receives a large number of rideshare offers and requests from its users. Riders looking for rideshare opportunities need to be matched with drivers who are offering rides and the resulting trips need to be scheduled. Time windows and other restrictions imposed by the system or the users need to be respected.

In ridesharing, each driver has a specific itinerary and is willing to pick up and drop off riders en route. To accommodate the riders, the driver has to make a detour and make extra stops. The length of the detour and the number of extra stops depend on the driver's willingness to extend his trip time. Limited flexibility in system participants' itineraries and schedules is a major challenge in ridesharing. It may result in many drivers and riders not finding a match.

In the first chapter of the thesis, we conduct a study to quantify the impact of different types of participants' flexibility on the performance of a ridesharing system. Our results consistently show that small increases in flexibility, e.g. in terms of desired departure time or maximum detour time, can significantly increase the expected matching rate, especially when the number of trip announcements in the system is small. The results clearly demonstrate the impact of participant flexibility on the performance of a ridesharing system (in terms of the matching rate achieved). In order for dynamic ridesharing to work, drivers and riders need to be flexible in their departure and arrival times but, most importantly, drivers need to be flexible in terms of the detour they are willing to make. The main contributions are that we introduce and define three different types of participant flexibility that are relevant in the dynamic ridesharing context, i.e. matching flexibility, detour flexibility, and scheduling flexibility. We quantify the impact of these types of flexibility on system performance and investigate the level of additional flexibility that is required to improve the effectiveness of a rideshare system.

In the second chapter of the thesis, we investigate the potential benefits of introducing meeting points in a ridesharing system. With meeting points, riders can be picked up and

dropped off either at their origin and destination or at a meeting point that is within a certain distance from their origin or destination. The increased flexibility results in additional feasible matches between drivers and riders, and allows a driver to be matched with multiple riders without increasing the number of stops the driver needs to make. We design and implement an algorithm that optimally matches drivers and riders in large-scale ridesharing systems with meeting points. We perform an extensive simulation study to assess the benefits of meeting points. We show that the introduction of meeting points in a ridesharing system can substantially improve a number of critical performance metrics, i.e. percentage of matched riders, percentage of matched participants, and mileage savings.

In the third chapter, we investigate the potential of integrating mass transit and ridesharing to offer fast and affordable transfers to/from transit stations to a large number of residents in suburban areas. We examine some of the options of implementing such a system and the associated ride-matching problems and evaluate the possible synergies that can arise as a result of integration by performing a large number of computational experiments. Our simulation framework captures the main characteristics of many real-world transit settings and quantifies the benefits of integrating ridesharing and public transit. The results show that the integration of a ridesharing system and a public transit system can significantly enhance mobility, increase the use of public transport, and reduce the negative externalities associated with car travel. We found that driver willingness to pick up and drop off more than one rider is critical to the system's performance.

In chapter four, we study how to identify feasible driver-rider matches more efficiently. This can have an important effect on the total runtime of the algorithm because, typically, only a very small fraction of the possible matches are feasible, meaning it is possible to do much better if we do not have to fully evaluate all pairs. We exploit two ideas: (1) direct drive times from origin to destination can be used to efficiently identify those riders who have sufficiently small drive times to be matched with a particular driver; and (2) rider time windows can be stored in a memory structure that allows one to find riders with time windows that overlap with the time window of a driver in sub-linear time. We develop and test a data structure that combines optimizations (1) and (2) and test its performance.

In chapter five, we present a new mixed-integer linear programming model for the single driver–multiple rider matching problem that arises in certain types of ridesharing systems. The model we devise allows users to opt for driver or rider roles or, alternatively, to let the model determine what is best. If there is no match for a rider or a driver, the model is capable of adding this opportunity cost to the objective function value. The model minimizes the cost of all the trips that have to be performed to move the users from their origin to their destination. We perform simulations on different instances constructed based on ridesharing practice between Slovenia's two largest cities and comment on the results.

We hope that the insights generated by this thesis will constitute a valuable contribution to the extant body of knowledge on ridesharing systems and their operation and that they will inform ridesharing system providers on how to design applications, matching algorithms and incentive schemes as well as form alliances with other systems such as public transport agencies and bike-sharing system providers.

Keywords: Ridesharing; Sustainable mobility; Discrete optimization; Simulation

MODELI IN ALGORITMI ZA OBLIKOVANJE PREVOZNIH SKUPIN V SISTEMIH DELJENJA PREVOZA: LAJŠANJE PROCESA ISKANJA UJEMANJ

IZVLEČEK

Deljenje prevoza je družbeni pojav, pri katerem si posamezniki z ujemajočimi se prevoznimi potrebami delijo eno vozilo in celotno pot ali nekatere dele svojih poti opravijo skupaj. Voznik vozila ima določeno pot in je v zameno za povračilo stroškov prevoza pripravljen po poti pobirati ter odlagati sopotnike.

Koristi deljenja prevoza za posameznike so nižji stroški transporta, možnost uporabe pasov za vozila z več potniki (t. i. pasovi HOV v ZDA) in manjša utrujenost sopotnikov. V primerih, ko sopotniki nimajo osebnega vozila, lahko govorimo tudi o skrajšanem času potovanja v primerjavi z javnim transportom.

Z deljenjem prevozov se lahko pomembno zmanjša število osebnih vozil, ki so potrebna za zadovoljitev potreb posameznikov po mobilnosti. Zlasti v regijah, v katerih mobilnost posameznikov temelji na uporabi osebnih vozil, je z deljenjem prevozov mogoče znižati število vozil v prometu in s tem razbremeniti prometno omrežje. Deljenje prevozov ima lahko torej pozitivne učinke na promet in okolje v smislu zmanjševanja pojavnosti gneče ter zastojev v prometu.

S pojavom interneta in mobilnih tehnologij so se pojavile raznovrstne, napredne oblike deljenja prevoza. Danes obstaja širok spekter različnih sistemov: od preprostejših e-oglasnih desk za deljenje prevoza do bolj kompleksnih storitev, do katerih se lahko dostopa preko spletnih in mobilnih aplikacij in ki ponujajo avtomatsko tvorjenje skupin, načrtovanje poti ter elektronska plačila. V tej disertaciji se osredotočamo na napredne sisteme, ki omogočajo deljenje prevozov preko pametnih telefonov in drugih sodobnih naprav ter avtomatsko povezujejo voznike in potnike, tj. tvorijo prevozne skupine.

Vsak voznik, ki ponuja prevoz v sistemu za deljenje prevoza, ima določeno pot. Da bi našel sopotnike, s katerimi bi si delil stroške prevoza, je pripravljen narediti nekaj dodatnih postankov, potrebnih za vstop v vozilo in izstop iz njega. Pogosto mora voznik napraviti tudi obvoz in se s tem približati izvoru ter cilju posameznega potnika. Največja dolžina obvoza in število dodatnih postankov sta odvisna od voznikove pripravljenosti, da podaljša potovalni čas. Omejena pripravljenost voznikov za izvedbo dodatnih postankov in daljših obvozov zelo otežuje deljenje prevozov. Če je voznik nefleksibilen, je namreč izredno težko najti ustreznega sopotnika. Posledica tega je lahko, da velikega deleža voznikov in potnikov, ki bi sicer želeli sodelovati v sistemu, ni mogoče povezati.

Ta disertacije je osredotočena na lajšanje procesa iskanja ujemanj v sistemih deljenja prevoza. V ta namen smo zasnovali in preizkusili več različnih modelov, algoritmov ter mehanizmov, ki bi lahko zagotovili visoko verjetnost ujemanja za uporabnike (maksimizacija števila ujemanj). Opravili smo veliko število simulacij, na podlagi katerih je možno sklepati o pomembnih lastnostnih sistemov za deljenje prevoza in tudi o potencialnih učinkih določenih strateških odločitev upravljavca sistema (npr. integracija sistema deljenja prevoza s sistemom javnega transporta ali vključitev nabornih točk v sistem deljenja prevoza).

V prvem poglavju disertacije proučujemo vpliv fleksibilnosti uporabnikov sistema deljenja prevoza na število ujemanj med vozniki in potniki, ki jih je možno vzpostaviti. Ta vpliv smo proučili tako, da smo zgradili posebno simulacijsko okolje, ki simulira sistem deljenja prevoza, v katerem se en voznik lahko poveže z največ enim potnikom. Definirali smo tri

tipe fleksibilnosti, ki igrajo pomembno vlogo pri dinamičnem deljenju prevozov, in skozi veliko število simulacij kvantificirali vpliv posamičnega tipa fleksibilnosti na število možnih ujemanj. Naredili smo tudi simulacijo, pri kateri smo proučili, koliko dodatne fleksibilnosti (in katere) je potrebno, da se odstotek povezanih uporabnikov poveča za določen odstotek.

V drugem poglavju razvijemo nov model za tvorjenje prevoznih skupin, ki vsebuje naborne točke. Do zdaj so modeli v literaturi predpostavljali, da potniki vstopajo in izstopajo v vozilo oz. iz njega na svojih dejanskih izvorih ter ponorih. Kakorkoli, ta predpostavka je precej omejujoča, saj se mora samo ena stran prilagajati drugi. Z vključitvijo standardnih nabornih mest v sistem je omogočeno, da se potnik iz svojega izvora premakne na najugodnejše naborno mesto oz. da se z določene ugodne naborne točke pomakne proti svojemu ponoru z različnimi transportnimi modalitetami. Takšne naborne točke se uporabljajo v veliko sistemih deljenja prevozov, vendar še niso bile omenjene oz. uporabljene v optimizacijski literaturi. Algoritem, ki smo ga zasnovali za rešitev tega problema, skuša optimirati ujemanja z vidika dveh kriterijev: maksimiranje števila uparjenih uporabnikov in maksimiranje števila prihranjenih prevoženih kilometrov v celotnem sistemu.

V tretjem poglavju predstavljamo nov model in algoritem za tvorjenje prevoznih skupin, ki omogoča integracijo s sistemom javnega transporta. Na ta način je omogočeno, da voznik potnika zapelje na njegovo končno destinacijo ali pa ga zapelje na postajo javnega transporta, od koder se z vlakom, avtobusom ali drugim prevoznim sredstvom pelje do svoje končne destinacije. Sistem poskuša sinhronizirati poti voznikov in potnikov z urnikom javnega transporta ter upošteva veliko število omejitev glede preferenc voznikov in potnikov (najhitrejši čas odhoda, najpoznejši čas prihoda, najdaljše trajanje poti ipd.). Podobno kot v prejšnjem poglavju tudi v tem algoritem, ki smo ga zasnovali za rešitev problema, skuša optimirati ujemanja z vidika dveh kriterijev: maksimiranje števila uparjenih uporabnikov in maksimiranje števila prihranjenih prevoženih kilometrov v celotnem sistemu.

V četrtem poglavju predstavljamo krajšo metodološko diskusijo na temo generiranja vseh dopustnih ujemanj med vozniki in potniki. Predstavljamo, kako je mogoče izkoristiti lastnosti problema za izboljšanje učinkovitosti metode generiranja vseh dopustnih ujemanj. Informacije o najkrajšem možnem trajanju poti posamičnega potnika in voznika izkoristimo pri grajenju podatkovne strukture, katere namen je omogočiti učinkovito poizvedbo o tem, kateri potniki so potencialno dopustni za dotičnega voznika.

V zadnjem, petem poglavju predstavimo matematični model za tvorjenje prevoznih skupin v sistemih za deljenje prevoza, ki omogoča, da posamičnega voznika povežemo z večjim številom potnikov. Gre za model, ki je primeren za deljenje prevoza med večjimi mesti. Model tudi dopušča, da se uporabnik ne opredeli glede vloge in mu jo sistem sam določi na podlagi razmerja med ponudbo ter povpraševanjem. Predstavimo tudi rutino za predprocesiranje problema. Izvedemo nekaj manjših simulacij.

Upamo, da bodo metode, dognanja in rezultati, predstavljeni v tej disertaciji, koristno prispevali k boljšemu razumevanju, upravljanju ter delovanju sistemov za deljenje prevoza. Zlasti upamo, da bodo rezultati simulacij upravljavce sistemov za deljenje prevoza spodbudili k uvedbi raznovrstnih spodbud oz. motivacijskih shem za voznike in potnike ter k proučitvi možnosti integracije z javnim transportom in drugimi sistemi, kot so npr. sistemi deljenja koles.

Ključne besede: deljenje prevozov; trajnostna mobilnost; diskretna optimizacija; simulacija.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Background

Ridesharing refers to collaborative transportation in which individuals with matching itineraries and schedules as well as other preferences ride together so as to save costs (Furuhata et al., 2013). The driver of the car has a specific itinerary and is willing to pick up and drop off other passengers (riders) en route in exchange for remuneration. The driver's route may be fixed or may be adapted to the pickup and drop-off locations of riders. Apart from cost reimbursement, other arrangements such as taking turns driving are also possible.

Ridesharing is also known as: carpooling, car-sharing, ride-sharing, lift-sharing, or covoiturage. Some terms tend to be used for specific arrangements, e.g. carpooling, but the literature and practice is inconsistent on this. We will use the general term ridesharing and study models relevant to a wide variety of types, most notably real-time ridesharing (Agatz et al., 2011) and carpooling (e.g. Yan and Chen (2011a)).

The Internet and, more recently, mobile technologies have given rise to new, enhanced forms of ridesharing. These range from simple bulletin boards to complex services that can be accessed through web and smartphone applications and offer automated matching, routing, and payment. In this thesis, we focus on systems that offer automated matching of drivers and riders within an urban area. An example of a provider offering such a service is Flinc (`https://flinc.org`).

The key benefits of ridesharing for the participants are lower transport costs, the possibility of using high-occupancy lanes (where applicable), as well as reduced fatigue and travel times on the part of the riders. (Time savings are possible for riders who do not own cars and would otherwise use public transport.) Riders may also appreciate that they do not need to drive or even own a vehicle.

For the U.S.A., Jacobson and King (2009) estimated that introducing one additional passenger in every 100 vehicles could lead to annual savings of 0.80–0.82 billion gallons of gasoline. While these estimates are rough and hypothetical, they do underline the potential environmental benefits of such systems.

Ridesharing can significantly reduce the number of cars needed to satisfy the mobility needs of participants and, thus, lower congestion and other externalities related to heavy traffic when people rely on individual transportation to satisfy their mobility needs. It will, at the same time, also reduce the need for parking space, which is becoming an increasingly scarce and expensive commodity in most urban areas. (Congestion and parking are interrelated as searching for a parking space prolongs the driving time and can thus contribute to congestion.) Challenges related to high congestion and limited parking space arise in a myriad of urban areas around the world. In the U.S.A., for instance, urban congestion is an acute problem with far-reaching consequences. It is estimated that the cost of extra time and fuel in 498 urban areas in the U.S.A. in 2011 alone was roughly USD 121 billion. Congestion in the U.S.A. is expected to grow in the foreseeable future in spite of the planned measures to curb it (Schrank et al., 2012). In this context, ridesharing appears as an interesting possibility since it may result in significant effects without large investments.

Motivation for the research

Experience of practitioners and academics alike suggests that the design and operation of dynamic ridesharing systems is challenging and requires a thorough understanding of the behavior of such systems in different circumstances. The studies by Kamar and Horvitz (2009a), Kleiner et al. (2011), and Agatz et al. (2011) identify important challenges in (dynamic) ridesharing from an organizational and system-design perspective. They also propose different solution approaches to solve problems related to the matching of riders and drivers and the determination of payments to drivers.

In ridesharing, each driver has a specific itinerary and is willing to pick up and drop off riders en route. To accommodate the riders, the driver has to make a detour and make extra stops. The length of the detour and the number of extra stops depend on the driver's willingness to extend his trip time. This distinguishes genuine ridesharing from services in which the drivers act as de facto taxicab drivers, e.g. Uber (`https://www.uber.com`). The level of service in such systems may be higher due to the flexibility of the drivers, but this comes at a higher cost to the rider compared to genuine ridesharing. With the exception of shared taxi services, such services also do not necessarily reduce congestion.

Settings with very low density (e.g. recently launched ridesharing services, off-peak hours, rural areas) suffer from the so-called chicken-and-egg problem (Furuhata et al., 2013), where demand for trips is insufficient to attract sufficient supply and vice-versa. Such a situation may lead to stagnation or an implosion in the number of users. To overcome such a situation, the ridesharing system has to be designed well and must employ an effective matching algorithm to ensure the largest possible number of participants is matched and the system has satisfied users. Only users who have been successfully matched and have had a positive experience can be expected to continue to use the service and promote the ridesharing service to others. Thus, a high matching rate is a critical success factor for a ridesharing service.

That being said, ridesharing systems also have to minimize the effort and inconvenience for the participants. One way to achieve this is to restrict the number of riders per trip to at most one rider. In a single rider match, at most one pickup and drop off take place during a driver's trip. This minimizes the inconvenience of the driver and also makes it easy to divide the trip costs between rider and driver.

Agatz et al. (2011) simulate the launching of a real-time ridesharing application in the Metropolitan Area of Atlanta. They find that such systems hold important potential and that it is theoretically possible to introduce and sustain them. However, they also underline that overcoming the launch phase and growing a user base is extremely difficult if users are discouraged by not finding matches and thus stop announcing trips. They conclude it is important to match as many participants as possible and to provide incentives so that participants continue announcing trips even though they are not always matched. In the simulations, approximately 15 to 40 percent of riders and drivers remained unmatched (depending on the setting of the simulation). The simulations also showed that the ratio of matched participants predominantly depends on the distribution density of announced trips in space and time.

The motivation of this thesis can be broken down into the following points:

- ridesharing systems can have important environmental and societal benefits – they may help mitigate important transport challenges such as congestion and are thus worthwhile studying;
- there are strong indications on the part of both practitioners and researchers that ridesharing systems are challenging to operate – newly launched ridesharing services may quickly fail if not designed carefully; and
- it is clear from the literature that there is a lack of theoretical understanding and knowledge about the behavior of ridesharing systems and of the effects that individual participants' preferences may have on the functioning of a ridesharing system as a whole.

We hope the insights generated by this thesis will constitute a valuable contribution to the extant body of knowledge on ridesharing systems and their operation and that they will inform ridesharing system providers on how to design applications, matching algorithms and incentive schemes as well as form alliances with other systems such as public transport agencies and bike-sharing system providers.

Overview of the research area

While research on the sociological, psychological, and economic aspects of ridesharing in the transport context dates back to the 1970s (Bell, 1978; Dumas and Dobson, 1979), the first two important studies on the optimization of ride-matching in ridesharing systems only appeared in 2004 (Baldacci et al., 2004; Calvo et al., 2004). We have been able to identify a wide variety of papers dealing with this topic, which mainly build on research about other related optimization problems such as pickup and delivery problems (especially the dial-a-ride problem), the set cover problem, network flow problems or, more generally, combinatorial optimization and graph theory.

Ride-matching models are abstractions of matching problems faced by ridesharing providers. The models and solution algorithms used assign riders to drivers in a way that is optimal or close-to-optimal with respect to the system objective(s). The prevailing ride-matching optimization models in the literature are single-objective models. The common objective of these models is to minimize vehicle driving distance savings (the amount of vehicle driving distance that is saved by sharing rides among ridesharing participants), to minimize trip costs or the weighted sum of system costs which may be made up of trip costs, different time costs, different penalties etc. (e.g. Agatz et al. (2011); Baldacci et al. (2004); Calvo et al. (2004); Yan and Chen (2011b)). In contrast, Herbawi and Weber (2012c) developed several multi-objective models which consider the total distance of vehicles' trips, total time of vehicles' trips, total time of riders' trips, and the number of matched riders' requests.

The constraints imposed on the matching, the information assumed to be available as well as many other details vary from study to study. One reason for this is that one may conceive a ridesharing system in many ways. Indeed, across the globe, these systems differ in many respects depending on a local or national context, prevalent transport patterns in the region and many other factors. Further, there are several segments or niches of ridesharing (see Furuhata et al. (2013)).

As far as solution algorithms are concerned, many approaches have been developed. They range from exact approaches (Agatz et al. (2011); Baldacci et al. (2004); Bruglieri et al. (2011); Yan and Chen (2011b)) through to metaheuristic methods for more complex, multi-objective problems like evolutionary algorithms, e.g. NSGAII (Non-dominated Sorting Genetic Algorithm II), SPEA2 (Strength Pareto Evolutionary Algorithm 2) in Herbawi and Weber (2012c), and swarm optimization, e.g. bee colony optimization in Teodorovic and Dell' Orco (2008), as well as dedicated heuristics, e.g. in Calvo et al. (2004) and Yan and Chen (2011b).

For a more comprehensive overview of the ridesharing, carpooling, and other closely related literature, we refer the reader to two recent and complementary reviews: Furuhata et al. (2013) provide a thorough overview of the ridesharing and carpooling literature. They also describe the state of the art of contemporary ridesharing systems and discuss some of the key challenges to any wider adoption of ridesharing. Agatz et al. (2012), on the other hand, provides an overview of the optimization challenges in dynamic ridesharing and surveys some closely related optimization problems and models in the literature. Both studies also provide an overview and different classifications of the various types of ridesharing systems encountered in practice. Important dimensions include the dynamics of the system and the number of riders and drivers who are involved in a rideshare match. The advance of Internet-enabled mobile technology makes it possible to consider more dynamic ridesharing systems in which riders and drivers announce non-recurring trips on short notice (Agatz et al., 2011; Amey, 2011). In traditional carpooling, people travel together on recurring trips for a particular purpose, often for traveling to work, see (Baldacci et al., 2004) and (Calvo et al., 2004).

Agatz et al. (2011) represent the single rider, single driver rideshare matching problem by a max-weight bipartite matching problem. They explore different approaches to matching drivers and riders in real-time and investigate the impact of various service characteristics of the system. Their study shows that the success of a ridesharing system strongly depends on the participation density, e.g. the number of participants per square mile, and that a minimum participation density is required to ensure a stable system (in which participants do not leave the system because they repeatedly fail to find a match). Wang et al. (2014) extend this analysis by investigating the trade-off between matchings that are optimal for the system as a whole and matchings that are optimal for each participant in the system. They introduce the concept of stable matches in the ridesharing setting. Lee and Savelsbergh (2015) consider the employment of a small number of dedicated drivers to serve riders who would otherwise remain unmatched. The aim is to guarantee a certain service level (i.e. a fraction of the riders who are matched), thereby ensuring a stable system.

Another way to increase the number of riders who find a match is to allow riders to transfer between different drivers, i.e. to allow a rider to travel with more than one driver to reach his destination. Herbawi and Weber (2011c) consider a multi-hop ridesharing problem in which drivers do not deviate from their routes and time schedules. As such, the drivers' rideshare offers form the transportation network for the rider, who has to find a route that minimizes the costs, time, and number of transfers. Drews and Luxen (2013) extend this work by also allowing reasonable detours and time deviations for the drivers. While rider transfers might be acceptable to a driver, they are inconvenient for a rider as they may involve waiting times between rides and they increase the risk of something going wrong during the execution.

A different strand of research explores the potential benefits of using auction-based approaches in the ridesharing context. For instance, Kleiner et al. (2011) show that in the case where each driver may drive with a maximum of one rider, the use of a modified Vickrey auction may outperform an exact optimization algorithm in terms of the average matching probability. The advantage of a Vickrey auction is that a rider may value a certain ride at more than the ride's cost, which typically only covers a fraction of the actual costs. Since he may be ready to pay more to the driver, this may induce the driver to make a bigger detour, increasing the likelihood of finding a match. Likewise, Kamar and Horvitz (2009a) as well as Nguyen (2013) studied auction-based cost allocation mechanisms in combination with standard optimization algorithms. They found the mechanisms they devised to be superior to simpler cost allocation schemes. The results of these three studies are largely incomparable with each other and also with other researchers' results chiefly because different datasets are used and different types of information are assumed to be available in the studies. In addition, e.g. Nguyen (2013) works on a special problem concerning sharing rides in taxis, which is quite distinct from normal ridesharing.

A recurring topic in the literature is how to build a critical mass of users in a ridesharing system. When a ridesharing application is initially launched, the number of users is low and it is difficult to find matches. Kamar and Horvitz (2009b) showed that the number of feasible solutions (matches) for a single user depends strongly on the total number of active users. Therefore, in the initial stage the chance of finding a ride at a specific time for a specific origin-destination (O-D) pair is small. This is also true of certain O-D pairs at specific times in mature services. The inability to find a match, especially several times in a row, may drive a user away from the system and is therefore a serious issue that needs to be addressed.

It is difficult to estimate the approximate number of users required for a ridesharing system to work well. We were unable to identify any thorough studies regarding this. This number will primarily depend on the distribution of the origins and destinations of the drivers and riders in the transport network, their departure and arrival time windows and other preferences, although many other important factors could also be considered. In all probability, the only practical assessment method is simulation.

Agatz et al. (2011) also conducts experiments to observe the effects of variations in departure time flexibility, the time flexibility of drivers, and the time flexibility of riders. The general conclusion is that additional flexibility can have important positive effects on the matching rate, but mostly when the initial flexibility is low. Diminishing returns to scale were observed for all types of flexibility. In both studies, these experiments were performed for a single setting and a single density level by examining a few different parameter values only.

Several papers investigate shared taxi or ridesharing services in which multiple riders can be served on a single trip. Since this involves deciding the sequence of the pickups and drop-offs of the riders, it is more computationally challenging than solving a single rider, single driver setting. Wang et al. (2016) consider a pickup and delivery problem in which a driver can use HOV lanes and/or receives toll savings when traveling with a certain number of passengers. They provide an integer programming formulation and present a Tabu search heuristic. Their results show that it can be beneficial to make detours to pick up (additional) passengers and be able to use HOV lanes when the time savings on HOV lanes are significant. Hosni et al. (2014) consider the problem of assigning passengers to shared taxis. They formulate a mixed integer programming model and present a Lagrangian decomposition approach as well as two heuristics to solve the problem. Xu et al. (2015a,b), on the other hand, study the interaction between ridesharing and traffic congestion at an aggregate system level. They focus on modeling whether or not travelers will participate in ridesharing given certain congestion conditions and financial incentives.

Research agenda

The overarching research agenda of this thesis is the design and exploration of theoretical approaches to improve the functioning of ridesharing systems. It can be viewed as a natural continuation of the body of work originally started by Niels Agatz, Alan Erera, Xing Wang, and Martin Savelsbergh.

The first important pillar in this body of knowledge is a simulation study of a ridesharing system in Metro Atlanta (Agatz et al., 2011). This research paper established an important share of the terminology and concepts we use throughout this study and highlighted many of the issues that are inherent to ridesharing systems (and especially dynamic ridesharing systems). It also proposed a simple and effective solution approach to solve single driver–single rider ridesharing problems. The review paper that was published in roughly the same period provides an overview of the optimization challenges in dynamic ridesharing and represents a point of reference for future research on this topic (Agatz et al., 2012).

This was followed by a research paper on stable matching in the dynamic ridesharing context (Wang et al., 2014) and a research paper that considered the employment of a small number of dedicated drivers to serve riders who would otherwise remain unmatched (Lee and Savelsbergh, 2015). Both papers can be seen as complementary to the work presented in this thesis as they deal with the same underlying question, namely, how to make ridesharing systems perform better.

Research topics and objectives

This thesis investigates three relevant and interrelated main topics. Below, we shortly describe each one and present the main research objectives regarding it. Each topic is described and discussed in detail in the body of the thesis.

The first research topic is the examination of the impact of participants' flexibility on the performance of a single-driver, single-rider ridesharing system. The research objectives for this research topic are as follows:

- to identify and define types of participant flexibility that are relevant in the dynamic ridesharing context;
- to quantify the impact of these types of flexibility on system performance by conducting an extensive computational study; and
- to investigate the level of additional flexibility required to improve the effectiveness of a rideshare system.

The second research topic entails the examination of the potential benefits of introducing meeting points in a ridesharing system. The research objectives for this research topic are as follows:

- to design and implement an algorithm that optimally matches drivers and riders in large-scale ridesharing systems;
- to perform an extensive simulation study in order to understand how meeting points affect the number of matched participants as well as the system-wide driving distance savings; and
- to perform sensitivity analysis for a wide range of potential factors to understand the robustness of the effects of introducing meeting points into a ridesharing system.

The third and last research topic is the examination of the potential benefits of integrating ridesharing and public transit. The research objectives for this research topic are as follows:

- to design and implement an algorithm to optimally create single or multi-modal rideshare matches;
- to conduct an extensive simulation study to quantify the benefits of integrating ridesharing and public transit; and
- to perform sensitivity analysis for a wide range of potential factors to understand the robustness of the observed effects.

In Chapters 4 and 5, we present several methods and models that are a bi-product of the research undertaken in the framework of the first three research topics and can be seen as complementary to the methods and results presented in the rest of the thesis.

Description of the research methodology

The research presented in this thesis can be classified as axiomatic. This type of research is primarily driven by an (idealized) model of reality. *In this class of research, the primary concern of the researcher is to obtain solutions within the defined model and make sure that these solutions provide insights into the structure of the problem as defined within the model. Axiomatic research produces knowledge about the behavior of certain variables in the model, based on assumptions about the behavior of other variables in the model. It may also produce knowledge about how to manipulate certain variables in the model, assuming desired behavior of other variables in the model, and assuming knowledge about the behavior of still other variables in the model. Formal methods are used to produce this knowledge. These formal methods are developed in other scientific branches, mainly mathematics, statistics and computer science (Bertrand and Fransoo, 2002).*

This thesis is preoccupied with: (1) developing normative mathematical models to solve different planning problems that arise in ridesharing; (2) developing, implementing, and testing exact optimization methods to solve these planning problems; and (3) developing simulation frameworks to gain insights into the behavior of the elaborated ridesharing system model in order to inform researchers and practitioners about the potential behavior of real-life systems.

This research is thus predominantly normative in nature as it seeks to find an optimal solution for newly defined problems as well as compare various strategies for addressing a specific problem. However, as far as the explanation of the characteristics of the model is concerned the analysis of the models is clearly descriptive.

The planning problems mentioned above are in fact optimization problems in which the objective is to find an optimal solution (values of a set of decision variables that yield the minimum or maximum value of the objective function). A different set of solution methods and a different simulation framework was built to address each of the three main optimization problems (and research topics). In addition, a fourth mathematical model and solution method was developed and tested, which is presented in Chapter 5.

All frameworks were built using the Python programming language. Mathematical optimization problems, most of which are integer programs (IP), were solved by using either the Python Cplex API or (in one case) the Cplex graphical user interface and the Optimization Programming Language [OPL]. Some parts of the solution methods were also implemented using Cython and the C programming language (to obtain significantly faster implementations).

The algorithms and simulation frameworks were conceived, developed, and tested over periods of several months. Several hundred parameters, statistics, graphs, and solution outputs (the itinerary for each participant) were produced and examined in order to assure that the end results are correct. Once a simulation framework was thoroughly tested and validated, simulations were carried out which all comprised several runs of randomized instances. Some of the datasets were built using our own methodology and others were built based on approaches from previous studies (Agatz et al., 2011). The details of each simulation study can be found in the main body of the thesis.

Limitations

We have already pointed out that ridesharing system can be conceived in many ways. A multitude of different settings is possible and there is a range of different theoretical matching rules and constraints. Ridesharing systems vary in many respects depending on a local or national context, prevalent transport patterns in the region and many other factors. Further, there are several segments or niches of ridesharing (see Furuhata et al. (2013)).

Each chapter in this thesis very precisely defines what is the exact setting it models and studies. The findings need to be interpreted with these limitations and constraints in mind and must not be over-generalized. Unfortunately, the result of ridesharing studies are largely incomparable with each other because different datasets are used and different types of information are assumed to be available. This is understandable given this is a relatively recent topic and that ridesharing systems as such are also a relatively new phenomenon.

Organization of the dissertation

We start the introduction by presenting the background and motivation for this research. We continue by providing an overview of the research area, defining the research agenda as well as the research objective and scope of the thesis. We also describe the research methodology, discuss the limitations and present the structure of the dissertation.

In Chapter 1, we conduct a computational study to quantify the impact of different types of participants' flexibility on the performance of a single driver–single rider ridesharing system with the aim of providing the basis for the design of information campaigns and incentives schemes aimed at increasing the performance and success of ridesharing systems.

In Chapter 2, we investigate the potential benefits of introducing meeting points in a ridesharing system. With meeting points, riders can be picked up and dropped off either at their origin and destination or at a meeting point that is within a certain distance from their origin or destination. We design and implement an algorithm that optimally matches drivers and riders in large-scale ridesharing systems with meeting points and perform an extensive simulation study in order to understand how meeting points affect the number of matched participants as well as the system-wide driving distance savings.

In Chapter 3, we examine the potential benefits of integrating ridesharing and public transit. Ridesharing and public transit can, in fact, complement each other. On one hand, ridesharing can serve as a feeder system that connects less densely populated areas to public transit. On the other hand, the public transport system can extend the reach of ridesharing and reduce drivers' detours. As such, it may help to overcome incompatibilities in the itineraries of drivers and riders and facilitate the matching process. We present a solution approach to optimally create single or multi-modal rideshare matches, and we conduct an extensive numerical study on artificial instances that capture the main characteristics of many real-world transit settings and quantify the benefits of integrating ridesharing and public transit.

In Chapter 4, we study how to identify feasible driver-rider matches more efficiently. This can have an important effect on the total runtime of the algorithm because, typically, only a very small fraction of the possible matches are feasible, meaning it is possible to do much better if we do not have to fully evaluate all pairs. We exploit two ideas: (1) direct drive times from origin to destination can be used to efficiently identify those riders who have sufficiently small drive times to be matched with a particular driver; and (2) rider time windows can be stored in a memory structure that allows one to find riders with time windows that overlap with the time window of a driver in sub-linear time. We develop and test a data structure that combines optimizations (1) and (2) and test its performance.

In Chapter 5, we present a new mixed-integer linear programming model for the single driver–multiple rider matching problem that arises in certain types of ridesharing systems. The model we devise allows users to opt for driver or rider roles or, alternatively, to let the model determine what is best. If there is no match for a rider or a driver, the model is capable of adding this opportunity cost to the objective function value. The model minimizes the cost of all the trips that have to be performed to move the users from their origin to their destination nodes. We perform simulations on different instances constructed based on the ridesharing practice between Slovenia's two largest cities and comment on the results.

In the conclusion, we provide a summary of the most important insights and discuss the practical and theoretical contributions of the thesis. We finish the main body of the thesis with concluding remarks and a discussion of potential topics for future research.

1 MAKING DYNAMIC RIDE-SHARING WORK: THE IMPACT OF DRIVER AND RIDER FLEXIBILITY

1.1 Introduction

Establishing matches on short notice requires a centralized system that automatically and efficiently matches riders and drivers based on their trip information. Several dynamic ride-sharing providers, such as Flinc and Carma, offer such a centralized matching service. Participants announce their trips to the system via a mobile or web app. The system then matches drivers and riders and assists the driver with his trip, i.e., provides navigation information. Flinc, for example, is fully integrated in the NAVIGON app so that drivers can easily offer a ride while navigating to their destination. The system guides a matched driver to the rider's pickup and drop-off location.

Successfully matching riders and drivers on short notice requires a sufficiently large number of participants (Kamar and Horvitz, 2009a; Agatz et al., 2011). Several studies and pilot programs have examined the use of incentives to attract more people to a ride-sharing systems in order to reach a critical mass (see e.g. Epperson (2015)). However, few, if any, studies consider the use of incentives to increase the effectiveness of ride-sharing systems by rewarding participants for being more flexible. When a participant accepts to be more flexible in his departure time, for example, it is more likely that a match can be found. To employ incentives that encourage flexibility effectively, a ride-share provider needs to understand when additional flexibility is most beneficial and what type of flexibility is most beneficial for the system. Creating that understanding is the focus of this chapter.

The main contributions of this chapter can be summarized as follows: (i) We introduce and define three different types of participant flexibility that are relevant in the context of dynamic ride-sharing, i.e., matching flexibility, detour flexibility, and scheduling flexibility; (ii) We quantify the impact of these types of flexibility on system performance by conducting an extensive computational study; and (iii) We investigate the required level of additional flexibility that is required to improve the effectiveness of a ride-share system. While previous studies have looked at some aspects of participant flexibility in isolation, this is the first study to explicitly and extensively investigate the interaction between system density, level of flexibility, and type of flexibility.

We focus on dynamic ride-sharing systems that automatically generate matches between a *single* driver and a *single* rider, since this is the setting that is prevalent in today's market. Our efforts can also be viewed as an important and necessary first step in studying the value of flexibility in more complex ride-sharing systems with multi-rider matches and/or transfers. Our findings are relevant to ride-share providers, to public authorities that are considering to use dynamic ride-sharing to address road congestion, and to academics in this emerging field of research.

A few of the key findings are that (1) when the number of trip announcements in the system is small, participants need to be flexible in their departure times to find a match, (2) the extent to which drivers are willing to make detours is critical to the success of a ride-sharing system, and (3) the flexibility required to be matched can vary significantly for system

This chapter is based on Stiglic et al. (2016b).

participants. The insights generated can stimulate and facilitate the design of incentive schemes that encourage and/or compensate participants to accept less desirable matches in order to improve overall system performance. We note that in our numerical experiments, we assume that all announcements for the day are known in advance, which means that the matching rate obtained provides an upper bound on the matching rate that can be obtained in a dynamic setting. Earlier studies (e.g., Agatz et al. (2011)) have shown that because rides are announced shortly before their departure, the upper bound is quite tight and thus that the matching rates should be representative of what can be expected in dynamic settings.

The remainder of this chapter is organized as follows. In Section 2, we introduce the workings of a dynamic ride-sharing system, we define the types of participant flexibility, and discuss the potential role of incentives in dynamic ride-sharing systems. In Section 3, we describe the generation of the instances used in our computational study. In Sections 4 to 6, we motivate and discuss the computational experiments conducted and present the results. We conclude with a summary of findings and recommendations in Section 7.

1.2 A dynamic ride-sharing system

We study a dynamic ride-sharing system which receives a set of ride-share announcements S over time. Each trip announcement $s \in S$ has an origin location o_s and a destination location d_s. We distinguish between trip announcements of riders searching for rides $R \subset S$ and drivers offering rides $D \subset S$. We denote the distance from location i to j with d_{ij} and the travel time between the two locations by t_{ij}. The service time associated with the pick up and drop-off of a rider is indicated by τ. This value reflects the time necessary to make a stop, e.g., getting in and out of the car, validation of the identity of the rider and the driver.

We assume that drivers that are not matched, drive to their destination alone. We do not make any specific assumptions on the behavior of unmatched riders, i.e., they may use their own car, public transportation or a taxi to reach their destination. Announcements are submitted to the system at announcement time a_s and have a latest possible arrival time at their destination l_s. A driver i who is not matched at his latest possible departure time l_i - t_{o_i,d_i} will drive to his destination alone. We do not allow en route matching so a match needs to be established before the driver departs.

We assume that the primary system objective is to maximize the number of matched participants. The number of successfully matched participants is critical for the long-term sustainability of a ride-sharing service, since people who are (regularly) disappointed may stop using the system. As in Agatz et al. (2011), we formulate the problem as a bipartite matching problem. We create a node for each driver $i \in D$ and rider $j \in R$ and add an edge between i and j if the match is feasible, i.e. acceptable for the participants. We let the binary decision variable x_{ij} indicate whether the edge (i, j) is in an optimal matching $(x_{ij} = 1)$ or not $(x_{ij} = 0)$. Let F be the set of feasible matches. Then, the single rider, single driver matching problem that maximizes the number of matches can be formulated as follows:

$$\begin{aligned}
\max \quad z_1 = & \sum_{(i,j)\in F} x_{ij} && (1.1)\\
\text{subject to} \quad & \sum_{i\in D,(i,j)\in F} x_{ij} \leq 1 \quad \forall j \in R && (1.2)\\
& \sum_{j\in R,(i,j)\in F} x_{ij} \leq 1 \quad \forall i \in D && (1.3)\\
& x_{ij} \in \{0,1\}, \quad \forall (i,j) \in F && (1.4)
\end{aligned}$$

Objective function 1.1 maximizes the number of matches. Constraints 1.2 and 1.3 assure that each driver and each rider is only included in at most one match in an optimal matching. To obtain a matching that maximizes the driving distance savings, the objective should be replaced by $z_2 = \sum_{(i,j)\in F} c_{ij}x_{ij}$. We use a hierarchical optimization approach in which we first maximize the number of matches z_1 and subsequently maximize the system-wide vehicle miles savings z_2 (ensuring that the number of matches does not decrease by including the constraint $\sum_{(i,j)\in F} x_{ij} \geq z_1$).

To determine the set of feasible matches F, we impose several constraints, mostly related to participants' preferences. One such constraint is that only matches for which there are distance savings are allowed. That is, for a driver i matched to a rider j, the difference between the joint individual direct distances and the total distance of the driver in the match needs to be positive, i.e., $d_{o_i d_i} - d_{o_i o_j} - d_{d_j d_i} > 0$. The reason to impose this constraint is that we assume that trip costs are proportional to the distance traveled and therefore it is only possible to save costs if a match has positive distance savings.

We assume that the departure times of participants are somewhat flexible and that an announcement s has an earliest possible departure time e_s with $e_s < l_s - t_{o_s,d_s}$. We detail the different types of time flexibility below.

1.2.1 Matching Flexibility

Matching flexibility refers to the willingness of participants to depart earlier or later in order for a match to be found. The matching flexibility f_s of a trip s is the difference between the latest arrival time l_s and the earliest departure time e_s minus the direct travel time from origin to destination, i.e., $f_s = (l_s - e_s) - t_{o_s d_s}$. For simplicity, we assume that all participants specify an earliest departure time and that a greater matching flexibility is associated with a later latest arrival time.

For a given driver $i \in D$ and rider $j \in R$, we can determine the matching flexibility that is required to make a match between them time feasible. The earliest departure time e_i^j of the driver is given by $\max(e_i, a_j)$. That is, the driver can drive towards the origin of the rider only after his earliest departure time and after the rider has announced his trip. This implies that the earliest pickup time e_j^i of rider j is equal to $\max(e_i^j + t_{o_i o_j}, e_j)$ with an associated earliest arrival time at the rider's destination of $l_j^i = e_j^i + t_{o_j d_j} + \tau$. Hence, the minimum matching flexibility f_j^i required from rider j to be matched with driver i is $e_j^i - e_j + \tau$. Furthermore, driver i arrives at his destination at time $l_i^j = e_j^i + t_{o_j d_j} + t_{d_j d_i} + \tau$.

Hence, the minimum matching flexibility f_i^j required from driver i to be matched with rider j is $l_i^j - (e_i + t_{o_i d_i})$. A match between driver i and rider j is time feasible if $f_j^i \leq f_j$ and if $f_i^j \leq f_i$.

1.2.2 Scheduling Flexibility

Scheduling flexibility refers to the time available for a match to be found. The scheduling flexibility of a trip $s \in S$ is the difference between the latest departure time $l_s - t_{o_s,d_s}$ and the announcement time a_s. In the period between its announcement time and its latest departure time, a trip announcement can be matched. If no match has been found at the latest departure time the trip announcement leaves the system without a match. The scheduling flexibility of announcement s is composed of two parts: the matching flexibility (f_s) and the announcement lead-time ($e_s - a_s$).

For a specific match between driver $i \in D$ and rider $j \in R$, the latest commitment time is equal to $min(l_i - t_{d_j d_i}, l_j) - t_{o_j d_j} - t_{o_i o_j} - \tau$. This allows the driver to pick up and drop off the rider while respecting both the rider's and its own latest arrival time.

Figure 1.1 provides an example of the time line of a trip announcement with a lead-time of 15 minutes, i.e., this means the participant is able to leave at the earliest 15 minutes from now. Furthermore, the participant wants to arrive at its destination location within one hour from now. With a direct travel time of 30 minutes this implies a matching flexibility of 15 minutes and a scheduling flexibility of 30 minutes.

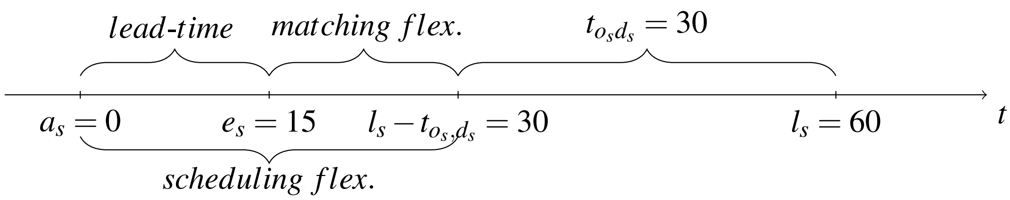


Figure 1.1: Scheduling and matching flexibility for announcement $s \in S$

1.2.3 Detour Flexibility

Detour flexibility is the willingness of drivers to make a detour in order to accommodate riders, i.e., the increase in trip duration drivers are willing to accept to pick up and drop off a rider. We assume the detour flexibility δ_i of driver $i \in D$ is a function of the duration of his direct trip, i.e., $\delta_i = c_f t_{o_i d_i} + \tau$, where c_f is the detour flexibility parameter and where τ, as usual, is the service time. The detour duration for driver $i \in D$ serving rider $j \in R$ is $t_{o_i,o_j} + t_{o_j,d_j} + t_{d_j,d_i} + \tau - t_{o_i,d_i}$.

1.2.4 Cost-sharing and incentives

People typically participate in ride-sharing to save money by sharing variable trip costs such as fuel, tolls, and parking. When a match is established, the ride-share provider typically (and automatically) assesses a trip fee to the rider, takes a commission, and compensates the driver. To increase the effectiveness of a ride-sharing system, i.e., establish more matches, the fees and compensations could be made dependent on a participant's flexibility. When participants join a ride-sharing service, for example, they may be asked to select a level of time flexibility with fees and compensations depending on the chosen level of time flexibility. More dynamic schemes may adjust fees and compensations based on the number of requests and offers in the system (as is done, for example, with Uber Surge pricing). The ultimate dynamic incentive scheme targets specific participants to enable a particular match. Of course, simply providing ride-sharing system participants with suggestions on how to improve their likelihood of finding a match may already increase the effectiveness of the system.

1.3 Generation of instances

To evaluate the impact of participant flexibility for different levels of system density, we generate sets of trip announcements of different size. We create the trip timing information for each announcement as follows. For each trip, we draw the earliest departure time from a truncated normal distribution with mean μ and standard deviation σ, truncated at 2σ to model a typical travel peak. We calculate the earliest arrival time by adding the direct travel time to the earliest departure time. Computational results reported in the body of the chapter are for instances generated with the standard deviation of departure time set to 30 minutes. In Appendix C, we report results for instances generated with smaller and larger standard deviations. For each trip announcement, we determine whether it represents a driver or a rider by flipping a coin, i.e., an announcement is equally likely to be a driver or a rider. In Section 1.4.7, we explore the impact of different driver-to-rider ratios.

1.3.1 Geography I: Corridor instances

In this set of instances, we focus on ride-sharing in a specific corridor, e.g., people living in a suburban residential area commuting to work in the center of an urban area. Several ride-share providers specifically focus their efforts on promoting their services in a specific corridor, as a corridor is typically characterized by high traffic densities and related traffic congestion. Since people traveling in a specific corridor have similar routes this simplifies the effort to find a match. Ride-share provider Carma participated in pilot programs focused on the heavily-congested State Route 520 corridor in Seattle, Washington and two tollways in Austin, Texas.

For analytical and presentational convenience, we work with a stylized corridor that is defined by a highway that provides a direct connection between the periphery and the urban center. We consider trips within a rectangle of 20 by 6 miles (see Figure 1.2). Destination locations for trips are drawn from a square with sides of 6 miles that is located at the east side of the rectangle (lightly shaded in Figure 1.2). More specifically, we create five non-overlapping circular areas (representing commercial centers) with a diameter of 1 mile within the square. Each of these circles contains 15% of the commuter destinations, while 25% of the commuter destinations can be anywhere in A (uniformly distributed in the square). Origin locations are drawn uniformly from the rectangle to the west of the square.

Since people may not consider ride-sharing on very short trips, we use rejection sampling to ensure that the direct trip distance is more than 1 mile for riders and more than 2 miles for drivers.

The corridor has a 20-mile east-west highway that splits the rectangle into a southern and northern half, with ramps at every mile. We assume that the average travel speed on the highway is 50 miles per hour ($v_h = 50$) and 20 miles ($v_l = 20$) per hour on the service streets. We assume that participants take the shortest (least-time) of two routes to go from their origin to their destination: (1) the shortest route using service streets only and (2) the shortest route using a combination of service streets and the highway. We approximate travel times using Manhattan distances and we assume that when a participant uses the highway, he uses the on-ramp closest to his origin and the off-ramp closest to his destination. Thus, the duration of the shortest route using the highway is computed as $t(p_1, p_2) = |\lfloor x_1 \rceil - \lfloor x_2 \rceil|/v_h + (|\lfloor x_1 \rceil - x_1| + |\lfloor x_2 \rceil - x_2| + |y_1 - y_0| + |y_2 - y_0|)/v_l$. See Figure 1.2 for an illustration of the corridor and of how travel times are computed. We note that this way of computing travel times means that the triangle inequality will not always hold. However, we have found the impact of this to be negligible.

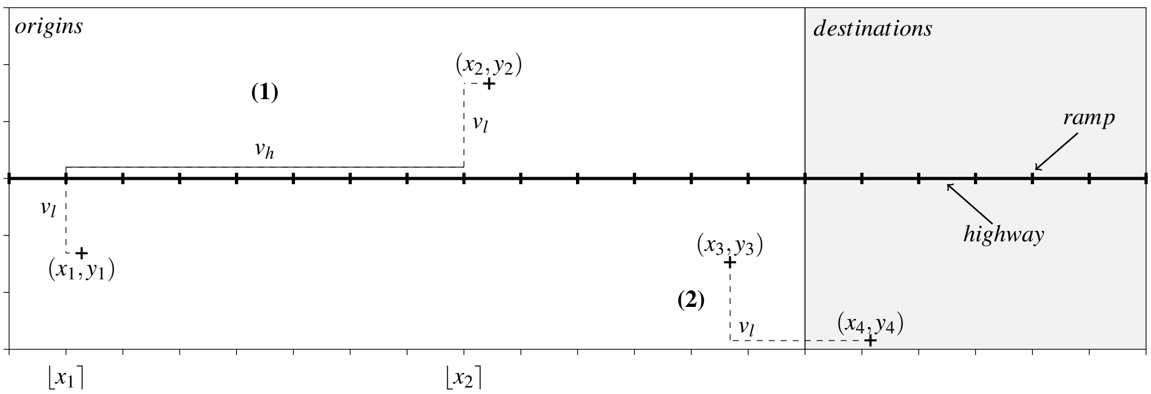


Figure 1.2: Illustration of the corridor and the travel time computations for paths that use the highway (1) and paths that do not (2)

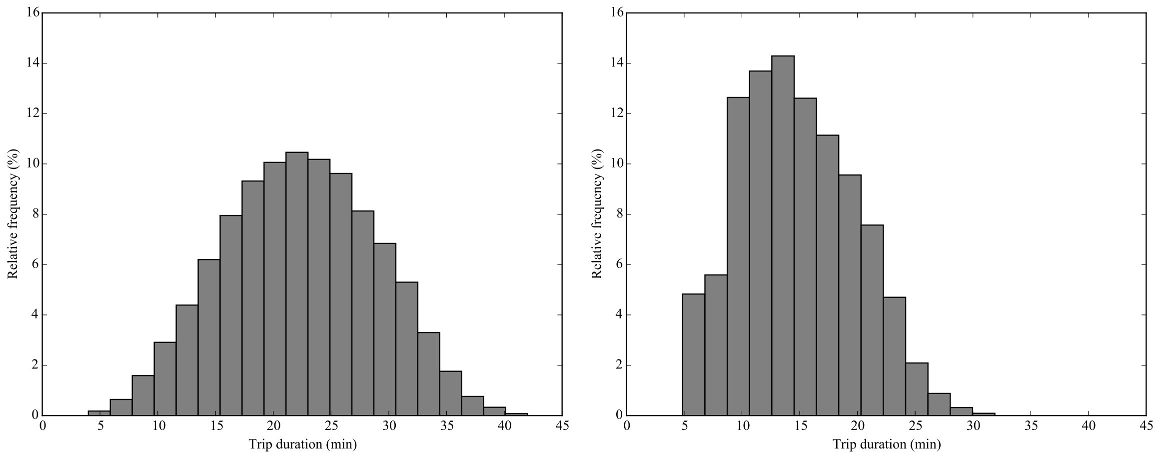
1.3.2 Geography II: Urban area instances

The second set of instances represents shorter, more spontaneous trips related to family and personal errands, shopping, as well as social and recreational activities. We model an urban area in the form of a square with sides of 6 miles. All origin and destination locations are generated uniformly randomly within this square. We calculate travel times based on Euclidean distances with a 30% uplift and a travel speed of 20 miles per hour. As before, we use rejection sampling to ensure that the airline distance between origin and destination locations is more than 1 mile for riders and more than 2 miles for drivers.

1.3.3 Base case setting

In all experiments, unless stated otherwise, we assume that all trips are announced 30 minutes before their earliest departure time. In the default setting, we assume participants are ready to postpone their arrival time by 20 minutes. Hence, the base case matching flexibility is 20 minutes and the base case scheduling flexibility is 50 minutes. The service time τ is set to 2 minutes and the value of the detour flexibility parameter c_f is set to 0.25, which means that we assume that drivers are willing to extend their trip duration by 2 minutes plus 25% of their direct trip duration. This limits the maximum detour duration of trips to approximately 12 minutes in corridor instances and 10 minutes in urban area instances.

Figure 1.3 shows the resulting distributions of trip durations for the two geographies. We see that urban area trips are shorter and that the distribution is skewed to the right because we do not consider very short trips. The jump between 5 and 10 minutes is due to the rule that the airline distance between origin and destination locations is more than 1 mile for riders and more than 2 miles for drivers.



(a) Corridor trips

(b) Urban area trips

Figure 1.3: Distributions of trip durations for 10,000 randomly generated trip announcements

1.4 Numerical results

In this section, we present the results of several computational experiments that seek to quantify the impact of density and participant flexibility on ride-sharing system performance, i.e., on the matching rate.

In all experiments, as in Agatz et al. (2011), we match the drivers and riders by solving a bipartite matching problem. We use a hierarchical optimization approach in which we first maximize the number of matches and subsequently maximize the system-wide vehicle miles savings for this maximum cardinality matching. When solving an instance, we assume we have perfect information, i.e., we know all the announcements for the day. The matching rate obtained therefore provides an upper bound on the matching rate that would be obtained in a dynamic setting. Previous research (Agatz et al., 2011; Wang et al., 2014) has shown that due to the short announcement times and limited flexibility, the difference in performance between a dynamic rolling horizon setting and a perfect-information setting is relatively small. Consequently, we are confident that our computational study provides meaningful insights.

The simulation framework is implemented in Python 2.7. The match generation module is implemented in C. We use CPLEX 12.6 to solve the matching problems. All results, unless stated otherwise, are averages over 20 runs. Section 1.4.9 provides more information about the variability observed in the results across the 20 runs.

1.4.1 Description of experimental framework

Hereunder, we provide a description and the pseudocode (Algorithm 1) of the experimental framework that was used to obtain the results that are displayed in Section 1.4.2 in Figure 1.4, in Section 1.4.3 in Figure 1.7, as well as in Section 1.4.5 in Figure 1.10.

We adopt an incremental computational approach that allows us to carry out the experiments in an efficient way. For each replication of an experiment, we randomly generate a large set of trip announcements A. We then repeatedly draw a random subset of trip announcements without replacement from A and add them to our sample S to be used as part of the experiment. For each subset of trips that is added to the sample, we identify all feasible matches involving a trip in the new subset and add these new matches to the set of all matches M. Furthermore, we maintain the matches in M in non-decreasing order of the flexibility required from the rider and the driver. As a consequence, we can easily identify those matches that correspond to a certain level of trip density and that require a certain level of participants' matching flexibility.

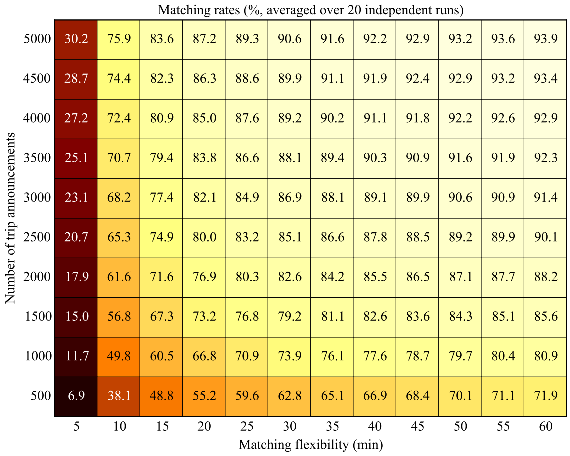
Algorithm 1: Outline of system-wide flexibility experiments

```
Randomly generate a set of trip announcements A ;
Set a value k by which we will increment the sample size ;
Set a range F of flexibility values which we wish to inspect ;
Initialize empty sample S ;
Initialize empty container of feasible matches M ;
Initialize empty results list R ;
while size(A) ≥ k do
    Pop k trips from A and append to S ;
    Determine all new feasible matches propagated by the k new trips ;
    Add all new matches to M ;
    Initialize optimization ;
    for f_max in F do
        Add all matches with f ≤ f_max to optimization ;
        Solve optimization problem and determine matching rate r and distance
          savings s ;
        Append (size(S), f_max, (r,s)) to R ;
    end
end
Return R ;
```

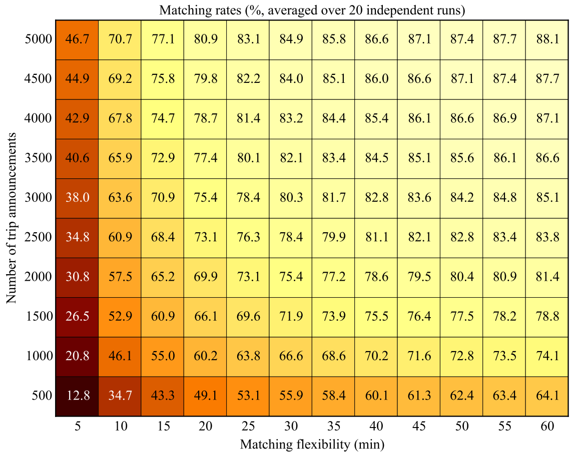
1.4.2 The impact of matching flexibility

In this section, we investigate the impact of the matching flexibility on the matching rate at various system densities. In the experiments, we assume that each trip has the same *system-wide matching flexibility*. We compute the average matching rates for various levels of system density (between 500 and 5000 trip announcements) and matching flexibilities (between 5 and 60 minutes).

Figure 1.4 presents the average matching rates for both corridor and urban area instances, where a lighter color is associated with a higher matching rate. The results show, as expected, that for a given level of matching flexibility, the average matching rate increases with the number of trips in the system, but that the (marginal) increases diminish. We also see that additional density is more beneficial at low matching flexibilities and more flexibility is more beneficial at low densities. For example, Figure 1.4a shows that increasing the matching flexibility from 10 to 30 minutes results in an increase of 20 percentage points (p.p.) of the average matching rate with 1,000 trips and less than a 15 p.p. increase with 5,000 trips.



(a) Corridor trips



(b) Urban area trips

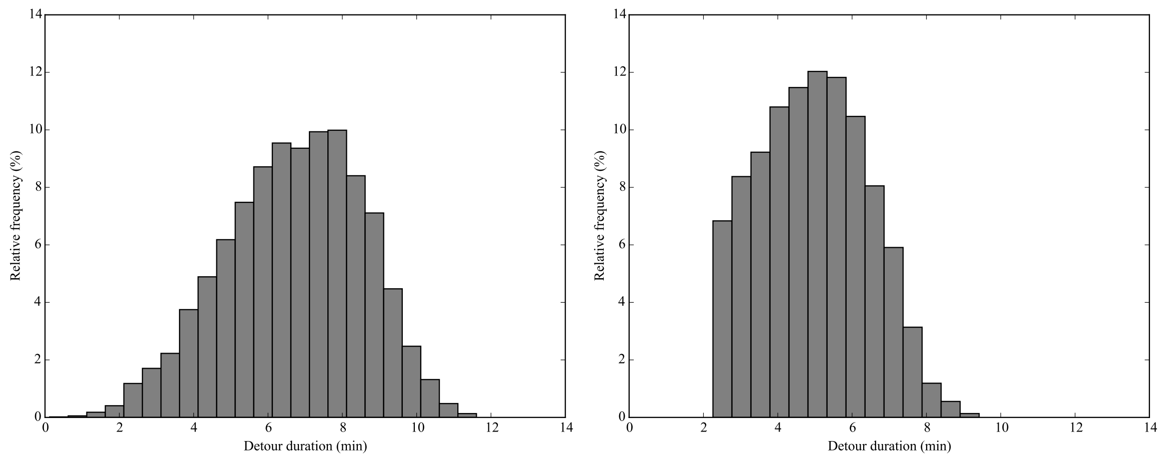
Figure 1.4: Average matching rates for different combinations of system-wide matching flexibility and system density

The results suggest that low system-wide matching flexibility of 5 minutes heavily limits the ability of the system to establish matches. That is, even at the highest system density (5,000 trips) the average matching rate is only 30.2% at a matching flexibility of 5 minutes. The reason for this is that low matching flexibilities only allow matches between trips with very similar time schedules and travel paths. That is, departure times have to be almost perfectly aligned to allow a match and, more importantly, there is limited time for driver detours. Both effects limit the number of ride-sharing opportunities and consequently also the number of matches that are established.

A higher matching flexibility, on the other hand, can make up for a lack of density. For example, a matching flexibility of 30 minutes results in matching rates of 55.9% on average at the lowest density (with urban area trips). However, while our results point to the fact that moderate levels of matching flexibility are very beneficial, the system does not gain much if participants increase their matching flexibility beyond 30-40 minutes.

We see that the matching rates are generally higher for the corridor instances (Figure 1.4a) than for the urban area instances (Figure 1.4b). When interpreting differences between corridor and urban area instances, it is important to realize that the corridor setting covers a larger area than the urban area setting (20 by 6 miles versus 6 by 6 miles) and, therefore, the system density in terms of origins and destinations per square mile is not identical for the same number of trip announcements. Also, there is only one prevailing trip direction in the corridor, whereas trip directions are random in the urban area. Finally, destinations in the corridor instances are more clustered.

Another difference between corridor and urban area instances is that trips are somewhat longer in the corridor instances, which implies that drivers have more detour flexibility (which is proportional to the trip length). The latter explains why the advantage of corridor instances over urban area instances disappears at a very low matching flexibility as this limits the detour flexibility. The impact of (indirectly) limiting detour flexibility becomes clear when we examine Figure 1.5, which shows the distribution of detour durations for a corridor and an urban area instance with 3,000 trip announcements.



(a) Corridor trips

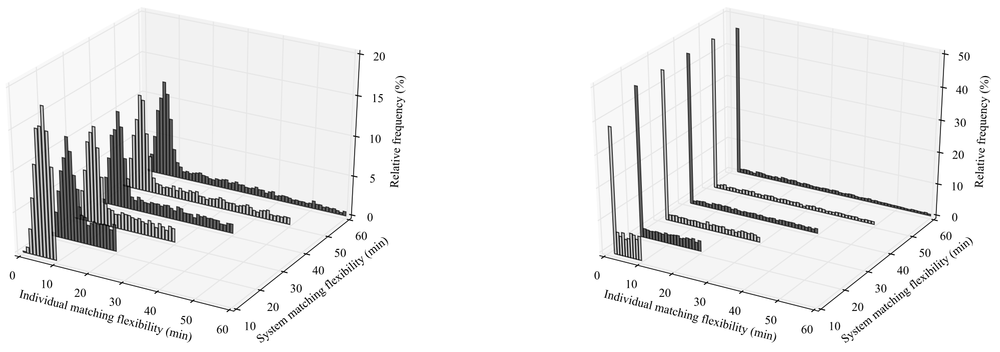
(b) Urban area trips

Figure 1.5: Distributions of detour durations for feasible matches in an instance with 3,000 trip announcements

To provide more insight into the flexibility that is actually used, Figure 1.6 shows the required matching flexibility for drivers and riders in a corridor instance with 1,000 trips for different system-wide matching flexibilities. The figure shows a different pattern for drivers than for riders. To understand these differences, it is important to realize that riders do not have to perform detours and, consequently, their matching flexibility only represents the 'waiting time' from departing later than their earliest departure time. For drivers, the actual matching flexibility captures this waiting time *and* their potential detours.

Approximately 45% of the riders can potentially be picked up at their earliest departure time, which implies that only 2 minutes of matching flexibility is required. The required matching flexibility of the remaining 55% of riders is spread relatively uniformly across the range of values. Detour durations range from 2 to 12 minutes, and the "humps" at the start of the distributions indicate that most drivers too depart at or shortly after their earliest departure time with the remaining drivers spread relatively uniformly across the range of values.

Our results show that it may be difficult to achieve high matching rates in a ride-sharing system if participants are not willing or able to accept a matching flexibility of at least 15 to 20 minutes. The results also show that matching flexibility is especially critical when the number of trip announcements is low. A related insight is that the "critical mass", i.e., the number of trip announcements at which a ride-sharing system becomes sustainable,



(a) Drivers

(b) Riders

Figure 1.6: Distributions of the required matching flexibility for different levels of system-wide matching flexibilities

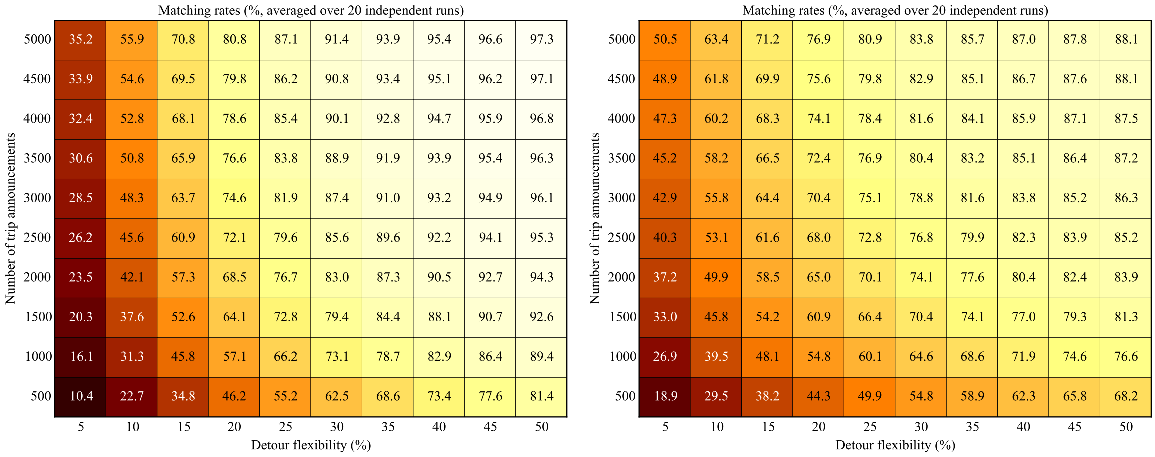
depends strongly on the matching flexibility of the participants. For example, we see in Figure 1.4a that a 90% matching rate can be achieved with 2,500 trips and a matching flexibility of 60 minutes. On the other hand, with a matching flexibility of 30 minutes, a matching rate of 90% can only be expected with 5000 trips.

1.4.3 The impact of detour flexibility

In this section, we present the results of an experiment designed to quantify the effects of detour flexibility on system performance. Similar to the previous section, we compute the matching rates for various levels of system density (between 500 and 5000 trip announcements) and vary the *system-wide detour flexibility* (between 5% and 50%) for a fixed matching flexibility of 20 minutes for all participants. Figure 1.7 presents the average matching rate for both corridor and urban area instances. The results show, again, that for a given level of detour flexibility, the average matching rate increases with the number of trips in the system, but that the marginal increases diminish. It is also apparent that additional density has a greater impact on the matching rate when the detour flexibility is small. (Note that differences between the base case results in Figure 1.4 (fourth column) and Figure 1.7 (fifth column) are due to randomness.)

We observe large gaps between settings with high and low detour flexibility: Apparently, the willingness of drivers to increase their trip duration by a higher percentage can increase the matching rate substantially. It also appears that increasing detour flexibility from moderate values of 25% to higher values of, e.g., 50%, results in significant increases in the matching rate even at high system density (e.g. 8-10 p.p. with 5000 corridor trips).

To provide more insight into the actual detours, Figure 1.8 plots histograms of detour durations for all matches in the optimal solutions with a detour flexibility of 25% and 50%, respectively. We see that the detours increase with the overall detour flexibility. However, the detours generally appear reasonable from the perspective of the required inconvenience of the drivers. While there are more matches with detours between 7 and 10 minutes at a detour flexibility of 50%, there are few matches, i.e. $< 5\%$, that have a detour of more than 10 minutes. This suggests that the most important impact of greater system-wide detour

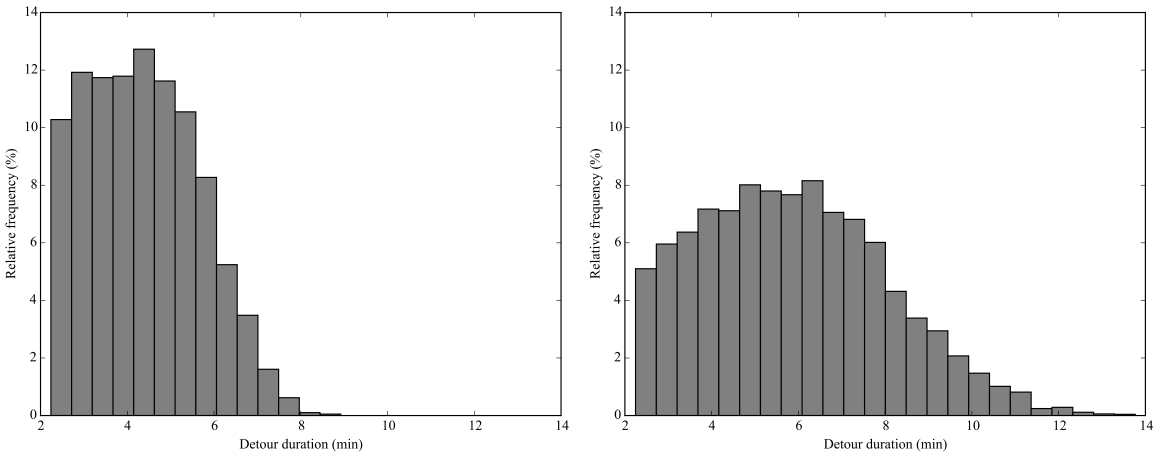


(a) Corridor trips

(b) Urban area trips

Figure 1.7: Results for different combinations of system-wide detour flexibility and system density

flexibility comes from the fact that many drivers increase their detour by only a few minutes. If a provider can find a reward scheme to make all of the drivers accept these detours, than a matching rate of 89.4% can be expected with only 1,000 participants.



(a) 25% detour flexibility

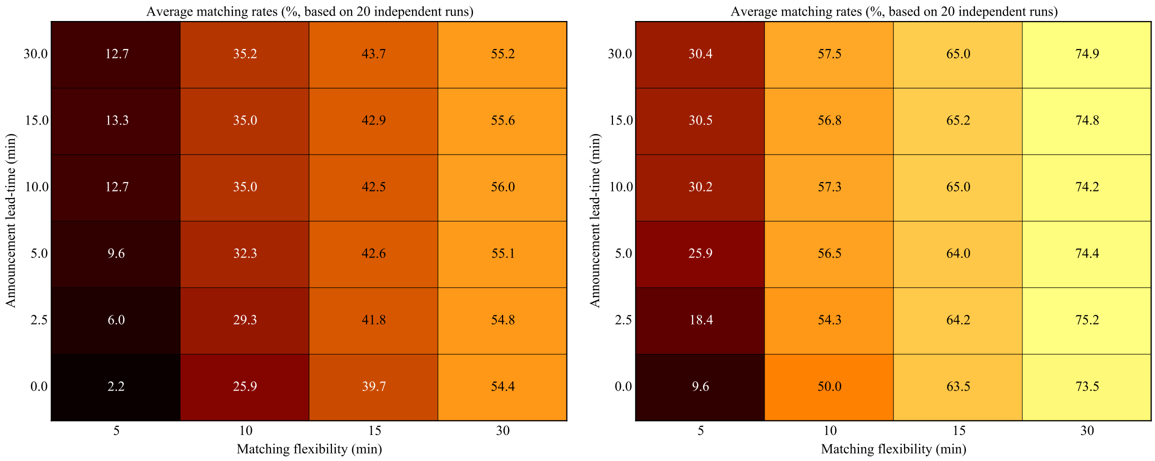
(b) 50% detour flexibility

Figure 1.8: Distributions of detour durations in optimal solution for an urban area instance with 1,000 trip announcements based on 20 independent random runs

1.4.4 The impact of scheduling flexibility

In this section, we present the results of an experiment designed to analyze an important part of the impact of *system-wide scheduling flexibility* on the system matching rate. In contrast to the previous experiments, we only consider the urban area instances (the effect is the same for the corridor instances). Moreover, we only consider two levels of system density: 500 and 2,000 trips and vary the announcement lead-time between 0 and 30 minutes and the matching flexibility between from 0 to 30 minutes. Recall from Section 1.2.2 that we define the scheduling flexibility as the sum of the matching flexibility and announcement lead-time.

Our simulation framework does not allow us to simulate the entire effect of the different values of announcement lead-times and matching flexibility. It allows us to identify situations in which the driver needs to leave his or her origin before the earliest departure time of the rider for the match to be feasible. Announcement lead-time plays an important role in all such cases. On the other hand, we do not take into account the inefficiencies that may appear in a dynamic match commitment process. These are due to the need to commit early, which means that some potential matches that appear in the optimization after the match is already committed are not considered. This is expected to have the biggest negative effect on the matching rate when both matching flexibility and scheduling flexibility are small. We know from previous studies ((Agatz et al., 2011), (Wang et al., 2014)) that the impact is negligible for moderate values of matching flexibility.



(a) 500 trip announcements (b) 2,000 trip announcements

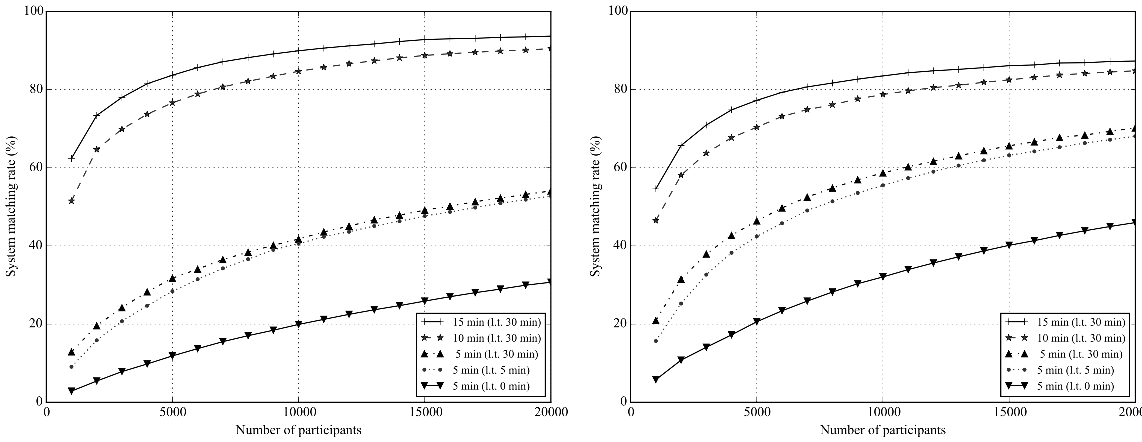
Figure 1.9: Results for different combinations of system-wide matching flexibility and system-wide announcement lead-time for urban area instances

Figure 1.9 shows that the announcement lead-time only makes a difference at relatively low matching flexibilities. For example, with 10 minutes matching flexibility, the difference in performance between a setting with no lead-time and a setting with a 30 minute lead-time, on average, is less than 10 p.p. and 8 p.p. for 500 and 2,000 trip announcements, respectively, and with a 15 minutes matching flexibility, the difference in performance between a setting with no lead-time and a setting with a 30 minute lead-time is less than 5 p.p. and 2 p.p. for 500 and 2,000 trip announcements, respectively.

The results show that a combination of short lead-times ($\leq$ 5 min) and small matching flexibilities ($\leq$ 10 min) severely limits the functioning of a ride-sharing system. Note that in a practical dynamic setting the performance is likely to be even worse because there is little time to establish the matches. Our results also suggest that announcement lead-time does not significantly affect the matching rate, on average, if the matching flexibility is sufficiently high.

1.4.5 The impact of density

The results thus far show that low matching flexibility and small lead-times have a negative effect on the matching rate. This suggests that highly dynamic, "on-the-fly" ride-sharing systems may not be feasible in practice. In this section, we investigate whether higher densities make such a truly on-demand transportation system feasible by increasing the number of trips in increments of 1,000 up to 20,000. For the most dynamic setting, with a system-wide matching-flexibility of 5 minutes, we also examine settings with shorter announcement lead times of 0 and 5 minutes, respectively. These settings represent an even more responsive system, where participants announce their trips at the moment they want to depart.



(a) Corridor trips (b) Urban area trips

Figure 1.10: Matching rates for different highly dynamic and high density ride-sharing systems

The results in Figure 1.10 indicate that even with a large number of trips, high matching rates ($> 60\%$) are not achieved in the most dynamic settings. With matching flexibilities of 10 and 15 minutes, much higher matching rates can be achieved. Importantly, we observe that the difference between an announcement lead time of 5 and 30 minutes, with a 5-minute matching flexibility, is quite small. At the same time, the setting with no announcement lead time performs much worse, suggesting that a moderate lead time of 5-10 minutes is certainly necessary. Note that the reason that a company like Uber can operate effectively with little or no announcement lead time is that it operates with (a large number of) dedicated drivers.

These findings have clear implications for providers: in the start-up phase of a ride-sharing system, when densities are small, it is beneficial to focus on attracting participants that have some flexibility in their schedule. Also, providers might consider designing their system in a way that encourages participants to be flexible, possibly even providing incentives to increase flexibility. The more flexible a participant will be, the better. As the system grows, matching participants becomes easier. However, our results suggest that moderate levels of matching flexibility are necessary for ride-sharing to work effectively even at high densities. Finally, our results suggest that highly dynamic, on-demand ride-sharing systems, with a pick-up service guarantee of 5 minutes or less, cannot work well even with very high participation rates.

1.4.6 The impact of trip characteristics

In this section, we investigate the effect that different trip characteristics have on the likelihood of finding a match. In particular, we consider the impact of trip duration, earliest departure time, and proximity to the highway for corridor instances.

We generate and solve 3,000 random corridor instances with 1,000 trip announcements, with all parameters set to base case values. We do not generate the five random non-overlapping circular areas in the destination square – destination locations are drawn from the destination square uniformly randomly. We also reject and replace (the few) trip announcements with trips that are shorter than 5 minutes and longer than 40 minutes. These two small modification of the procedure detailed in Section 1.3.1 were made for presentational convenience.

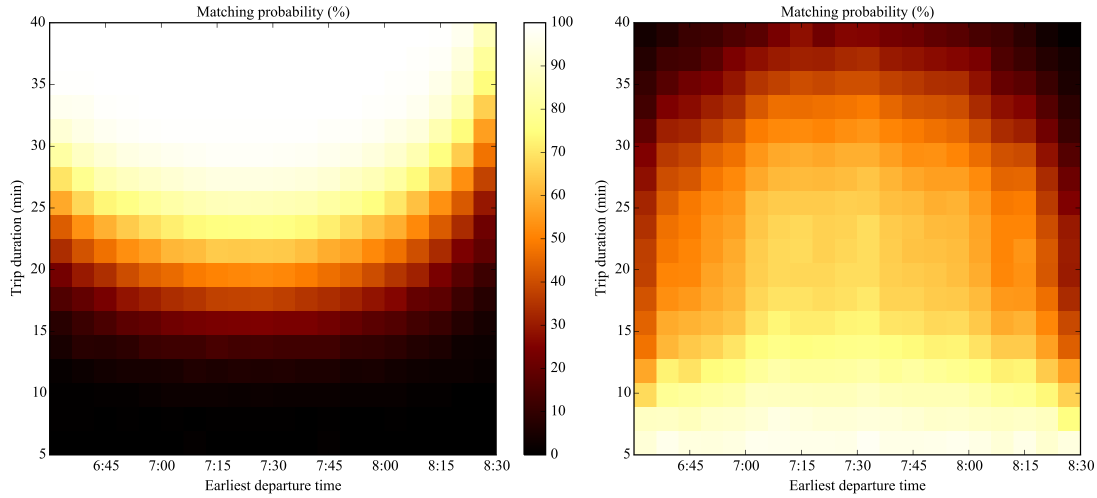
For each trip announcement in each random run, we register the *role*, *earliest departure time*, *trip duration*, *origin and destination location coordinates*, and whether it was *matched in the optimal solution*. We use the aggregated information from all the 3,000 random runs to visualize average matching probabilities in three different heat maps.

The first heat map has departure time on the x-axis and trip duration on the y-axis. The second one represents the area where origin locations are drawn from and the third represents the area where destination locations are drawn from. Each cell provides information about the average matching probability for trip announcements with certain trip characteristics (e.g. trip duration between 20 and 22 minutes and departure time between 7:30 and 7:35). The heat maps *do not* provide estimates of matching probabilities for individual trip announcements. These may be much greater or smaller depending on other determinants that are not captured in a particular figure.

Figure 1.11 shows the average matching probability for different earliest departure times and trip durations for both rider and driver announcements. Consistent with the results of Agatz et al. (2011), we observe that longer trip durations are associated with higher matching rates for drivers. For example, drivers with trips of more than 35 minutes have a matching rate of 90-95%, while drivers with trips of less than 15 minutes only have a matching rate of less than 30%. The opposite is true for the riders; riders with short trips are more likely to find a match. However, the differences in matching rates between riders with short trips and riders with long trips is not as pronounced as for drivers. As expected, we also see that it is easier to match participants that depart close to the mean of the earliest departure time distribution (slightly before the mean for drivers and slightly after the mean for riders). Those participants that can only depart relatively late are hardest to match (which is intuitive, since a departure time can only be shifted forward).

A different perspective is provided in Figure 1.12 and 1.13, which show the average matching probabilities for drivers and riders in a corridor instance based on their origin locations (to the left) and destination locations (to the right). Label 0 on the vertical axis marks the position of the highway and the other labels on the vertical axis mark the distance to the highway; label 0 on the horizontal axis marks the position of the border between the origin locations area and the destination locations area and the other labels on the horizontal axis mark the distance to this border.

The heat maps confirm what we expected: drivers with longer trips have a higher chance of being matched (both in terms of distance from origin to area with destinations and distance from destination to area with origins) and riders that have easy access to the highway have a higher chance of being matched (both in terms of origin locations and destination locations). Due to the presence of highway ramps, a clear pattern with periodicity of 1 mile, is apparent in each of the four figures.



(a) Driver announcements

(b) Rider announcements

Figure 1.11: Matching probabilities for combinations of departure time and trip duration (based on 3,000 runs)

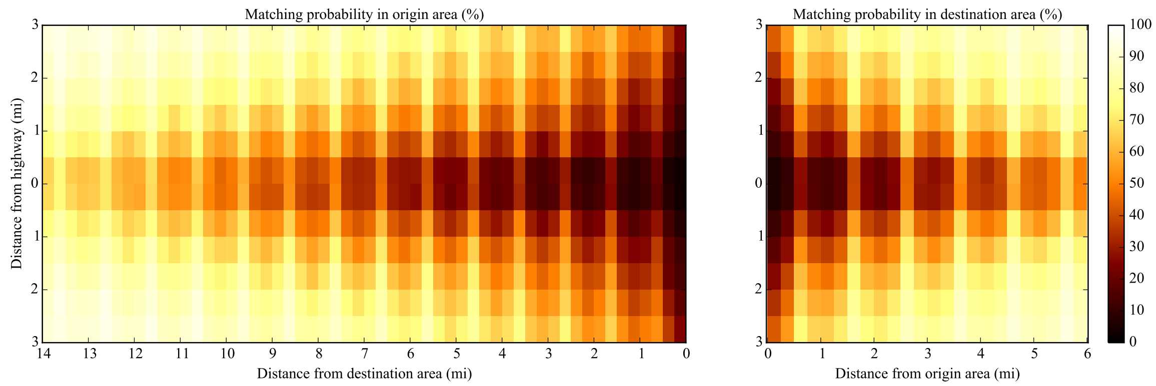


Figure 1.12: Matching probability for specific origin and specific destination locations for ***drivers*** (based on 3,000 runs)

Riders, but especially drivers, with origin locations closer to the area with destination locations have a smaller chance of being matched. These drivers have shorter trips on average and thus also shorter maximum detour duration. Consequently, it is harder to match them, which results in matching probabilities below 30% towards the far right of the origin area. The matching probabilities for riders in the same area are much higher than that, especially around the highway. This is because drivers with origins to the left pass by and may pick them up by making a detour from the highway. However, we see that the further riders are from the highway, the less likely this is to happen. Such riders can still be matched in local matches (with drivers that have their origin close to the rider's origin).

The results for the destination area are a mirror image of those in the origin area - especially for the drivers. Drivers with destinations on the far left are hardest to match and those with a destination on the far right are easiest to match. This result is intuitive since those with destinations on the far right potentially pass close by a large number of rider destinations. For the riders, we see that those with destinations towards the upper, lower, or right border of the destination area are hardest to match and those that are close to the highway are easy to match.

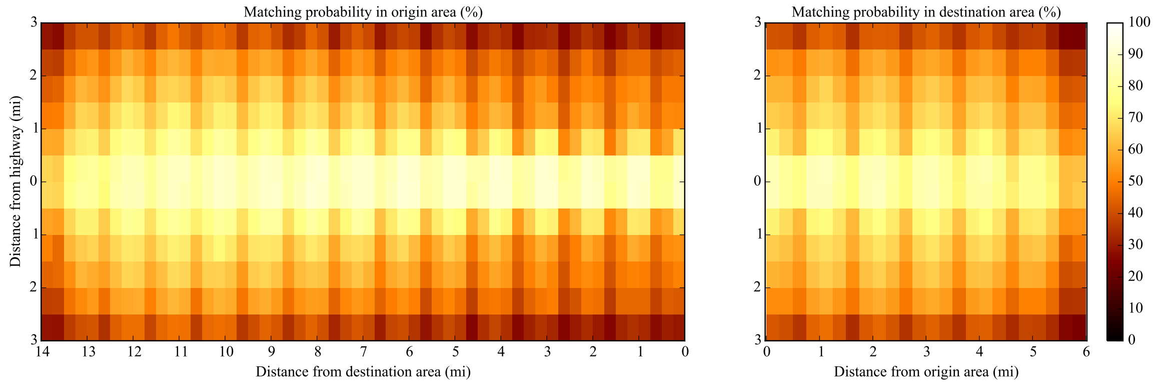


Figure 1.13: Matching probability for specific origin and specific destination locations for ***riders*** (based on 3,000 runs)

While the above results are hardly surprising, they highlight the importance of trip characteristics in the chance of finding a match. The origin and destination location characteristics that lead to a high chance of being matched are almost diametrically opposed for drivers and riders. This suggests that a ride-share provider might encourage a participant to flip his role (from driver to rider or vice versa) if his chances of being matched with his preferred role are small.

1.4.7 The impact of the driver-to-rider ratio

In the experiments discussed above, the instances had approximately equal numbers of drivers and riders. To assess the impact of different driver-to-rider ratios on system performance, we conduct a series of experiments using corridor instances with 1000 participants with different driver-to-rider ratios. In particular, we examine driver-to-rider ratios 2:1, 1:2, and 1:1. To be able to evaluate the interaction of the driver-to-rider ratio with the different types of flexibility, we also vary the matching flexibility and the detour flexibility in the experiments. The results can be found in Table 1.1, where we show the matching rates for the drivers and the riders, respectively.

Driver-rider ratio	1:2			1:1			2:1		
Matching flex.	10 min	20 min	30 min	10 min	20 min	30 min	10 min	20 min	30 min
Drivers									
Detour flex.: 0.15	*41.34*	*58.28*	*68.07*	*31.57*	*46.27*	*52.83*	*20.71*	*29.84*	*33.77*
Detour flex.: 0.25	*62.96*	***83.00****	*89.30*	*49.98*	***66.72****	*73.56*	*31.74*	***39.62****	*41.99*
Detour flex.: 0.35	*71.23*	*93.17*	*96.97*	*59.25*	*78.21*	*84.71*	*37.54*	*44.82*	*45.74*
	-41.66	-24.72	-14.93	-35.15	-20.45	-13.89	-18.91	-9.78	-5.85
Diff. (p.p.) **	-20.04	**-**	+6.30	-16.74	**-**	+6.84	-7.88	**-**	+2.37
	-11.77	+10.17	+13.97	-7.47	+11.49	+17.99	-2.08	+5.20	+6.12
Riders									
Detour flex.: 0.15	*20.91*	*29.54*	*34.03*	*32.24*	*46.92*	*53.23*	*42.14*	*59.22*	*67.19*
Detour flex.: 0.25	*32.12*	***42.78****	*45.10*	*50.57*	***66.51****	*73.15*	*63.40*	***79.46****	*84.21*
Detour flex.: 0.35	*35.48*	*46.12*	*47.70*	*59.45*	*79.91*	*85.01*	*76.06*	*89.20*	*92.58*
	-21.87	-13.24	-8.75	-34.27	-19.59	-13.28	-37.32	-20.24	-12.27
Diff. (p.p.) **	-10.66	**-**	+2.32	-16.06	**-**	+6.64	-16.06	**-**	+4.75
	-7.30	+3.34	+4.92	-7.06	+13.40	+18.50	-3.40	+9.74	+13.12

* Matching rate for default flexibility parameter values.

** Differences to default case in percentage points.

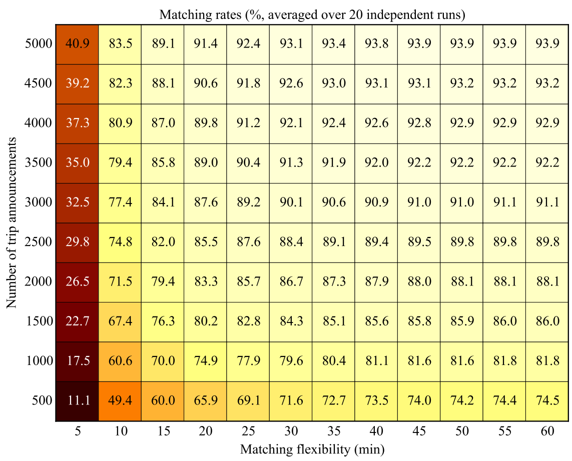
Table 1.1: Average matching rates for *drivers* and *riders* (in %) for different driver-to-rider ratios and different combinations of system-wide matching flexibility and system-wide detour flexibility (based on 20 independent runs)

As expected, we observe different matching rates for drivers and riders in the unbalanced scenarios. Relatively more participants of the smaller group are matched and relatively fewer participants of the larger group are matched. Moreover, we see that both the matching and the detour flexibility have a noticeable impact on the performance, with detour flexibility being more important than matching flexibility, as there are positive entries only in the lower right below the diagonal for changes from the base case. Examining the differences from the base case also reveals that additional flexibility has a larger impact when the driver-to-rider ratio is 1:1. For example, an increase in the matching flexibility from 20 to 30 minutes and in the detour flexibility from 0.25 to 0.35 increase the driver and rider matching rates by 17.99% and 18.50%, respectively, for driver-to-rider ratio 1:1, but only by 13.97% and 4.92%, respectively, for driver-to-rider ratio 1:2 and only by 6.12% and 13.12%, respectively, for driver-to-rider ratio 2:1. It is also interesting to observe that a surplus of riders appears to be better for the performance of the system than a surplus of drivers, with a driver and rider matching rate of 83.00% and 42.78%, respectively, for driver-to-rider ratio 1:2 and of 39.62% and 79.46%, respectively, for driver-to-rider ratio 2:1 in the base case.

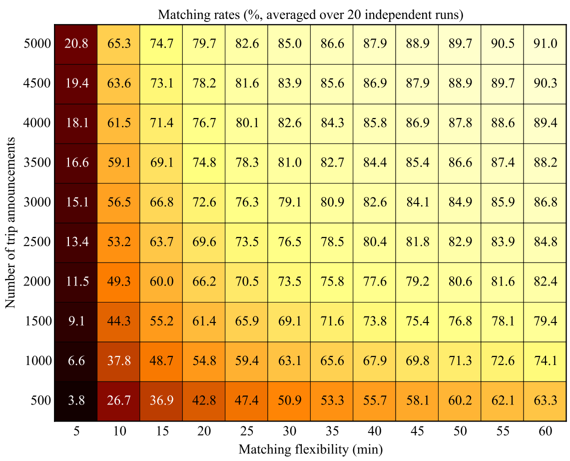
Since the maximum number of possible matches is determined by the minimum of the number of drivers and the number of riders, it is clear that an imbalance in the number of drivers and riders is undesirable. Our experiments have shown that an imbalance in the number of drivers and riders also reduces the value of additional flexibility. Thus, it is important for ride-share providers to explore avenues (e.g., incentive schemes) that results in a driver-to-rider ratio close to 1:1.

1.4.8 Impact of variability in departure times

In this section, we present the results of computational experiments using corridor instances, but generated with a standard deviation in departure times of 15 and 60 minutes. The results are shown in Figures 1.14a and 1.14b, respectively. (The results for the base case, i.e., with a standard deviation of 30, can be found in Figure 1.4a.)



(a) Corridor trips ($\sigma = 15$)



(b) Corridor trips ($\sigma = 60$)

Figure 1.14: Average matching rates for different combinations of system-wide matching flexibility and system density

Comparing the results from the three different departure time distributions, i.e., (1) $\sigma = 15$ resulting in a range of 60, (2) $\sigma = 30$ resulting in a range of 120, and (3) $\sigma = 60$ resulting in a range of 240, we see, not surprisingly, that the earliest departure time distribution impacts the matching rate, especially for low levels of matching flexibility, but that the trends are similar: (1) for a given level of matching flexibility, the average matching rate increases with the number of trips in the system, but the (marginal) increases diminish, and (2) an increase in the number of trips in the system is more beneficial at low levels of matching flexibility and an increase in matching flexibility of more beneficial with a smaller number of trips in the system. In all cases, we see that a very low level of matching flexibility severely limits the ability of the system to establish matches.

Looking at the results in more detail, we see that the value of σ has an impact on the level of matching flexibility above which we see little marginal improvement. For example, with 2,500 participant announcements in the system, the difference between the matching rates with 30 and 60 minutes of matching flexibility is 1.4 p.p. for $\sigma = 15$, 7.9 p.p. for $\sigma = 30$, and 15.0 p.p. for $\sigma = 60$.

1.4.9 Variability in results

Hereunder, we briefly discuss the variability in our results. We look at how sample size influences the distribution of sample matching rates. To analyze this, we fix scheduling flexibility to 20 minutes and perform 100 runs in which we increase the sample size from 100 to 2,000 participants in increments of 100. We visualize the results in the box and whisker plot depicted in Figure 1.15. The whiskers' caps represent the first and the ninety-ninth percentile. Outliers are marked with stars.

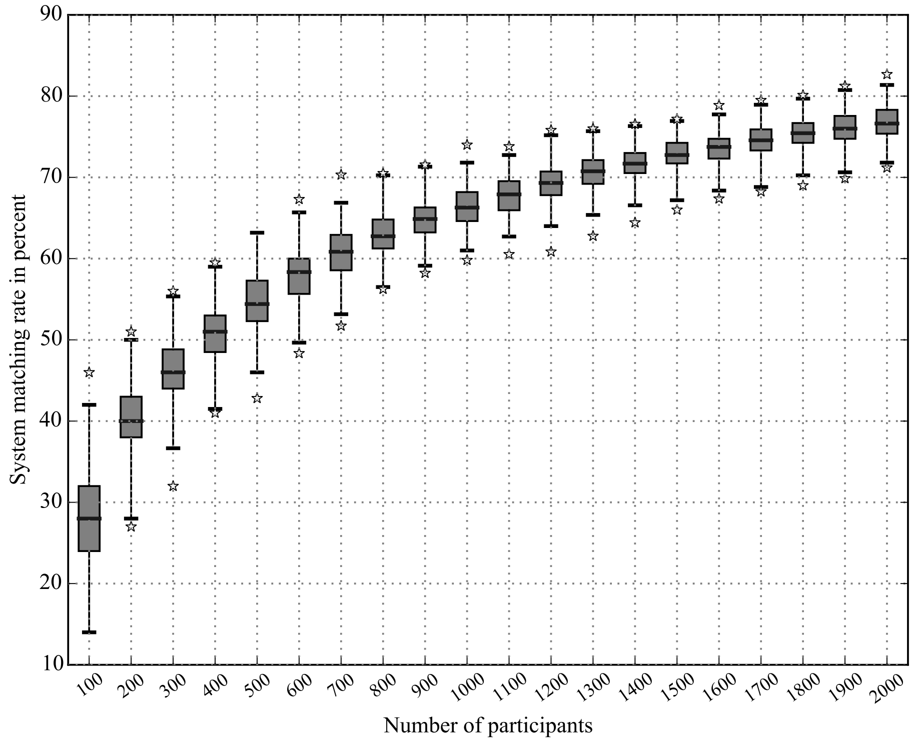


Figure 1.15: Results of System-wide Matching flexibility experiment: Variability in results

With 100 participants, we may expect a matching rate of a little under 30%, but the outcome is quite uncertain, since the values stemming from different random runs range between 5% and 45%. With 1,000 participants, the outcome is more predictable, with a mean value of approximately 66%, and both the minimum and the maximum value within approximately $\pm$ 8%. The variance continues to slowly decrease with increasing sample size. We also examined histograms for individual sample sizes and did not find any noteworthy abnormalities. It seems reasonable to assume these matching rates are normally distributed and that the variance falls with sample size.

This analysis also has some implications for ride-share providers: High variability in performance may be expected when the number of participants is small (up to $\pm$ 15 p.p. with 100 participants) and substantial variability may be expected even with 2,000 participants (app. $\pm$ 6 p.p.).

1.5 Matching the unmatched

As discussed in the introduction, the primary purpose of our investigation is to generate insights that ride-share providers can use to decide if, and if so how, to use incentives to increase matching rates. Our approach in this section is different from the approach taken in the previous sections: we start from the set of matches obtained using default flexibility parameter settings and investigate strategies aimed at increasing the number of matched participants. This leads to a better understanding of the type of flexibility that should be targeted to increase the effectiveness of a ride-sharing system.

For a given instance, we start by determining the matching rate when the default values of the flexibility parameters are used. Next, we establish the additional flexibility that is required to increase the matching rate by a certain number of p.p. (i.e., the increase in the default flexibility parameter values required to achieve the desired increase in matching rate on a participant by participant basis). We analyze the additional flexibility required in terms of the number of drivers and riders that need to be more flexible, in what way these drivers and riders need to be more flexible, and how much more flexible they need to be.

More specifically, we determine the matching rate M with default detour and matching flexibility parameters, i.e., $c_f = 0.25$ and $f_s = 20$, and establish the additional flexibility that is required to increase the matching rate from M to M'. That is, we determine a set of drivers and riders that will experience a detour or matching flexibility that is higher than the default (or both) in order to be able to increase the matching rate to M'. We set the maximum acceptable detour and matching flexibility to $c_f = 0.5$ and $f_s = 60$, respectively.

To establish the *minimum* additional flexibility required to increase the matching rate by a given number of percentage points, we consider two objectives: (1) minimizing the number of participants required to be more flexible, and (2) minimizing the total number of minutes of additional flexibility required. This is accomplished by enumerating all feasible matches of drivers and riders, i.e., those with a matching flexibility less than 60 minutes, detour flexibility less than 0.5, and positive distance savings, and assigning a weight (or penalty) to each match based on the deviation from the default flexibility parameters. For example, if a rider in a match needs to depart 30 minutes after his earliest departure time, then 10 minutes is included in the weight of that match when we are minimizing the total number of minutes of additional flexibility (because the default matching flexibility is 20 minutes) and a weight of one is assigned to that match when we are minimizing the number of participants that need to be more flexible.

Our default optimization approach is to solve a side-constrained minimum weight matching problem, where the side constraint forces the matching rate to be at least a given percentage (Objective (1)). However, because there may be many alternative optimal solutions when the objective is to minimize the number of participants that need to be more flexible, we solve a subsequent side-constrained minimum weight matching problem in which we have two side constraints, one forcing a matching rate and one limiting the number of participants that need to increase their flexibility, and where the weight of a match represents the additional minutes of flexibility required (Objective (2)). We also report results when the order of objectives in the objective hierarchy is reversed, i.e., first optimize with respect to Objective (2) and then with respect to Objective (1).

Figure 1.16 presents the results of an experiment in which we examine how much additional flexibility is required to increase the matching rate obtained with default flexibility parameter values for a corridor instance with 1000 trip announcements. The results represent averages over 10 such instances. The average matching rate with default flexibility parameters is 66%. More specifically, Figure 1.16 shows how much additional flexibility is required to match up to 200 additional participants in increments of 10. Figures 1.16(a) and 1.16(b) show the (average) number of drivers and riders that need to increase their base flexibility when using the two objective hierarchies. Figure 1.16(c) displays the average number of additional minutes of flexibility required per participant with the error bars denoting the 95% confidence interval.

The results show that up to one hundred additional matches can be created by increasing the base flexibility of a relative small number of participants by only a few minutes. Especially when we minimize the number of participants that need to become more flexible (Figure 1.16(a)) typically only one participant has to become more flexible to create an additional match between two participants. More importantly, we see that primarily drivers have to become more flexible to enable new matches. As expected, the results for the two objective hierarchies illustrate that there are different ways to facilitate additional matches, i.e., fewer participants with larger deviations or more participants with smaller deviations.

Next, we look at the type of additional flexibility that is required to increase the number of matches from 345 (the number of matches obtained with default flexibility parameter values) to 445 for a corridor instance with 1000 trip announcements where we minimize the number of participants required to be more flexible. Figure 1.17 shows, for each of the 128 participants that need to be more flexible, how and by how much they need to be more flexible. More specifically, we report for all participants involved in matches involving additional flexibility, the breakdown of their used flexibility, i.e., the service time τ, the waiting time at their origin, any additional waiting time at their origin (in excess of the default matching flexibility), the detour time, and any additional detour time (in excess of the default detour flexibility). The latter two values only apply to drivers.

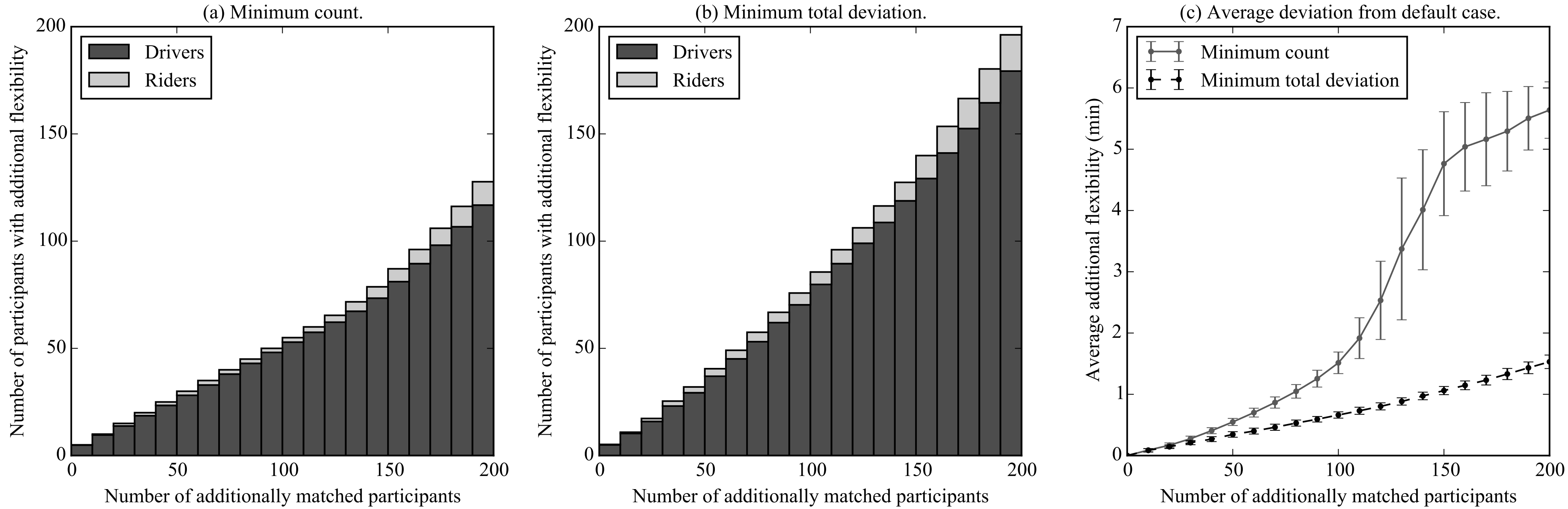


Figure 1.16: Analysis of deviations from base case flexibility that are necessary to match between 10 and 200 additional participants in a corridor instance with 1,000 trips (Averages over 10 independent runs)

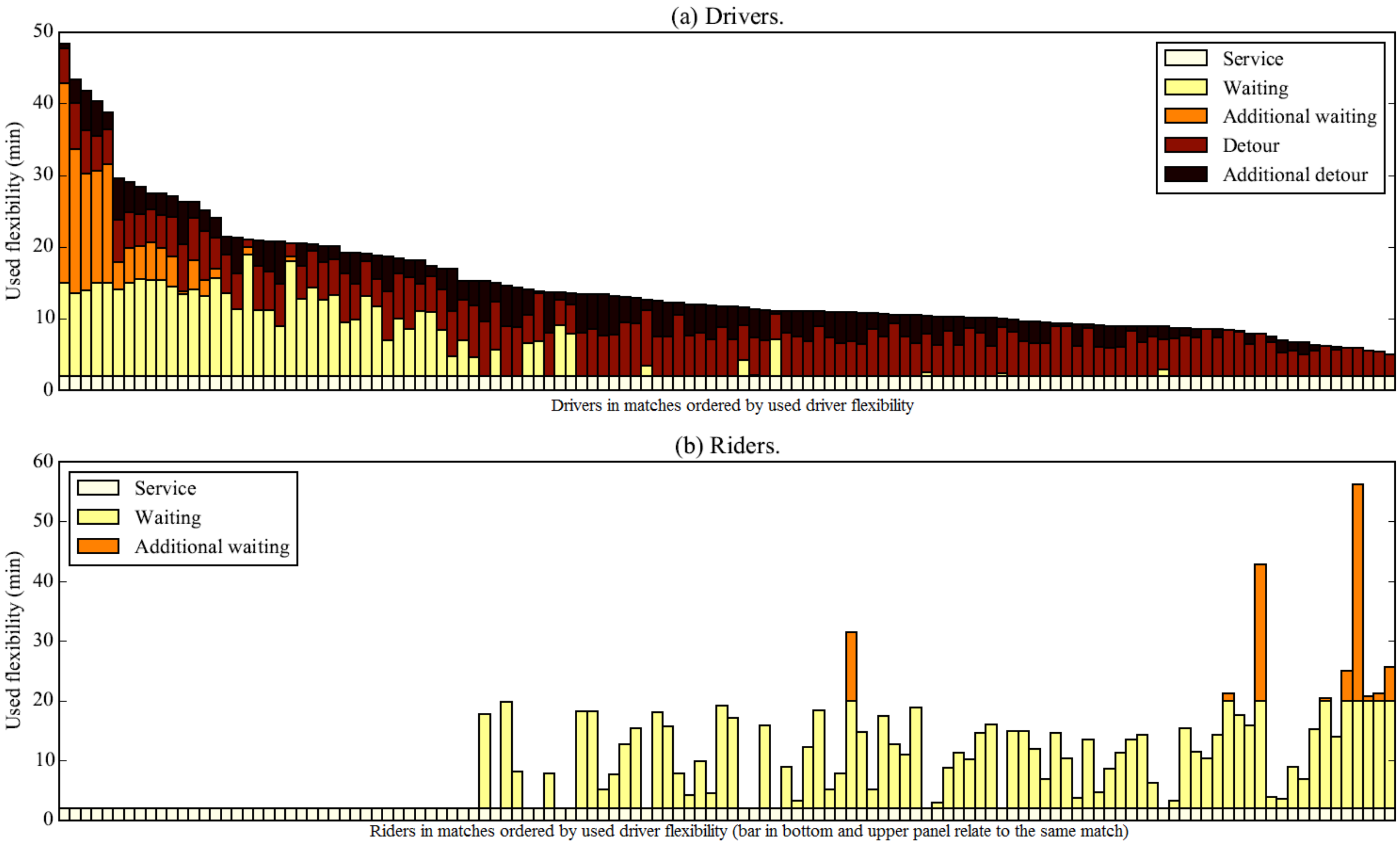


Figure 1.17: Breakdown of the used flexibility for matches involving additional flexibility for an increase of 20 p.p. (Matches in top and bottom figure are ordered based on total used flexibility by the driver. An individual bar in upper pane represent the breakdown of the flexibility used by the driver in a match. The bar in the lower pane that is horizontally alligned with the driver bar represents the breakdown of used flexibility for the rider in that match.)

The results show that in almost all matches that require additional flexibility, additional driver detour flexibility is required. Additional matching flexibility is required in only 23 matches. Fewer than 10 matches do not require any additional flexibility on the part of the driver. Hence, driver flexibility plays a fundamental role in establishing additional matches. Maybe most importantly, we see that the additional detour and waiting time required tends to be small, rarely more than a few minutes. More than 20 minutes of additional matching flexibility is used in fewer than 10 matches.

We continue by considering a setting in which only drivers can increase their base flexibility and a setting where only riders can increase their base flexibility. These two settings serve to evaluate the relative importance of the drivers versus the riders in establishing additional matches. We repeat our first experiment from the beginning of this section (results displayed in Figure 1.16). We use the default objective hierarchy. Again, we attempt to increase the number of matched participants from 10 to 200. The important difference with our previous experiment is that in the settings we examine here, it is not always possible to increase the number of matched participants up to 200. We track for each setting and each of the 10 instances when this happens.

Figure 1.19 shows the results for the case when only drivers increase their base flexibility and Figure 1.18 shows the results for the case when only riders increase their base flexibility. In both figures, the left pane visualizes the relationship between the number of additionally matched participants and the number of participants that use additional flexibility for each of the three settings. The right pane visualizes the relationship between the number of additionally matched participants and the average additional flexibility in minutes per match.

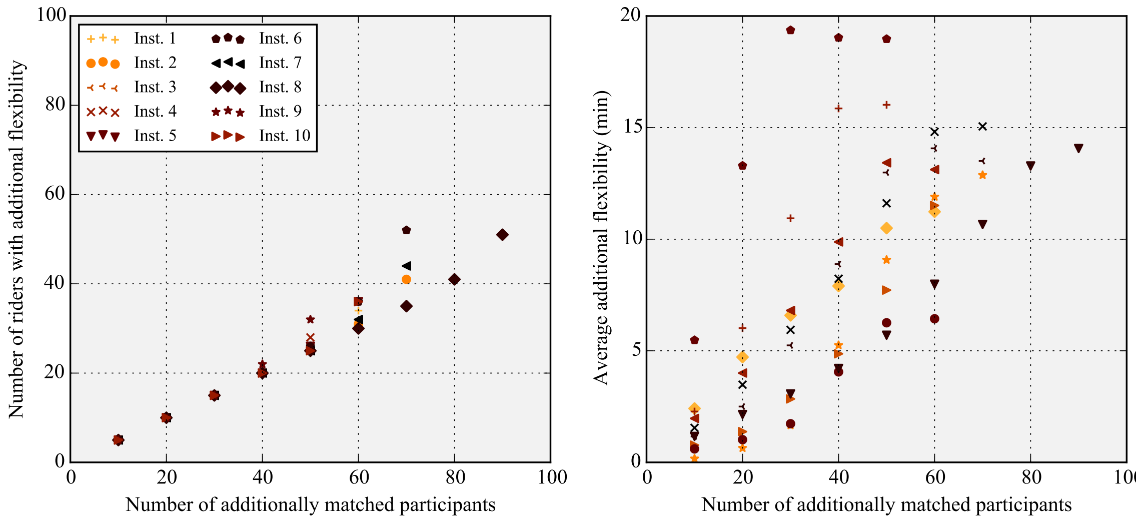


Figure 1.18: Analysis of the number of participants with deviation from base case flexibility that are necessary to match between 10 and 200 additional participants in a corridor instance with 1000 participants (Based on 10 independent runs)

When only riders are considered, we see that it is only possible to increase the number of matched participants by up to 90 in the best case. For several instances, no more than 70 additional participants can be matched. Furthermore, we see that, on average, a lot more additional flexibility has to be used to achieve these increases (up to 20 minutes of additional matching flexibility on average). This demonstrates that while it may help in certain cases that a rider is ready to accept a (much) later departure/arrival time, this alone is not an effective strategy of increasing the system matching rate. It also appears that much more matching flexibility has to be used on average than in the case of drivers.

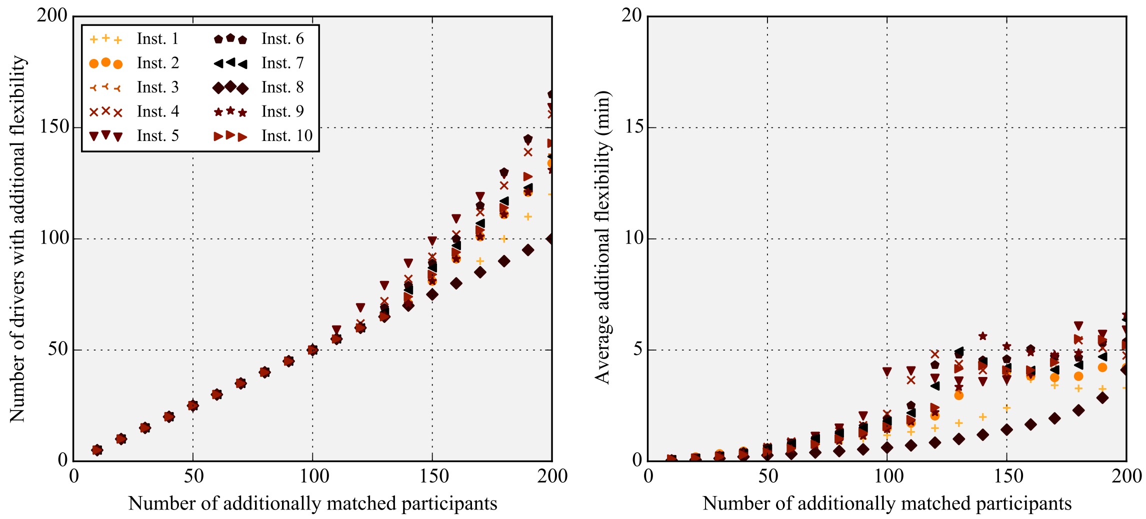


Figure 1.19: Analysis of total additional flexibility in minutes that is necessary to match between 10 and 200 additional participants in a corridor instance with 1000 participants (Based on 10 independent runs)

When only drivers are considered, there is hardly any difference to the setting with flexible riders and drivers. This is understandable since riders played a very marginal role in that setting. It is possible to achieve an increase in the number of matched participants of up to 200 in all of the 10 instances. Interestingly, there is also no significant impact on the average additional flexibility in minutes per match or on the number of participants that use additional flexibility, even for an increase of 200 participants (compared to 1.16). This demonstrates that the same outcomes can be expected even if only the drivers are more flexible.

So far, we have assumed that, in principle, the flexibility of all participants in an instance can be increased – even the flexibility of those participants that can be matched with default flexibility parameters. In practice, this may not be possible or desirable. In our final experiment, we investigate the effects of a policy that does not allow a degrading in the quality of matches for participants that can matched with default flexibility parameters. (This is accomplished by fixing participants that can be matched with default flexibility parameters in the matching and by not considering matches that require these participants to increase their flexibility in subsequent optimizations.)

We see that with fixed default case matches, it is not possible to increase the number of matched participants beyond 170. In certain instances, even an increase beyond 110 additionally matched participants is not possible. Interestingly, the number of participants that need to use additional flexibility is very similar to case (a). However, much more additional flexibility has to be used on average compared to (a), especially for increases above 50 participants. This points to the fact that is a bad idea to split the matching problem into a tranche of matches that can be established with base case flexibility and another tranche within which lower quality matches are established. Such a strategy does guarantee that a a group of users benefits from high quality matches, but this comes at the expense of the other group of users that is much worse off.

The results of this section have important implications. In terms of the additional flexibility

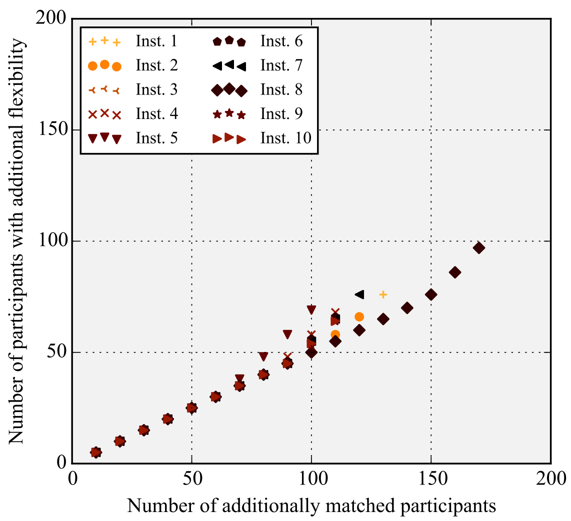


Figure 1.20: Analysis of total additional flexibility in minutes taht is necessary to match between 10 and 200 additional participants in a corridor instance with 1000 participants (Based on 10 independent runs)

in minutes that is needed, it is typically easiest to establish (additional) matches by trying to make drivers accept longer detours. Our results show that, often, relatively small increases in the detour flexibility of a specific driver can make a match feasible. On the other hand, while there are cases when postponing the departure of a rider can help, this is not very effective on average.

1.6 Using financial incentives for drivers

In the previous sections, we have demonstrated that it is typically easiest to ensure a high matching rate in a ridesharing system by motivating drivers to accept longer detours. In this final section, we present a model and an experiment that explicitly considers payments by the riders and remuneration for the drivers. We present a ride-matching model that can provide financial incentives to drivers who are willing to accept longer detours.

We consider the business model that is used by many ridesharing providers (e.g. Car.ma, Split.us, BlaBlaCar.com): The system pools all payments made by the riders, takes an arbitrary commission for its services (e.g. 1%) and redistributes the rest to the drivers. We proceed by explaining the assumptions of our model and introduce several additional parameters that were not used in the previous sections.

In this section, we allow matches with negative distance savings. That is, for driver $i \in D$ matched to rider $j \in R$, the difference between the joint individual direct distances and the total distance of the driver in the match can be negative. We relax this constraint because we want to try to match as many riders as possible. Note that such matches may still be beneficial from a transportation policy point of view since a matched rider will not need to use his own vehicle, which may free up parking space in the city center or another area with limited parking space availability.

Riders pay a base fare p^b that is fixed for each ride. In addition, each rider pays a variable fare per ride-share mile p^v. Drivers are remunerated according to several rules. Each driver receives a variable fare per mile for the shared part of his trip (this is the part of the trip in which the driver and rider ride together). We denote this fare as f^s. Driver detours are treated separately. The rationale for treating a driver detour differently than the shared part of the trip is that driver detours are inconvenient for the drivers and undesirable from a transportation policy perspective. It is thus desirable to minimize them. We compute the detour distance δ^d_{ij} for driver $i \in D$ serving rider $j \in R$ as $d_{o_i,o_j} + d_{o_j,d_j} + d_{d_j,d_i} - d_{o_i,d_i}$. The driver is remunerated with the full officially allowed standard mileage rate for this part of the trip. We denote this rate per mile as f^r. The driver also needs to extend his trip duration by the detour duration – each driver is given remuneration that is proportional to the duration of the detour. We denote this last component as f^t. The detour duration for driver $i \in D$ serving rider $j \in R$, which we denote as δ^t_{ij} is defined as $t_{o_i,o_j} + t_{o_j,d_j} + t_{d_j,d_i} + \tau - t_{o_i,d_i}$. For each match in the set of feasible matches F, we compute the remuneration for driver i associated with this match and denote it as r_{ij}, where $r_{ij} = f^s d_{o_j,d_j} + \delta^d_{ij} f^r + \delta^t_{ij} f^t$ and the payment by rider j is denoted as p_{ij}, where $p_{ij} = p^b + p^v d_{o_j,d_j}$. Finally, we assume the system operator may choose to take a commission from the sum of the gathered payments from the riders in the form of a fixed percentage. Alternatively, he may choose to add his own funds to the sum of gathered payments in order to pay for driver detours and increase the matching rate. The former case may be feasible in a mature ridesharing service while the latter case could be used to facilitate the matching process in the launch phase of a ridesharing system. We define a coefficient O_c, where a value of 0.99 indicates a 1% commission and a value of 1.01 indicates that the operator adds 1% to the total sum of driver payments in an attempt to increase the system matching rate.

As in the previous experiments, we match the drivers and riders by solving a bipartite matching problem. We use one optimization phase in which we maximize the number of matches. We add an additional, market balancing constraint to the optimization model presented in expressions (1.1) – (1.4). This constraint assures that the sum of rider payments multiplied by the operator coefficient is greater than or equal to the total remuneration paid to drivers:

$$\sum_{(i,j)\in F} x_{ij} r_{ij} \leq \sum_{(i,j)\in F} O_c(x_{ij} p_{ij}) \tag{1.5}$$

For this final experiment, we use a more realistic setting based on the travel demand model for the metropolitan Atlanta region. The instance generation methodology is explained in detail in Chapter 2, Section 2.5.1. The only difference is that we do not use meeting points in this particular experiment. Table 1.2 summarizes our assumptions and the parameter values used.

We perform the experiment as follows. We generate random rideshare instances with the parameter values described in the above table. For these instances, we first compute the results for the base case in which we respect all constraints that relate to rider and driver flexibility. We solve a hierarchical optimization model as defined in Section 1.2 and do not consider the market balancing constraint in the base case. We then relax the assumption that the constraints regarding driver detour and matching flexibility need to be respected and solve the optimization model with the market balancing constraint (1.5). Table 1.3 compares the metrics obtained for the default model and the model that uses driver incentives.

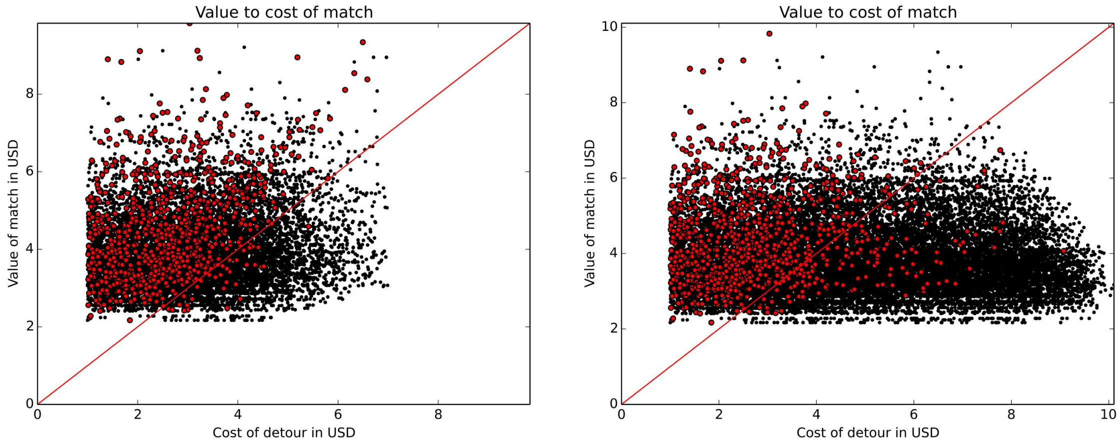
Table 1.2: Assumptions of experiment

Trip pattern:	suburb to center
Number of participants:	2854
Number of drivers:	1450
Number of riders:	1404
Driving speed:	20 mi/h
Detour flexibility:	0.25
Matching flexibility:	20 min
Scheduling flexibility:	30 min
Service time:	2 min
Base rider fare – p^b:	USD 1
Variable rider fare – p^v:	USD 0.5 per mile
Driving cost remuneration for shared trip – f^s:	USD 0.15 per mile
Driving cost remuneration for detour – f^r:	USD 0.5 per mile
Driver variable detour remuneration – f^t:	USD 0.25 per minute
Operator coefficient – O_c:	1.0

Table 1.3: Results of experiment with financial incentives for drivers

	Default	Incentives
Matching rate (%)	61.03	70.02
Total mileage savings	5107.87	5523.77
Number of matches with violation	0	201
Average detour distance (mi)	1.08	1.16
Average detour time (min)	8.32	8.62
Average rideshare length (mi)	7.01	6.74

As can be seen, it was possible to increase the matching rate by approximately 9 p.p. by using incentives for drivers. This increase was possible because the constraints that relate to the driver detour and matching flexibility were violated in 201 matches, representing roughly 20% of all established matches. The two scatter plots in Figure 1.21 provide us with more information about the change that occurred between the default case and the case with incentives. Each scatter diagram maps all matches included in the optimization (black dots) and all matches in the optimal solution (red dots) according to the cost of the driver detour associated with a match in USD (i.e. the remuneration the system provides to the driver for the detour) and the value of the match in USD (i.e. the total price the rider pays for the ride associated with a match).



(a) Distribution without driver incentives

(b) Distribution with driver incentives

Figure 1.21: Scatter diagram mapping all matches included in the optimization (black dots) and all matches in the optimal solution (red dots) according to the remuneration for the driver detour associated with a match (in USD) and according to the value of the match (in USD)

The 45-degree red line represents the limit between matches for which the total payment by the rider is greater than the cost of the driver detour (upper left corner) and matches for which the total payment by the rider is less than the cost of the driver detour (bottom right corner). The scatter diagrams very clearly demonstrate the difference between the default case and the case with driver incentives. It shows that employing incentives and the market balancing constraint redistributes the value of matches: It uses high net values of certain matches to subsidize less attractive matches and thereby increases the matching rate. In this section, we have presented a ride-matching model with financial incentives for drivers. Our results indicate there is a potential to increase the matching rate in a ridesharing system by using such incentives for drivers to increase their detour. We note that more research is needed to establish the implications and viability of such a scheme in the context of ridesharing.

1.7 Conclusion

Our computational study clearly demonstrates (and quantifies) the impact of participant flexibility on the performance of a single-driver, single-rider ride-sharing system (in terms of the matching rate achieved). The study shows that participant flexibility plays a key role in easing the matching process, especially in systems with low participation rates.

In order for dynamic ride-sharing to work, drivers and riders need to be flexible in terms of departure and arrival times (at least 10 to 15 minutes depending on origin and destination locations), but, most importantly, drivers need to be flexible in terms of the detour that they are willing to make.

We hope that the insights generated by our study can be used by ride-sharing system providers to design effective incentive schemes to increase system performance, where we interpret the term incentive scheme broadly, e.g., meaning anything from providing a participant with information on the likelihood of being matched or on the increase in likelihood of being matched when the participant's flexibility is increased by a certain amount, all the way to providing monetary benefits to participants that increase their flexibility.

2 THE BENEFITS OF MEETING POINTS IN RIDESHARING SYSTEMS

2.1 Introduction

As has been demonstrated in Chapter 1, limited flexibility in the participants' itineraries and schedules is a major challenge in ride-sharing. It may result in many drivers and riders not finding a match. In this chapter, we investigate the benefits of introducing meeting points to take advantage of any flexibility on the part of the riders. Meeting points allow the construction of routes with smaller detours, while maintaining a satisfactory level of service for the riders. Riders may be picked up and dropped off at meeting points that are within an acceptable distance from their origin or destination. (A pick up or drop off can, of course, still take place at the rider's origin and destination as well.) By exploiting the rider flexibility, more matches may be found. Furthermore, meeting points allow a driver to be matched with multiple riders without increasing the number of stops on the driver's trip.

Consider the example depicted in Figure 2.1 with driver $d1$ and rider $r1$ and two meeting points $m1$ and $m2$, where the number above an arc represents the time it takes to travel between the nodes, and where the driver is willing to accept an increase in trip time of at most five minutes.

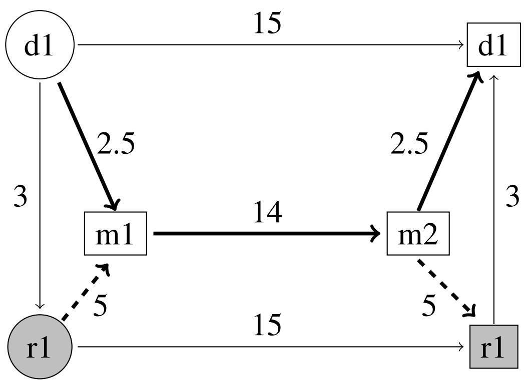


Figure 2.1: Rider (grey) and Driver (white) traveling from Origin (circle) to Destination (square) via Meeting Points

Without the use of meeting points, a match between $d1$ and $r1$ is not feasible because the required increase in trip time (6 min) exceeds the driver's limit. If, however, the rider is willing to walk 5 minutes to and from a meeting point, a feasible match between $d1$ and $r1$ is possible, because $d1$ has to make a smaller detour. (The rider's trip will be 9 minutes longer than if he drove by himself, but he will loose no time finding a parking space and he will not be using his own car.)

Note that the savings in driving distance in the example above is about 37% (where the savings in driving distance is obtained by comparing the driving distance when both participants drive by themselves to the driving distance when they are matched, i.e., 30 versus 19 in the example above). It is customary to consider a match distance feasible if there is a positive driving distance savings and also to measure the value or benefit of a match by the

This chapter is based on Stiglic et al. (2015a).

driving distance savings. Capturing the value of a match in this way may not be perfect, but it is pragmatic. Not all riders for which no match can be found will drive themselves. Some may ask a friend to drive them or use public transportation; others may not undertake the planned trip at all. Ride-sharing has the potential to provide increased mobility to those that do not own their own vehicle, but it is hard to capture and quantify this benefit. Therefore, we, as has been done in previous studies, focus on driving distance savings.

Meeting points can also result in more matches because they allow a driver to be matched with multiple riders without extra stops. Consider the example depicted in Figure 2.2 with driver $d1$ and riders $r1$ and $r2$ and two meeting points $m1$ and $m2$, where the number above an arc represents the distance between the nodes. (As before, the dashed lines represent walking of riders to and from meeting points.)

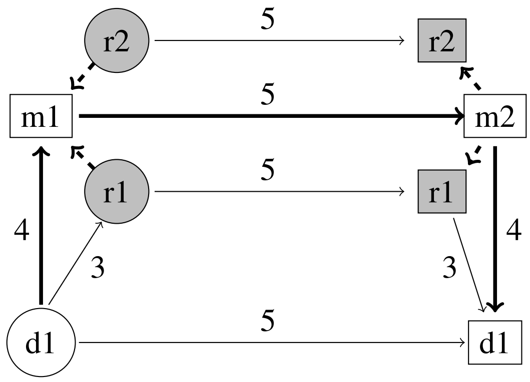


Figure 2.2: Riders (grey) and Driver (white) traveling from Origin (circle) to Destination (square) via Meeting Points

None of the matches between $d1$ and $r1$ and $d1$ and $r2$ (with or without a pickup at $m1$ and/or a drop-off at $m2$) leads to positive savings in driving distance. However, a multi-rider match between driver $d1$ and riders $r1$ and $r2$ (with a pickup at $m1$ and a drop-off at $m2$) does lead to positive driving distance savings (15 versus 13).

In the setting we consider in this chapter, a driver can be matched with multiple riders, as long as the capacity of his vehicle is not exceeded, and the riders are picked up at the same meeting point at the same time and dropped off at the same meeting point (at the same time). Allowing only one pickup and one drop-off point per shared ride ensures that the trips are easy to execute and minimize the inconvenience for the driver; additional stops and detours increase the inconvenience for participants and the risk of complications arising during execution. Multi-rider matches may have other, harder to quantify, benefits: waiting for a ride and sharing a ride as a group may increase the feeling of safety and social cohesion and might thereby improve the image of ride-sharing.

The viability of introducing meeting points in a ride-sharing system may depend on a variety of circumstances, e.g., the availability of safe locations for meeting points, the prevailing weather conditions, and the cultural attitudes towards transportation. In many suburban areas in the U.S., for example, it may be difficult to find safe locations within easy walking distance from a person's home. In regions where adverse weather conditions occur frequently, the prospect of having to wait outside for a pickup may not be appealing. In places where people are used to live and commute in climate controlled environments, it may be difficult to overcome initial reluctance towards walking to meeting points. However, meeting points are already an integral component of some existing ride-sharing systems, e.g., slugging or casual carpooling, where passengers form (slug) lines at specific locations and wait for rides (the incentive to pick up riders is typically that it allows drivers to use faster HOV lanes and/or share the cost of tolls), and long-distance ride-sharing, which tends to be scheduled in advance and has less restrictive requirements regarding meeting place and time. The locations that can be used for meeting points varies by region or country. For instance, in Slovenia, bus stops and gas stations are commonly used as meeting points. However, the use of bus stops may be perceived as unsafe and inappropriate in many parts of the US (or may even be illegal). It is conceivable that neighborhood pools, fast food restaurants, or coffee shops can act as meeting points in the US. Park & ride facilities and entrances to well-known institutions/buildings are additional options.

In this chapter, we discuss the design and implementation of an algorithm that optimally matches drivers and riders (based on an extension of the traditional bipartite matching formulation) in large-scale ride-sharing systems with meeting points. We perform an extensive simulation study (based on real-world traffic patterns) to assess the benefits of meeting points. The results demonstrate that meeting points can significantly increase the number of matched participants as well as the system-wide driving distance savings.

The chapter is organized as follows. In Section 2, we provide an overview of related literature and explain how we build upon it. In Section 3, we introduce notation and a mathematical model of the ride-share optimization problem with meeting points. In Section 4, we detail the solution approach we have developed for this optimization problem. In Section 5, we motivate and discuss the simulation study we have conducted and we present and analyze its results. Finally, in Section 6, we summarize the key findings and suggest directions for future research.

2.2 Related literature

Agatz et al. (2011) represent the single rider, single driver ride-share matching problem by a max-weight bipartite matching problem. They explore different approaches to match drivers and riders in real-time and investigate the impact of different service characteristics of the system. Their study shows that the success of a ride-sharing system strongly depends on the participation density, e.g., the number of participants per square mile, and that a minimum participation density is required to ensure a stable system (in which participants do not leave the system because they repeatedly fail to find a match). Wang et al. (2014) extend this analysis by investigating the trade-off between matchings that are optimal for the system as a whole and matchings that are optimal for each of the participants in the system. They introduce the concept of stable matches in the ride-sharing setting. Lee and Savelsbergh (2015) consider the employment of a small number of dedicated drivers to serve riders that would otherwise remain unmatched. The aim is to guarantee a certain service level (i.e., fraction of riders that is matched) thereby ensuring a stable system.

Another way to increase the number of riders that find a match is to allow riders to transfer between different drivers, i.e., allowing a rider to travel with more than one driver to reach his destination. Herbawi and Weber (2011c) consider a multi-hop ride-sharing problem in which drivers do not deviate from their routes and time schedules. As such, the drivers' ride-share offers form the transportation network for the rider, who has to find a route that minimizes costs, time, and number of transfers. Drews and Luxen (2013) extend this work by also allowing reasonable detours and time deviations for the drivers. While rider transfers might be acceptable to a driver, they are inconvenient for a rider as they may involve waiting times between rides and they increase the risk of anything going wrong during execution.

In contrast to the existing work, we explicitly consider a setting in which riders are willing to walk to and from meeting points to facilitate easy pick up and drop off. We are aware of only one paper that considers meeting points in a related context. Kaan and Olinick (2013) consider vanpooling, in which up to 15 people share a van to travel to a common location. The commuters in the vanpool drive to a park-and-ride location and then ride together to a final location. The authors consider the problem of assigning commuters and vans to park-and-ride locations, and present a mixed integer programming formulation and several heuristics for its solution. The main difference with our setting is that, in the end, the vans provide scheduled transportation. The setting is also simpler in that all riders travel to one common final location.

The use of pickup locations is not unique to the ride-share setting. It is prevalent in the school bus transportation in which students in urban areas are assumed to walk to a bus stop from their homes to take the bus to school. The selection of bus stops and the assignment of students to bus stops is a subproblem in the school bus routing problem that is related to our work. While several papers address the school bus routing problem, only few papers explicitly consider the selection of bus stops (Park and Kim, 2010). Some recent papers have integrated the selection of bus stops with the bus route generation using both exact (Riera-Ledesma and Salazar-Gonzalez, 2013) and heuristic methods (Schittekat et al., 2013).

Ride-sharing, especially when incorporating meeting points, requires the coordination of rider and driver itineraries. This is related to the area of routing problems with synchronization constraints. This line of research deals primarily with vehicle routing problems in which more than one vehicle may be required to fulfill certain tasks. For a recent review, see Drexl (2012). In general, vehicle routing problems with synchronization constraints are difficult to solve so heuristics are most commonly used, see for example Goel and Meisel (2013) and Meisel and Kopfer (2014).

The ride-sharing setting we consider has a simple routing structure, because we allow the drivers to make only one pickup and one drop-off. The number of feasible driver-rider matches is also relatively small due to capacity and time constraints. As a consequence, we can enumerate the feasible routes and represent the problem of optimally routing drivers as a matching problem. This allows us to use an exact approach to solve even large instances of the ride-share problem to optimality.

The research that is, potentially, the most relevant to our own also considers ride-sharing with meeting points, but focuses on how to find the best meeting points once a match between a driver and a rider has been established (Aissat and Oulamara (2014)). This is especially relevant in case of matching a driver and a rider for recurring trips. The authors place no restrictions on the locations of the pickup and drop-off points relative to the origin and destination, respectively, of the rider's trip and, thus, make no assumptions on how the rider might reach the pickup point and on how the rider might reach the destination.

2.3 Problem Definition

We are provided with a set of trip announcements S. With each trip announcement $s \in S$ is associated, an origin location o_s and a destination location d_s as well as an earliest departure time e_s and a latest arrival time l_s. We assume the departure times of participants are somewhat flexible so that the difference $l_s - e_s$ is greater than the travel time from origin to destination. The set of announcements S can be partitioned into $D \subset S$, the set of trip announcements by the drivers, and $R \subset S$, the set of trip announcements by the riders. Each driver $i \in D$ also specifies a maximum trip duration T_i, which implies the extra time the driver has available to accommodate a ride-share, and a vehicle capacity C_i, which gives the maximum number of people the driver's vehicle can accommodate. Each rider $j \in R$ also specifies a maximum distance m_j that he is willing to walk to and from a meeting point. (For presentational convenience, we will sometimes also use o_i and d_i to indicate the origin and destination of a driver i and o_j and d_j to indicate the origin and destination of a rider j.)

We denote the distance from location i to j with d_{ij} and the travel time between the two locations by t_{ij}. Furthermore, we denote the set of meeting point locations that can be reached by at least one rider by M. The set of feasible pickup meeting points for rider j is $M_j^p := \{k \in M \mid d_{ko_j} \leq m_j\}$, and the set of feasible drop-off meeting points for rider j is $M_j^d := \{k \in M \mid d_{kd_j} \leq m_j\}$. We introduce the concept of a *meeting point arc a* to denote a combination of a pickup point and a drop-off point. The set of feasible meeting point arcs for rider j is $A_j := \{(k,l) \mid k \in o_j \cup M_j^p,\ l \in d_j \cup M_j^d\}$. Thus, each rider j can be picked up at his origin o_j or a meeting point in M_j^p and dropped off at his destination d_j or a meeting point in M_j^d. Let $A = \bigcup_{j \in R} A_j$. Finally, we denote the service time at each meeting point $m \in M$ by τ_m, i.e., the time needed to get into and out of the vehicle at a pick up or drop-off meeting point.

2.3.1 Definition of a Feasible Match

A match is defined as a combination of driver $i \in D$, a set of riders $J \subset R$, and a meeting point arc $a \in A$. Hence, it can be defined by a triplet (i,J,a). Note that since we do not allow more than one pick-up and one drop-off, in a feasible match (i,J,a), we must have $a \in \bigcap_{j \in J} A_j$. Furthermore, a feasible match implies a unique route for the driver and for every rider. A feasible match (i,J,a) must also have $|J| + 1 \leqslant C_i$ and must satisfy the time constraints of the participants. A match is time feasible if it is possible for all participants to traverse the meeting point arc a at the same time, while respecting the earliest departure times from their origins and the latest arrival times at their destinations and, for the driver, the maximum ride time.

In order to check the time feasibility of a match (i,J,a), with $a = (k,l)$, we construct an implied time window at k for each participant in the match. We denote the implied time window for a participant p (either i or $j \in J$) at k by $[e_p^k, l_p^k]$, where $e_p^k = e_p + t_{o_pk}$ and $l_p^k = l_p - (\tau_k + t_{kl} + \tau_l + t_{ld_p})$. To check the time feasibility of the match, the intersection of the implied time windows has to be non-empty, which implies that we must have

$$\max\left(\max_{j \in J} e_j^k, e_i^k\right) \leq \min\left(\min_{j \in J} l_j^k, l_i^k\right) \tag{2.1}$$

When the above inequality holds, $\max(\max_{j\in J} e_j^k, e_i^k)$ is the earliest time, and $\min(\min_{j\in J} l_j^k, l_i^k)$ is the latest time, at which the shared ride can depart from meeting point k. The maximum ride time for the driver is satisfied, if

$$t_{o_ik} + \tau_k + t_{kl} + \tau_l + t_{ld_i} \le T_i. \tag{2.2}$$

A match between driver i and riders in J on meeting point arc $a = (k,l)$ has an associated driving distance savings of $\sigma_{(i,J,a)}$:

$$\sigma_{(i,J,(k,l))} = d_{o_id_i} - \left(d_{o_ik} + d_{kl} + d_{ld_i}\right) + \sum_{j\in J} \left(d_{o_jd_j} - (d_{o_jk} + d_{ld_j})\right). \tag{2.3}$$

A match (i,J,a) is considered distance feasible when $\sigma_{(i,J,a)} > 0$. Note the walking distances are taken into account in (2.3) to break ties when two or more arcs have similar savings.

2.3.2 Matching Problem

The single rider, single driver ride-share matching problem can naturally be formulated as a maximum weight bipartite matching problem (Agatz et al. (2011)). We extend this formulation to the single driver, multiple rider ride-share matching problem. We note that the formulation introduced below can represent a variety of ride-share matching problems in which a driver can be matched with multiple riders, because the identification of feasible matches and the associated routing is handled in a subproblem. By accommodating single driver, multiple riders matches the formulation becomes a maximum weight bipartite matching problem with side constraints, which, in theory, is no longer solvable in polynomial time, but still solves extremely fast in practice. We note too that maximizing system-wide driving distance savings does not guarantee that a maximum number of participants is matched. Consider, for example, the situation depicted in Figure 2.3 with drivers $d1$ and $d2$ and riders $r1$ and $r2$ and two meeting points $m1$ and $m2$, where the number above an arc represents the distance between the nodes.

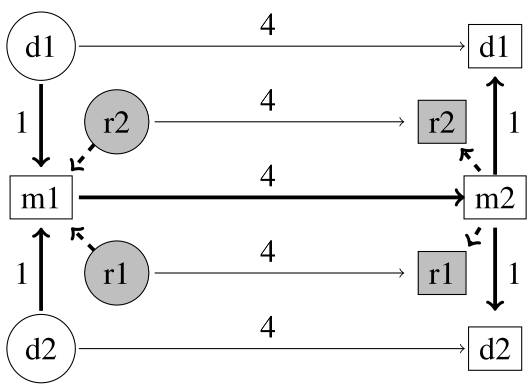


Figure 2.3: Riders (grey) and Drivers (white) traveling from Origin (circle) to Destination (square) via Meeting Points

The maximum driving distance savings is achieved when either $d1$ or $d2$ is matched with both $r1$ and $r2$ (and, thus, one of the drivers will not be matched). By matching driver $d1$

with rider $r1$ and driver $d2$ with rider $r2$, all system participants are matched, but with lower driving distance savings (6 vs 4).

As in Agatz et al. (2011), we create a node for each driver $i \in D$ and each rider $j \in R$ and an edge connecting node i and j if there is a feasible match between driver i and rider j. In addition, we introduce nodes that represent a set of riders J, e.g., a pair of riders, a triple of riders, etc., and introduce an edge connecting driver $i \in D$ and set of riders J, if there is a feasible match between driver i and the set of riders in J. Each edge e has two weights associated with it: number of participants in the match v_e, and maximum driving distance savings σ_e. Note that a particular combination of a driver and a set of riders may have more than one feasible match because there may exist more than one feasible meeting point arc. However, we are clearly only interested in the one with the highest driving distance savings. Figure 2.4 illustrates the bipartite graph for an example with two drivers and two riders. The numbers above the edges denote the number of participants in the match and the associated distance savings. In this example, the match between driver $d2$ and rider $r1$ and the match between driver $d2$ and rider pair $(r1, r2)$ are not feasible. The optimal solution is to match driver $d1$ with rider $r1$ and driver $d2$ with rider $r2$ for a total distance savings of 18.

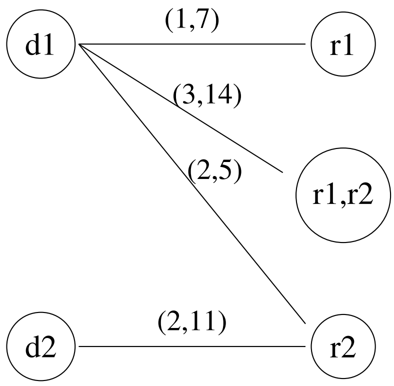


Figure 2.4: Bipartite graph with two drivers (d1 and d2) and two riders (r1 and r2)

Let E represent the set of all edges in the bipartite graph and let the binary decision variable x_e for edge $e \in E$ indicate whether the edge is in an optimal matching ($x_e = 1$) or not ($x_e = 0$). Furthermore, let E_i and E_j represent the set of edges in E associated with driver i and rider j, respectively. Then, the single driver, multiple riders ride-share matching problem with the objective of maximizing the number of matched participants can be formulated as the following integer program:

$$\max z_1 = \sum_{e \in E} v_e x_e \tag{2.4}$$

subject to

$$\sum_{e \in E_i} x_e \leq 1 \quad \forall i \in D, \tag{2.5}$$

$$\sum_{e \in E_j} x_e \leq 1 \quad \forall j \in R, \tag{2.6}$$

$$x_e \in \{0, 1\} \quad \forall e \in E. \tag{2.7}$$

Objective function (2.4) maximizes the number of matched participants. Constraints (2.5) and (2.6) assure that each driver and each rider is only included in at most one match in an optimal matching, respectively.

To obtain a matching that maximizes the driving distance savings, the objective should be replaced by

$$\max z_2 = \sum_{e \in E} \sigma_e x_e. \tag{2.8}$$

Since both objectives, i.e., maximizing the number of matches and maximizing the driving distance savings, are relevant in the ride-sharing context, we take both objectives into account in a hierarchical fashion, where we consider z_1 as the primary objective and z_2 as the secondary objective. We first solve (2.4) subject to (2.5) - (2.7). Let z_1^* be the number of matched participants. We then solve (2.8) subject to (2.5) - (2.7) plus the additional constraint $\sum_{e \in E} \nu_e x_e \geq z_1^*$. This type of hierarchical approach is known in the literature as lexicographical goal programming (Ignizio (1976)).

Finally, we observe that it is possible to extend the model with a set of participants with flexible roles F similar to Agatz et al. (2011). The nodes corresponding to flexible participants may appear on either side of the bipartition, but can never be connected with an edge. The model can be extended by introducing sets E_f representing edges in E associated with flexible participants and adding another set of constraints $\sum_{e \in E_f} x_e \leq 1, \forall f \in F$.

2.4 Solution Approach

When the number of participants and of meeting points is large, it can become computationally prohibitive to determine the time and cost feasible single matches (especially since multi-rider matches have to be considered as well). Therefore, we have implemented this component of the solution approach carefully and efficiently. For expository purposes, we assume that the locations are in a Euclidean plane, that distances are Euclidean, and that traveling (either walking or driving) occurs at a constant speed. However, most of these assumptions can relatively easily be relaxed so as to cover more realistic settings.

2.4.1 Determining Feasible Meeting Points for a Rider

We store the set of meeting points in a $k-d$ tree (Bentley (1990)). $K-d$ trees support Euclidean distance nearest neighbor search, n nearest neighbors search, and fixed-radius near neighbor search in logarithmic time. We use the $k-d$ tree to efficiently find, for each rider j, the meeting points within a radius m_j from the rider's origin o_j and the rider's destination d_j.

2.4.2 Determining Time and Cost Feasible Matches

Our approach for determining time and cost feasible matches critically depends on the following observation.

Observation 1. *A match between a driver i and a set of riders $J \subseteq R$ with $|J| \geq 2$ is time feasible if the match between driver i and subset of riders $J' \subseteq J$ is time feasible for all $J' \subseteq J$.*

Hence, for a match of one driver and two riders to be time feasible, the match of the driver with each of these two riders must be time feasible as well. Similarly, for a match of one driver and three riders to be time feasible, the match of the driver with each of the possible pairs of riders must be time feasible as well. And so forth.

It is not necessarily the case that in a distance feasible match between one driver and two riders, the matches between the driver and the individual riders are distance feasible as well (recall Figure 2.2). In fact, one of the benefits of meeting points is that this does not have to be the case.

A Basic Algorithm

The basic algorithm considers drivers one by one and finds all time and cost feasible matches for that driver. A straightforward enumeration algorithm with run time complexity $O(nmk)$ finds all feasible single-rider matches, where n is the number of drivers, m is the number of riders, and k is the average number of feasible meeting point arcs per rider.

It follows from Observation 1 that only riders that could feasibly be matched with a driver have to be considered when constructing matches with two riders for that driver. Furthermore, we only need to start from pairs of riders that could feasibly be matched with a driver when constructing matches with three riders for that driver, and so forth. This realization is important because, typically, a driver can be feasibly matched with only a small fraction of the riders. Furthermore, not all pairs of feasible single-rider matches result in feasible two-rider matches, etc. This greatly reduces the number of combinations of riders that have to be considered when determining all feasible matches for a driver.

We thus construct feasible matches for a driver i recursively. We first find all feasible matches involving only one rider, then find all matches involving two riders, etc., up to the available capacity C_i. If a time and distance feasible match is found, an edge e is added to the ride-share matching problem with associated coefficients σ_e and ν_e.

A Refined Algorithm

By exploiting the structure of the problem and the characteristics of an instance, it is typically possible to significantly reduce the number of driver, rider pairs that are (fully) evaluated. Rider time windows are stored in a memory structure, which allows us to find the riders with time windows that overlap with the time window of a driver in sub-linear time. We have further enhanced this refinement by developing a method that reduces the size of the time windows stored in the memory structure by considering the minimum required overlap in the time window of a rider and a driver (related to the time a driver and a rider will spend travelling together in a feasible match). The details are given in ***Appendix 4***.

Next, we use the locations of the origin and destination and the time window of a driver and a rider to recognize that there cannot be a feasible match without considering meeting point arcs explicitly. The idea is similar to the logic employed to determine the feasibility of a match (i, J, a), but rather than using a meeting point arc, we calculate an implied time window considering origin and destination information only - not the actual pickup and drop-off points. The implied time window for the driver is calculated assuming that the rider is picked up and dropped off on the boundary of his walking range, i.e., on the two circles around o_j and d_j in Figure 2.5.

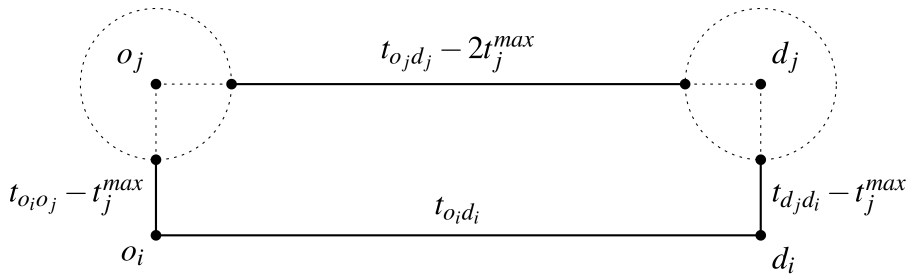


Figure 2.5: Detecting infeasibility of a match between driver i and rider j without considering meeting point arcs

Furthermore, we assume that the rider travels to the boundary of his walking range at driving speed. If there is no feasible match under that assumption, then there is no feasible match when the rider is walking.

Let t_j^{max} denote the time needed to drive distance d_j^{max}, which is the longest distance a rider is willing to walk to and from a meeting point. Driver i cannot pick up rider j before $e_i' = e_i + (t_{o_io_j} - t_j^{max})$ and he cannot pick up rider j after $l_i' = l_i - t_{o_jd_j} - t_{d_id_j} + 3t_j^{max}$ if he wishes to arrive to d_i in time. We can assume rider j cannot be picked up before $e_j' = e_j + t_j^{max}$ and after $l_j' = l_j - t_{o_jd_j} + t_j^{max}$. If $\max(e_j', e_i') > \min(l_j', l_i')$, then there cannot be a feasible match between driver i and rider j. From Figure 2.5, it is also clear that there cannot be a feasible match between driver i and rider j if $(t_{o_io_j} - t_j^{max}) + (t_{o_jd_j} - 2t_j^{max}) + (t_{d_jd_i} - t_j^{max}) > T_i$.

Only when the two checks above indicate that there may be a feasible match between a driver i and a rider j, we examine the matches of driver i and rider j for each meeting point arc (k, l) where $k \in M_j^p$ and $l \in M_j^d$. If a time and distance feasible match is found, an edge e is added to the ride-share matching problem with associated coefficients σ_e and ν_e.

The last refinement is based on the following observation.

Observation 2. *A match between a driver and a set of riders can only be feasible if the driver and the riders have at least one meeting point arc in common.*

In the basic algorithm, we store, for each feasible match of k riders, all time feasible meeting point arcs, i.e., not only the time feasible meeting point arc that resulted in the maximum driving distance savings. These meeting points arcs are used to construct matches with $k+1$ riders. Observation 2 shows that only meeting point arcs that are time feasible for at least $k+1$ riders are relevant. Hence, to construct matches with $k+1$ riders, we iterate over the meeting point arcs with feasible matches involving k riders, rather than over the feasible matches with k riders, and construct all feasible matches with $k+1$ riders on a meeting point arc, using the riders that are part of k-rider matches on that particular meeting point arc.

We provide the pseudocode for this solution approach in Algorithm 2.

Algorithm 2: Refined Feasible Match Generation

```
build k − d tree with meeting points ;
build interval container with rider time windows ;
for each rider do
    query k − d tree and store feasible meeting point arcs ;
end
for each driver do
    query interval container to obtain time compatible riders ;
    for each time compatible rider do
        if driver and rider spatially compatible then
            for each rider meeting point arc do
                if match driver, rider, meeting point arc is time feasible then
                    store meeting point arc ;
                    compute driving distance savings ;
                    if driving distance savings > best match driving distance savings
                      then
                        update best match driving distance savings ;
                        update best match ;
                    end
                end
            end
        end
        if best match driving distance savings > 0 then
            append best match to match list ;
            append rider to feasible rider list ;
        end
    end
    if number of feasible riders > 1 then
        for k = 2,...,C_i − 1 do
            Retrieve meeting point arcs ;
            Remove meeting point arcs that are feasible for less than k riders ;
            ... ;
            if driving distance savings > 0 then
                append match to match list ;
            end
        end
    end
end
return match list ;
```

2.5 A Computational Study

In this section, we report the results of an extensive computational study conducted to assess the benefits of the introduction of meeting points in different ride-sharing environments.

2.5.1 Generation of Ride-share Data Sets

Similar to Agatz (2011), we use the travel demand model for the metropolitan Atlanta region, developed by the Atlanta Regional Commission, as the basis for generating daily vehicle trips between different travel analysis zones (TAZs) within the region (the area covered by a TAZ is 4.1 square miles on average). For a subset of TAZs within the city of Atlanta, 229 to be precise, we generate five random streams of trips as follows. Each TAZ is a possible origin and a possible destination for a trip. For each origin-destination pair, we calculate an expected number of daily trip announcements by multiplying the average number of single-occupancy home-based work vehicle trips with a fixed percentage of vehicle-trips that we assume might consider participating in a ride-sharing system. Then for each pair, we determine the number of actual trip announcements using a Poisson random variable with expected value equal to the computed expected number of trips based on a participation rate of around 5.5%. For each trip announcement, we generate the origin and destination points within a fixed radius of 1.1 mile around the center of the travel analysis zone based on a uniform distribution. Each trip announcement is equally likely to be a rider announcement or a driver announcement. The minimum distance of a ride-share trip is 4 miles, i.e. we only consider trips between origins and destinations that are at least 4 miles apart. For each TAZ, we also randomly generate 4 meeting points around its center within a fixed radius of 1.1 mile.

Trip timing information is not available from the travel demand model. Therefore, we create the time windows for each announcement as follows. For each trip, we draw the earliest departure time from a normal distribution with mean 7:30 a.m. and standard deviation of half an hour to model a typical travel peak and calculate the earliest arrival time by adding the direct travel time to the earliest departure time. Subsequently, we calculate the latest departure (arrival) time by adding fixed time flexibility to the earliest departure (arrival) time. We assume the fixed time flexibility to be 30 minutes for all participants. The difference between the latest arrival time and earliest departure time is hence equal to the sum of the direct travel time from origin to destination and the fixed time flexibility. (Note that this means that we are investigating a morning commute.)

The travel distances between all points are computed using the haversine formula (which computes the great circle distance between two points) with a 30% uplift. To compute travel times, we assume a driving speed of 15 miles per hour (we are considering an urban area). For each driver $i \in D$, we define a limit on the total duration of his trip $T_i = t_{o_i,d_i} + min(4 + c_{flex} \cdot t_{o_i,d_i}, 20)$. Coefficient c_{flex} is the driver flexibility parameter - our assumption about how the willingness of the drivers to make detours depends on their original trip duration (fixed at 0.25 in base case - see below). The maximum trip duration for driver i is thus defined as his original trip duration plus an additional time that positively depends on his original trip duration. We assume that each driver that wishes to participate in ridesharing is always ready to extend his trip by at least 4 minutes, which is the time associated with one pick-up and one drop-off operation. We also assume that drivers are not willing to extend their original trips by more than 20 minutes, irrespective of their original trip length.

We assume a walking speed of 4 feet per second (LaPlante and Kaeser, 2004). The maximum walking distance for the rider to or from a meeting point is 0.5 miles, which corresponds to 11 minutes of walking at this speed. In addition, we impose the constraint that the total walk time cannot exceed the total time in a ride-share trip for a rider. In other words, the time that is spent walking in a trip must not exceed the time that is spent in the vehicle. This constraint can be manipulated by adjusting the rider flexibility parameter (assumed 1.0) which is the maximum ratio of the travel time to and from a meeting point to the time spent in the shared ride. (This additional restriction is enforced when searching for feasible meeting point arcs in the $k-d$ tree for each rider $j \in R$.)

A rider may be picked up at a meeting point or at his origin and dropped off at a meeting point or at his destination. A match involving two or more riders always starts and ends at a meeting point. Irrespective of the pickup or drop-off location, we always assume a service time per stop of 2 minutes. Each driver has a capacity of 3 spare seats. We limit ourselves to matches with no more than three riders, since this is the number of free seats in a typical sedan if the driver is driving alone. Also, back benches in most personal vehicles typically cannot accommodate three adults without compromising comfort.

The characteristics of the base case instances are summarized in Table 2.1.

Table 2.1: Characteristics of the base case instances

Trip pattern:	suburb to center
Avg. number of participants:	2849.4
Avg. number of drivers:	1425.8
Avg. number of riders:	1423.6
Avg. trip distance for driver:	7.58 mi
Avg. trip distance for rider:	7.64 mi
Avg. trip duration for driver:	30.34 min
Avg. trip duration for rider:	30.56 min
Max. distance to a meeting point:	0.5 mi
Travel (walk) speed to/from meeting point:	4 ft/s
Max. walk time to meeting point:	11 min
Driving speed:	15 mi/h
Rider flexibility parameter:	1.0
Driver flexibility parameter:	0.25
Maximum flexibility of driver:	20 min
Vehicle capacity:	3 seats

2.5.2 Performance

Both the algorithm for generating feasible matches and the simulation framework are implemented in Python 2.7. CPLEX 12.6 is used for solving matching problems.

The base case instances solve in less than 150 seconds on a quad-core i5-3360M machine with 4GB of RAM. CPLEX solves the two integer programs (recall that we employ hierarchical optimization) in a few seconds in all settings; virtually all the time for these instances is spent generating feasible matches. Instances with increased rider flexibility (Section 2.5.5) and increased participant density (Section 2.5.6) take more time, up to 10 minutes in a few cases, because the number of feasible matches increases.

These run times suggest that the algorithm is appropriate for use in practice. The instances used in our computational study represent trip announcements accumulated over several hours. In practice, in a dynamic setting, instances with a much smaller set of driver and rider announcements have to be solved at any one time. Furthermore, instead of having to generate a set of matches from scratch for each optimization, the existing set of matches has to be updated given any new information that has become available (e.g., matches involving certain riders and drivers have to be deleted and new matches involving riders and drivers that have just announced their trips have to be generated). This will be a matter of seconds rather than minutes. Performance wise, we expect to see similar results in a dynamic setting (albeit somewhat worse). For an environment without meeting points, Wang et al. (2014) have shown that the gap between a dynamic rolling horizon solution and a static benchmark is quite small. The gap will likely increase somewhat in an environment with meeting points, because some of the matches have to be committed to earlier (i.e., at the time that the rider has to start walking towards the meeting point).

2.5.3 Experiments

The main aim of this research is to analyze and quantify the benefits that meeting points can bring to a ride-sharing system. The solution approach that has been implemented provides a good basis for this, because it not only provides an optimal set of matches (for different objectives), but also furnishes the set of all feasible matches. We use the instance data and the set of feasible matches to compute and evaluate a number of metrics that provide insight into the quality of the optimal matching. In all the experiments, we either use the base case setting or a setting in which one of the characteristics is changed in order to assess the sensitivity of an optimal matching to this characteristic.

We evaluate and compare solutions using the following metrics: (1) the *matching rate for participants*, i.e., the fraction of participants that are matched, (2) the *matching rate for drivers*, i.e., the fraction of drivers that are matched, (3) the *matching rate for riders*, i.e., the fraction of riders that are matched, (4) the *mileage savings*, i.e., the relative mileage savings – system-wide vehicle-miles savings as a fraction of system-wide vehicle-miles when all participants drive alone, (5) the *driver trip time increase*, i.e., the average relative increase in the trip time of a driver – driver trip time increase as a fraction of original trip time, (6) the *rider trip time increase*, i.e., the average relative increase in the trip time of a rider – rider trip time increase as a fraction of original trip time, and (7) the *walking time*, i.e., the average walking time for a matched rider with a match that involves at least one meeting point.

2.5.4 Benefits of Meeting Points

As mentioned above, this research focuses on analyzing and quantifying the benefits of meeting points in a ride-sharing system. In Table 2.2, we compare the solution for the base case setting without meeting points to the solutions for the base case settings with 1,2 and 4 meetings points per TAZ, averaged over 5 randomly generated instances. The different meeting point densities represent the variation in the number of appropriate meeting points in different practical settings. To gain further insight, we also report statistics for two additional settings: in the first setting (labeled 4*), there are 4 meeting points per TAZ, but only single rider – single driver matches are allowed, and in the second setting (labeled 4**), there are 4 meeting points per TAZ, but only rider – driver matches using the closest meeting point to a rider's origin and destination are allowed. This reflects a setting in which the riders specify a particular meeting point upfront.

Table 2.2: Results for different numbers of meeting points and types of matches

	0	**1**	**2**	**4**	**4***	**4****
System:						
Matching rate (%)	68.00	71.14	72.90	**74.83**	74.13	69.71
Mileage savings (%)	27.39	28.36	28.93	**29.63**	29.24	27.59
Drivers:						
Matching rate (%)	67.96	70.93	72.45	**74.08**	74.08	69.65
Trip time increase (%)	25.45	25.98	26.31	**26.41**	26.19	25.77
Riders:						
Matching rate (%)	68.11	71.43	73.43	**75.65**	74.26	69.84
Trip time increase (%)	13.09	19.27	22.74	**26.54**	16.43	16.42
Walk time (min:sec)	-	8:06	8:28	**8:56**	8:45	5:06

We see that the introduction of meeting points results in a substantial increase in the number of participants matched (our primary objective) as well as in the mileage savings (our secondary objective). The matching rate increases by 6.8% when there are 4 meeting points per TAZ. The matching rate increase is slightly larger for riders than for drivers, because of matches involving more than one rider. The average trip time for matched drivers increases less than one percent (from 25.45% to 26.41%), but, as expected, the average trip time for matched riders increases noticeably, by slightly more than 12 percent (from 13.09% to 26.54%). This increase is due to the walking that is required for certain riders to or/and from a meeting point; on average the total walking time is between 8 and 9 minutes, which corresponds to a distance of about 0.4 miles. Riders with a match involving a pickup meeting point need to plan and execute their trips more carefully so as to ensure that they arrive at the meeting point in time. This may be considered an inconvenience, but, on the other hand, the service level (in terms of the chance of being matched) improves significantly. The results also suggest that most of the benefits can be achieved with single rider – single driver matches (4*) and that it is essential to consider all meeting points within range of a rider's origin or destination (4**).

The meeting points in the instances used in these experiments are drawn uniform randomly from a circle with radius 1.1 mile around the center of a TAZ. To assess the impact of the choice of meeting points in the case when there are 4 meeting points per TAZ, we performed the same experiments, but now with each of the 4 meeting points drawn uniform randomly from one of the quadrants of the circle, i.e., ensuring that the 4 meeting points are geographically spread out. In these experiments, there was a very slight increase in the observed matching rates. Because the differences were so small, we only use the original (more conservative) instances in the remaining experiments.

Figure 2.6 shows the number of single, double, and triple rider matches in the optimal solution for different numbers of meeting points.

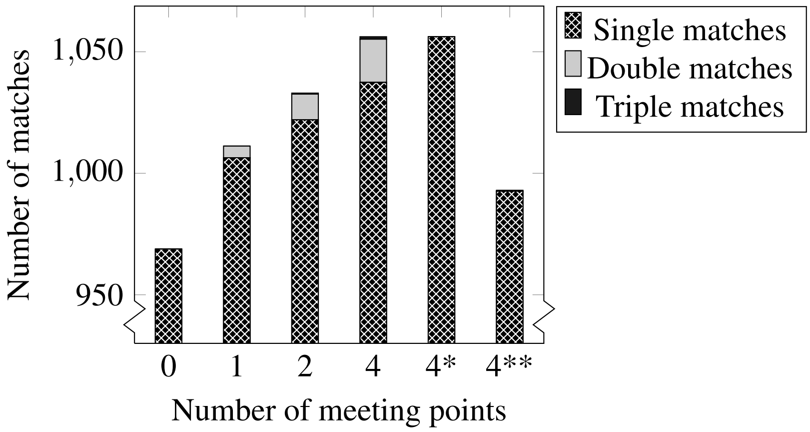


Figure 2.6: Number of single, double, and triple matches for different numbers of meeting points

We see that the number of participants in matches with two or three riders is quite small, 2.5% and 0.2%, respectively, of the total number of matched participants when there are 4 meeting points per TAZ. This suggests that the primary benefit of the introduction of meeting points is an increase in the number of single rider – single driver matching opportunities (rather than being able to create multiple rider – single driver matches). However, to some extent, this result may be a consequence of our choice of objective hierarchy: maximizing the number of matched participants followed by maximizing the mileage savings. When the number of drivers and riders in the system is roughly the same (as in our base case instances), it is more desirable to have single rider – single driver matches. That is, if it is possible to match two riders with the same driver, but it is also possible to match the two riders with different drivers, then the latter option is preferred as it results in four matched participants while the former results in only three matched participants.

Next, we examine the use of meeting points in more detail. In Figure 2.7, we show how many of the matches in the optimal solution use two meeting points, only a pick-up meeting point, only a drop-off meeting point, or no meeting points at all.

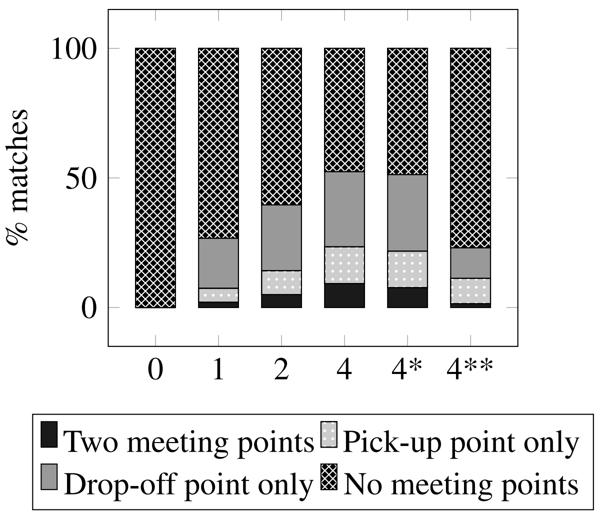


Figure 2.7: Use of meeting points in matches for different numbers of meeting points

As expected, the fraction of matches involving meeting points increases as the number of meeting points per TAZ increases. The fact that the fraction of matches that use only a drop-off point is much larger than the fraction of matches that use only a pickup point is a consequence of the fact that the instances represent trips during a morning commute with destinations mostly in the center of Metro Atlanta, which has a higher concentration of TAZs (each covering a smaller geographic area) and consequently a higher concentration of meeting points.

Table 2.3 provides further information regarding the matches in an optimal solution.

Table 2.3: Charateristics of the matchings in the optimal solution in terms of their use of meeting points for different numbers of meeting points

	0	**1**	**2**	**4**	**4***	**4****
No meeting points used (%)	100.00	73.31	60.38	**47.57**	48.72	76.98
Higher mileage savings (%)	-	23.26	35.13	**47.77**	47.45	20.17
Feasible because of meeting points (%)	-	12.75	19.40	**25.30**	24.75	7.47
- Detour became feasible (%)	-	11.89	18.14	**23.41**	22.75	6.77
- Time windows became feasible (%)	-	1.58	2.17	**3.20**	3.23	0.84
- Mileage savings became positive (%)	-	-	-	-	-	-

Specifically, we report the fraction of matches in the optimal solution that did not involve a meeting point, the fraction of matches in the optimal solution for which the mileage savings are higher because of the use of meeting points, and the fraction of matches in the optimal solution that would have been infeasible if it were not for the use of meeting points. For the latter set, we also identify the reason(s) that the use of meeting points resulted in a feasible match, i.e., the driver detour would have been infeasible without the use of meeting points, the participants' time windows would have been incompatible without the use of meeting points, the distance savings would have been negative without the use of meeting points. Note that a match can be counted in several categories, e.g., a match that is feasible because of the use of meeting points, could have been detour infeasible and time window infeasible. Note that all matches involving multiple riders are (by definition) feasible because of the use of meeting points and, for simplicity, all such matches are considered to have resulted in higher mileage savings.

We observe that when there are 4 meeting points per TAZ, the fraction of matches in the optimal solution that do not use meeting points is a little less than 50% and the fraction of matchings that would have been infeasible without meeting points is a little more than 25%. Furthermore, the use of meeting points makes matches feasible predominantly because it allows a smaller detour for the driver (only in a few cases, it makes rider and driver time windows compatible).

The fact that the fraction of matches in the optimal solution that do not use meeting points is close to 50% suggests that a more careful selection of meeting point locations may result in larger mileage savings. (Recall that meeting points have been selected randomly within a TAZ in these instances.)

Finally, in Table 2.4, we take a look at the number of additional feasible matching options generated by the introduction of meeting points. We show the number of riders (or pairs of riders or triples of riders) with at least one feasible match and the total number of feasible matches.

Table 2.4: Analysis of the number of feasible matches for different numbers of meeting points

	0	**1**	**2**	**4**	**4***	**4****
Single riders with feasible match	1290.6	1302.6	1308.4	**1316.0**	1316.0	1302.0
Number of single rider matches	20253.0	22940.6	24297.0	**25836.0**	25836.0	21565.0
Pairs of riders with feasible match	-	9.8	28.6	**53.4**	-	0.6
Number of rider pair matches	-	145.4	481.2	**994.6**	-	9.0
Triples of riders with feasible match	-	-	0.8	**2.4**	-	-
Number of triple rider matches	-	-	10.8	**28.6**	-	-

Without meeting points, approximately 90.6% of the riders have at least one feasible match. With meeting points, this fraction increases to approximately 92.5%. We see too that as the number of meeting points increases, the number of feasible matches grows steadily. There are about 27.5% more feasible matches for riders when there are 4 meeting points per TAZ. Not surprisingly, the increases are even more pronounced for matches involving pairs and triples of riders. This demonstrates that to increase the number of multi-rider matches, it will be critical to have a large number of carefully located meeting points.

In our results tables, we report averages across the five instances. While there was some variability among the five instance, the effects we were hoping to quantify proved stable across the five instances. Tables 2.5 and 2.6 provide detailed information with confidence intervals for the main results reported in this section. We see that the differences between the scenarios were indeed very stable across the five replications.

Table 2.5: Matching rates for the 5 base case instances

	Meeting point density				*Change*
	0	1	2	4	*0 – 4*
Instance 1	67.47	71.36	73.24	75.47	+ 8.00
Instance 2	68.22	71.51	73.49	75.16	+ 6.94
Instance 3	65.99	68.71	70.00	72.20	+ 6.21
Instance 4	67.00	70.24	71.88	74.13	+ 7.13
Instance 5	71.30	73.88	75.91	77.20	+ 5.90
Average	68.00	71.14	72.90	74.83	+ 6.84
CI (95%)	1.77	1.66	1.91	1.61	0.72

Table 2.6: Vehicle miles savings for the 5 base case instances

	Meeting point density				*Change*
	0	1	2	4	*0 – 4*
Instance 1	27.45	28.64	29.19	29.79	+ 2.34
Instance 2	27.70	28.65	29.20	29.93	+ 2.23
Instance 3	26.41	27.32	27.84	28.53	+ 2.12
Instance 4	27.30	28.32	28.97	29.75	+ 2.45
Instance 5	28.10	28.89	29.45	30.15	+ 2.05
Average	27.39	28.36	28.93	29.63	+ 2.24
CI (95%)	0.55	0.54	0.55	0.56	0.14

2.5.5 Impact of Time Flexibility

In this section, we study the impact of the time flexibility of the participants on the performance of the system and the benefits of meeting points. We vary the time flexibility of the drivers, the time flexibility of the riders, and the flexibility in participants' departure times.

In the base case, we consider a driver time flexibility of 25% of the original trip time ($c_{flex} = 0.25$). This time flexibility c_{flex} refers to the maximum extra trip time the drivers are willing to accept to serve one or more riders. Furthermore, all participants are assumed to have 30 minutes of flexibility in their trip departure time. To assess the impact of the time flexibility on the performance of a ride-sharing system, we evaluate the system performance when the time flexibility is lower, i.e., $c_{flex} = 0.15$, and when the time flexibility is higher, i.e., $c_{flex} = 0.35$. We note that extra trip time for drivers (and extra trip time for riders) always includes the service time incurred at a pick-up and a drop-off location. The results of these experiments are found in Table 2.7.

Table 2.7: Effects of driver time flexibility

	$c_{flex} = 0.15$		$c_{flex} = 0.25$		$c_{flex} = 0.35$	
	0	**4**	**0**	**4**	**0**	**4**
System:						
Matching rate (%)	56.68	64.96	68.00	74.83	75.41	**82.11**
Mileage savings (%)	23.70	26.65	27.39	29.63	29.23	**30.89**
Drivers:						
Matching rate (%)	56.66	64.27	67.96	74.08	75.36	**81.19**
Trip time increase (%)	19.35	20.66	25.45	26.41	30.65	**32.31**
Riders:						
Matching rate (%)	56.77	65.71	68.11	75.65	75.54	**83.11**
Trip time increase (%)	13.09	25.01	13.09	26.54	13.09	**27.91**
Walk time (min:sec)	-	8:52	-	8:56	-	**9:08**

We see that the willingness of drivers to accept a larger extra trip time has a substantial effect on the matching rate and the mileage savings. We also see that the negative impact of a decrease in time flexibility is larger than the positive impact of an increase, which suggests that there will be diminishing returns from increasing time flexibility. We observe too that the benefit of meeting points is negatively correlated with the time flexibility of the drivers. That is, the difference in participant matching rates is highest for the most constrained case (8.28%) and smallest for the least constrained case (6.70%). This points to the fact that meeting points are most valuable when drivers are reluctant to add extra time to their trip (e.g., on their way to work in the morning).

In the next set of experiments, we vary the time flexibility of the riders. In particular, we vary the travel speed and the travel range of the riders, i.e., the time it takes a rider to reach a meeting point and the distance a rider is willing to travel to reach a meeting point. Such an increase may be possible if riders use other modes of transportation instead of walking to get to a meeting point, e.g. using a (folding) bike, public transport, riding with a member of their household, etc. We increase the speed from the speed of walking (4 ft/s) to the speed of a cyclist (12 ft/s), and we increase the allowable range for the meeting points from 0.5 (base case) to 0.75 miles. Note that we maintain the assumption that the total travel time to and from a meeting point cannot exceed the time spent in the shared ride. We report selected results for this experiment in Table 2.8.

Table 2.8: Effects of rider time flexibility

Travel speed to meeting point	-	Low		High	
Maximum distance to meeting point (mi)	-	0.5	0.75	0.5	0.75
System:					
Matching rate (%)	68.00	74.83	79.84	76.17	**83.84**
Mileage savings (%)	27.39	29.63	31.32	30.01	**32.14**
Drivers:					
Matching rate (%)	67.96	74.08	77.72	75.14	**81.5**
Trip time increase (%)	25.45	26.41	28.02	26.85	**27.33**
Riders:					
Matching rate (%)	68.11	75.65	82.02	77.26	**86.22**
Trip time increase (%)	13.09	26.54	38.78	16.88	**21.27**
Trip time to/from m. point (min:sec)	-	8:56	13:13	3:11	**5:56**
Trip distance to/from m. point (mi)	-	0.40	0.59	0.43	**0.70**

We see that the willingness to consider more distant meeting points combined with the ability to get to a meeting point faster than by walking can greatly increase system performance. The matching rates are much higher and also the number of feasible double and triple matches increases significantly. It is important to observe that only increasing the walking range results in improved system performance.

If we examine the structure of the optimal matchings in the most flexible scenario in more detail, we find that 74.47% of the matches use meeting points, compared to 52.4% in the base case (see Figure 2.7). Also, we find that 45.11% of these matches would be detour-infeasible without the meeting points, compared to 23.41% in the base case (see Table 2.3).

These findings stress the importance of encouraging riders to consider more distant meeting points and of encouraging drivers to accept longer detours. A ride-sharing service may investigate the benefits of incentive payments to riders and drivers that are willing to be more flexible as a way to increase the matching rate and the mileage savings.

Finally, we perform an experiment in which we vary the flexibility in the departure time for all participants (30 minutes in base case). We evaluate a scenario with lower flexibility (20 minutes) and a scenario with higher flexibility (40 minutes). The results are given in Table 2.9.

Similar to the driver detour and rider walking time flexibility, we see that departure time flexibility has an important positive effect on the observed matching rates. However, as with the driver flexibility, we see diminishing returns from increasing the flexibility of the departure time. We also note that the benefits of meeting points are higher when participants are more flexible in their departure times (as seen by the difference between the matching rates with and without meeting points). The reason for this is that the meeting points create more matches that are detour-feasible while additional departure time flexibility creates more matches that are time-feasible.

Table 2.9: Effects of flexibility in departure time

	20*min*		30*min*		40*min*	
	0	**4**	**0**	**4**	**0**	**4**
System:						
Matching rate (%)	61.27	67.07	68.00	74.83	71.75	**79.21**
Mileage savings (%)	24.21	26.03	27.39	29.63	29.25	**31.6**
Drivers:						
Matching rate (%)	61.24	66.88	67.96	74.08	71.71	**78.05**
Trip time increase (%)	24.59	25.37	25.45	26.41	25.81	**27.18**
Riders:						
Matching rate (%)	61.37	67.33	68.11	75.65	71.86	**80.43**
Trip time increase (%)	13.09	23.92	13.09	26.54	13.09	**27.95**
Walk time (min:sec)	-	8:11	-	8:56	-	**9:14**

2.5.6 Effect of Trip Patterns and Density

In this section, we study the effect of the number of participants in the system and their trip patterns on the system performance. Table 2.10 gives an overview of the characteristics of the instances that we generated for this purpose.

Table 2.10: Characteristics of instances with different trip patterns and densities

	drivers : riders				
	1 : 1	2 : 2	1 : 2	2 : 1	$1 : 1^c$
Trip pattern	default	default	default	default	corridor
Avg. number of participants	2849.4	5578.6	4272.4	4272.4	2594.4
Avg. number of drivers	1425.8	2777.8	1425.8	2845.2	1295.4
Avg. number of riders	1423.6	2800.8	2846.6	1427.2	1299.0
Avg. trip distance for driver (mi)	7.58	7.60	7.58	7.59	9.36
Avg. trip distance for rider (mi)	7.64	7.62	7.64	7.68	9.35
Avg. trip duration for driver (min)	30.34	30.39	30.34	30.38	37.43
Avg. trip duration for rider (min)	30.56	30.47	30.56	30.70	37.38

First, we consider a setting with twice as many participants than in the base case (denoted by 2 : 2). We also consider a setting with twice as many riders but the same number of drivers as in the base case (denoted by 1 : 2). This represents an environment in which the pool of ride-share participants is skewed towards the riders, who have more to gain from participating. For completeness sake, we also consider the opposite case: a setting with twice as many drivers as there are riders (denoted by 2 : 1). To study the effect of a different trip patterns, we create a set of instances in which participants travel along a narrow South-North corridor in the Atlanta region. While in the base case (denoted by default) the area is shaped like a square with trips originating in suburban areas and heading towards the urban center, the corridor instances represent trips that occur in a narrow rectangle. To allow for a fair comparison, the geographic area covered in the five corridor instances is roughly the same as in the base case, and, similarly, the number of trips, TAZ locations, and meeting points is roughly the same as in the base case (this setting is denoted by $1 : 1^c$). Table 2.11 presents the results for the different experiments.

Table 2.11: Effects of trip patterns and density

	drivers : riders									
	1 : 1		2 : 2		1 : 2		2 : 1		1 : 1^c	
	0	**4**	**0**	**4**	**0**	**4**	**0**	**4**	**0**	**4**
System:										
Matching rate (%)	68.00	74.83	75.02	82.91	52.40	59.75	61.34	62.69	72.16	78.26
Mileage savings (%)	27.39	29.63	31.06	33.65	21.83	25.42	26.39	27.11	31.35	33.26
Drivers:										
Matching rate (%)	67.96	74.08	75.34	81.57	78.49	84.22	46.07	47.09	72.31	77.51
Trip time increase (%)	25.45	26.41	25.75	27.43	25.45	26.13	20.13	19.40	22.56	23.46
Riders:										
Matching rate (%)	68.11	75.65	74.73	84.26	39.35	47.52	91.87	93.88	72.23	79.21
Trip time increase (%)	13.09	26.54	13.13	28.16	13.09	29.22	13.09	19.12	10.70	21.54
Walk time (min:sec)	-	8:56	-	9:19	-	9:41	-	7:50	-	9:00

As expected, we see that the matching rate increases with the number of participants. More surprising is the fact that the relative advantage of the use of meeting points in terms of the overall matching rate also seems to increase slightly with the density. A potential explanation for this is that opportunities for matches with multiple riders increase.

With twice as many riders than drivers in the system, we see that 47.52% of the riders are matched, which is almost best possible (50%) if we ignore the possibility of double and triple matches. The number of double and triple matches has increased compared to the base case, but it is still relatively small. A more careful choice of meeting points may result in an increase of the number of double and triple matches, but it is more likely that an increase in both rider and driver time flexibility is needed.

Maybe as expected, in the setting with twice as many drivers as riders, the introduction of meeting points has only a small impact on the matching rate. The matching rate is already very high without meeting points. With the introduction of meeting points, the share of riders that are in at least one feasible match increases from 94.83% to 95.78%, while the matching rate for riders increases from 91.87% to 93.88%. Note, though, that this is the one setting in which the trip time increase for drivers decreases with the introduction of meeting points (from 20.13% to 19.40%).

For the corridor instances, we see that both the matching rate and the mileage savings are approximately 4% higher than for the default instances, and that the benefits of the meeting points are similar. The same holds for the trip time increase for drivers and the walking distance for riders.

2.5.7 The Impact of Objective Hierarchies

All the results discussed so far were obtained using the objective hierarchy in which the number of matches (z_1) is maximized first followed by maximizing the mileage savings (z_2). We observed in Section 2.5.4 that this objective hierarchy tends to favor solutions involving single matches. In this section, we compare the system performance for three natural objective hierarchies. The first (*Hierarachy 1*) is the default one, and maximizes participant matches followed by mileage savings, the second (*Hierarachy 2*) maximizes mileage savings followed by participant matches, and the third (*Hierarachy 3*) maximizes rider matches followed by mileage savings. The latter may be more desirable than the default hierarchy, in which the primary objective is maximizing participant matches, because unmatched riders may not necessarily have the option of using their own car to perform their trip. The results for the five base case instances and 4 meeting points per TAZ can be found in Table 2.12.

Table 2.12: Results for different objective hierarchies

	Hierarchy 1 (Matches – Savings)	*Hierarchy 2 (Savings – Matches)*	*Hierarchy 3 (R. Matches – Savings)*
System:			
Matching rate (%)	**74.83**	73.88	74.36
Mileage savings (%)	29.63	**29.79**	29.74
Drivers:			
Matching rate (%)	**74.08**	72.66	73.14
Trip time increase (%)	**26.41**	25.58	26.09
Riders:			
Matching rate (%)	**75.65**	75.17	**75.65**
Trip time increase (%)	26.54	26.01	26.41
Walk time (min:sec)	8:56	8:51	8:57
Matching:			
Num. of single matches	**1037.4**	1003.8	1011.0
Num. of double matches	17.8	**30.6**	30.4
Num. of triple matches	1.0	**1.4**	**1.4**

We see that the difference for all but one of the system performance metrics for the three objective hierarchies is less than one percentage point. The exception is the matching rate for the drivers, which for the default objective hierarchy (*Hierarchy 1*) is 1.42% larger than for the objective hierarchy in which the primary focus is on maximizing mileage savings (*Hierarchy 2*). This reflects the structural difference in the optimal matchings: there are almost twice as many double matches when the primary objective is to maximize mileage savings. Interestingly, the matching rate for riders does not increase when maximizing the number of matched riders is taken as the primary objective (rather than maximizing the number of match participants). Triple matches are still rare in all solutions. From a ride-sharing service provider's perspective, the default objective hierarchy is likely to be preferred, as their revenue is linked to the participant matching rate. However, from a societal perspective, the alternative objective hierarchy in which the number of riders matched is the primary objective is probably preferable, as it strikes a better balance between rider mobility and mileage savings (which are linked to congestion and emissions).

Since the number of matched participants is an integer, it is relatively easy to compute the Pareto frontier that characterizes the trade-off between the number of matched participants and the mileage savings. Figure 2.8 depicts the Pareto frontier for the third of the five base case instances (the frontier for the other base case instances look similar).

As already indicated by the small differences in the values of the performance metrics in Table 2.12, the two objectives are well aligned. This is also reflected in the small number of points that constitute the Pareto frontier.

Figure 2.9 provides more detail regarding the change in the structure of the optimal matchings as we move from a solution obtained with the default objective hierarchy to a solution

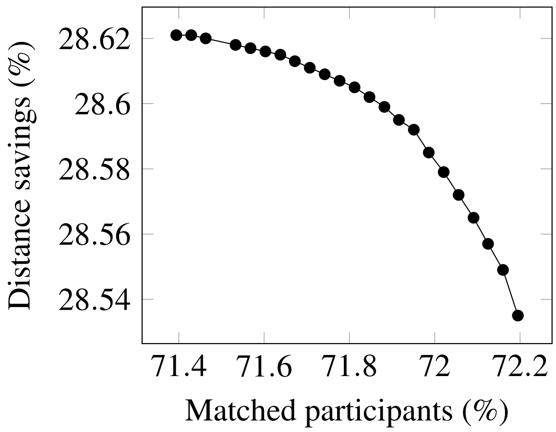


Figure 2.8: Pareto frontier for the third base case instance

obtained with the alternative objective hierarchy in which the primary focus is on mileage savings. We see that single rider – single driver matches are replaced by matches involving two or three riders. As a consequence, the number of drivers with matches decreases. Surprisingly, the number of riders with matches also decreases, which indicates that in many situations riders are "competing" for the same drivers.

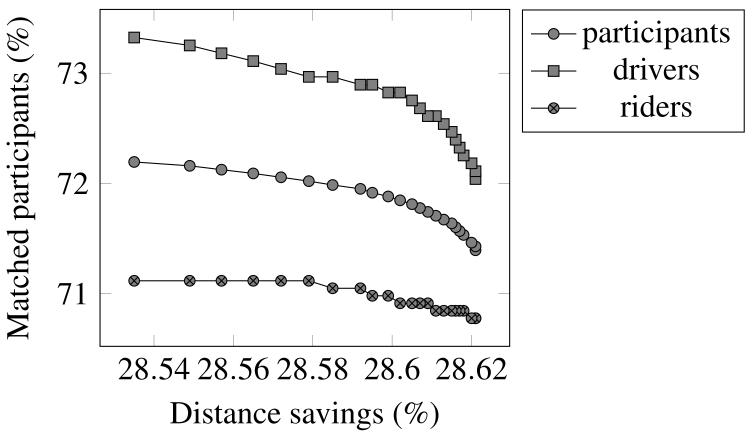


Figure 2.9: Matching rates for Pareto efficient points for the third base case instance

2.6 Conclusion

In this study, we have shown that the introduction of meeting points in a ride-sharing system can substantially improve a number of critical performance metrics, i.e., percentage of matched riders, percentage of matched participants, and mileage savings. The price that has to be paid to achieve these performance increases is minor: riders may have to walk a short distance and may have to plan their time more carefully so as to ensure that they arrive on time at the meeting point where they are to be picked up (it is unlikely that drivers will be willing to wait for a rider at a pickup point for more than a minute or two). Even though the number of possible matches increases significantly with the introduction of meeting points, our computational experiments have demonstrated that all feasible matches can be generated efficiently with a carefully designed and implemented algorithm.

The observed increases in performance of a ride-sharing system resulting from the introduction of meeting points may even be greater when the meeting points are chosen carefully based on observed travel patterns. This is an interesting opportunity for further research.

As expected, driver and, especially, rider flexibility strongly impact the performance of a ride-sharing system. This points to two additional and interrelated future research directions: (1) how to stimulate (and reward) riders to increase their flexibility and be willing to use more distant meeting points, and, similarly, how to stimulate (and reward) drivers to increase their flexibility and be willing to make longer detours, and (2) how to (better) integrate ride-sharing systems with other available transportation systems, e.g., bike-sharing systems, so as to ensure that riders can reach meeting points that are further away fast and easy.

3 ENHANCING URBAN MOBILITY: INTEGRATING RIDESHARING AND PUBLIC TRANSIT

3.1 Introduction

People all around the world use private cars to travel to work. Most of these commuter trips are single-occupant vehicle trips. In the U.S.A., for example, single-occupant trips represent approximately 77% of all commuter trips (Polzin and Pisarski, 2013); similar percentages are found in Europe (EEA, 2010). Low vehicle occupancy rates combined with the high number of trips made during peak hours often creates severe traffic congestion in urban areas. The resulting stress and the air pollution caused by vehicle emissions can have serious negative health effects.

To reduce the negative externalities of car travel, local governments encourage the use of public transport. Unfortunately, many suburban and rural areas are not adequately served as they lack the population density to justify public transit, i.e. the public transport is not economically viable. In cities with sprawling suburban areas, the utilization of public transit to commute to work is often low, e.g. less than 5% in metropolitan areas like Houston and Atlanta (McKenzie, 2010).

To attract more riders from suburban areas to public transit, transportation agencies must find adequate solutions to accommodate the first and last mile from the riders' home to and from the transit stations. Possible solutions for a transportation agency include operating a fleet of demand-responsive feeder vehicles and collaborating with local taxi service providers. In the U.S.A., for example, public transport providers have started collaborating with Uber and Lyft to better coordinate their service offer (Murphy, 2016). While the services provided by Uber and Lyft are convenient for the riders, they are often (too) costly for the transportation agency and/or the riders.

A cheaper and more environmentally sustainable alternative is to use already existing trips as a feeder for public transit. A recent TRB report highlights ridesharing as an important opportunity for transportation agencies when seeking to address the "last mile problem" (Murray et al., 2012). Deutsche Bahn (German Railways) is running several pilots to synchronize bus and rideshares to provide convenient door-to-door transport to their travelers (Annual report DB 2015). The Flinc ridesharing smartphone app will soon integrate train timetables and regional public transport schedules across Germany.

In this chapter, we examine the potential benefits of integrating ridesharing and public transit. Ridesharing and public transit can, in fact, complement each other. On one hand, ridesharing can serve as a feeder system that connects less densely populated areas to public transport. On the other hand, the public transit system can extend the reach of ridesharing and reduce the detours to be made by drivers. As such, it may help overcome incompatibilities in the itineraries of drivers and riders and facilitate the matching process.

Consider the example depicted in Figure 3.1 with driver d_1 and rider r_1 and two stations s_1 and s_2, where the circles indicate the origins and the squares indicate the destinations,

This chapter is based on Stiglic et al. (2016a).

where the number above an arc represents the time it takes to travel between the nodes, and where the driver is willing to accept an increased trip time of at most five minutes.

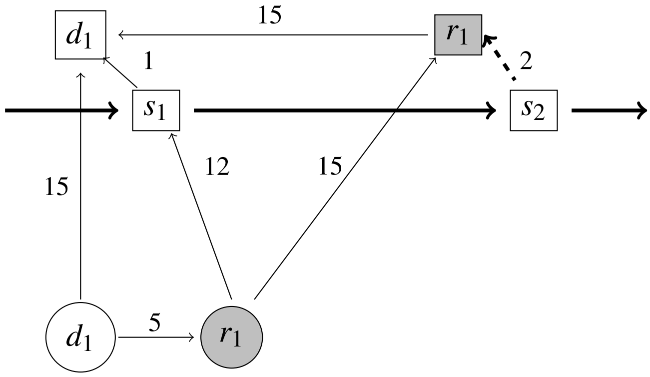


Figure 3.1: Rider (grey) and Driver (white) traveling from Origin (circle) to Destination (square) via Public Transit

Without the use of public transit, a match between d_1 and r_1 is not feasible because the required increase in trip time (20 min) exceeds the driver's limit. If, however, the rider is willing to take the transit line from s_1 to s_2 and walk 2 minutes to his final destination, a feasible match between d_1 and r_1 is possible because d_1 has to make a smaller detour. (We have to ensure the rider can reach his final destination in time based on the transit line's timetable.)

We consider a centralized system that automatically establishes matches between drivers and riders. A driver can either move a rider from his origin directly to his destination or to a transit station so that he can take the train to his final destination. The transit system operates a fixed timetable and drivers and riders announce their itineraries and time schedules on short notice. In this chapter, we assume the system creates matches in a way that maximizes the number of riders who are matched. As a secondary objective, the system minimizes the additional driving distance of the matched drivers. As such, this hierarchy aims to maximize rider mobility while minimizing the inconvenience of the matched drivers and, at the same time, the negative externalities of car travel.

Given that such a system potentially involves two or even more stakeholders (rideshare provider, transit agency, local government), determining operational objectives might not be a straightforward task. The ridesharing literature suggests several alternatives such as maximizing the number of matched participants, maximizing the number of matched riders, or maximizing the distance savings. These objectives are well aligned with the objectives of a rideshare provider. The increase in total trip time duration for participants is an important aspect too since participants may be very sensitive to how much (additional) time they spend in transit. In contrast, a transit agency will very likely be interested in increasing ridership on public transport lines and the local government could be interested in reducing the number of total vehicle miles.

When there is a 'park and ride' facility at the station, a driver may even decide to park his car after taking a rider to the station and take public transport to his final destination himself. Drivers provide the ride in exchange for a small fee that covers the vehicle's operating costs and a small remuneration for the time that has been "lost" (due to the pickup and drop-off as well as the detour). The public transit provider may choose to cooperate with other stakeholders and offer additional benefits to drivers who are willing to accommodate riders. These might come in the form of a free park-and-ride ticket, a toll waiver, an HOV lane permit, or priority parking in the city center. As such, this service can be offered at a relatively low cost to the rider. This may help to increase rider-ship on public transit (which will lead to an increase in public transportation revenues) and reduce the number of personal vehicles on the road.

The main contributions of our research are that: (i) we introduce a new and relevant problem that considers the integration and synchronization of ridesharing and scheduled public transit; (ii) we present a solution approach to optimally create single or multi-modal rideshare matches; and (iii) we conduct an extensive numerical study on artificial instances that capture the primary characteristics of many real-world transit settings and quantify the benefits of integrating ridesharing and public transport. The remainder of the chapter is organized as follows. In the next section, we discuss relevant and related literature. In Sections 3.3 and 3.4, we formally define the problem and introduce a solution approach. In Section 3.5, we discuss the details of our numerical experiments and present the core results. We finish the chapter with a summary of our findings and recommendations for practitioners.

3.2 Related literature

The use of demand-responsive transportation services to enhance the performance of public transport systems has been promoted for some time and a wide variety of such services is offered in practice, often focusing on, or reserved for, people with disabilities and subsidized by the government.

Cayford and Yim (2004) present a basic demand-responsive feeder system for the city of Milbrae in California. In low-demand periods, buses can deviate from their ‘fixed’ routes as well as from their scheduled departure times based on actual demand. The route deviation is limited to skipping portions of the fixed route when there is no demand for drop-offs on those portions of a particular trip. Buses operate according to the fixed schedule in high-demand periods.

One of the common integration options between a fixed-schedule system and an on-demand feeder system is the so-called Demand Responsive Connector (DRC). Koffman (2004) presents examples of DRCs in several US cities and observes that it is one of the most popular types of flexible transit services. Such systems typically operate within a service area and move passengers to and from a transfer point that connects to a major fixed-route transit network.

A critical question in this context is under which conditions is it better, in service and cost terms, to operate the feeder system using a fixed-route policy and under which conditions is it better to operate the feeder system as a demand-responsive service. Quadrifoglio and Xiugang (2009) and Li and Quadrifoglio (2010) develop analytical and simulation models for this purpose and find that the switching point between a demand-responsive and fixed-route policy is in the range of 10 to 50 customers/mi^2/h.

A DRC is typically deployed within a specified zone where the zone has one transfer point and a single transportation company provides the service. Li and Quadrifoglio (2011) consider the optimal zone design problem and develop an analytical model based on continuous approximations. Lee and Savelsbergh (2017) study a setting in which a zone has multiple transfer stations and a passenger can be dropped off at any of these transfer stations as long as the passenger's desired service is met. They demonstrate that a more flexible system can offer substantial cost savings, especially when transit services are frequent and/or transit stations are relatively close together.

In this chapter, we consider the use of ridesharing as a feeder system for scheduled transit. Several recent papers focus on algorithmic approaches to optimally match riders and drivers in the ridesharing context. The work in this area typically considers door-to-door trips in which a driver moves one or more riders from their origin to their destination (see, for example, Agatz et al. (2011), Wang et al. (2016), Lee and Savelsbergh (2015)). Furuhata et al. (2013) and Agatz et al. (2012) provide a detailed overview of this line of research and the links to other modes of transport.

In Chapter 2, we do not require door-to-door transportation but allow riders to walk to and from their meeting points to facilitate more convenient rides for the drivers. In this chapter, we extend this work by also allowing transfers to a transit service with a fixed schedule. While there are only a few meeting points within walking distance of a specific rider, many more transit stations are within reach by using scheduled transit. The synchronization of the rideshare trips with the transit service links our work to multi-hop ridesharing which involves the temporal synchronization of different rideshare trips (see Herbawi and Weber (2011a), Coltin and Veloso (2013) and Drews and Luxen (2013) for algorithmic approaches to these types of problems).

3.3 Problem definition

We consider a transit service provider that receives a sequence of trip announcements A from participants. The set of announcements A can be partitioned into a set of trip announcements by the drivers, $D \subset A$, and a set of trip announcements by the riders, $R \subset A$. A trip announcement $a \in A$ has a submission time, σ_a, at which it becomes known in the system, has an origin location, o_a, and a destination location, d_a, as well as an earliest departure time, e_a, at the origin location and a latest arrival time, l_a, at the destination location. We assume that the participants' departure times are somewhat flexible so that the difference $l_a - e_a$ is greater than the travel time from the origin to the destination. For each $a \in A$, there is also a maximum acceptable trip duration, T_a, a maximum acceptable walking distance from a transit station, M_a, and a maximum acceptable waiting time at a transit station, W_a. We denote the distance from location i to j by d_{ij}, the vehicle travel time from location i to j by t_{ij}, and the time it takes to walk from i to j by $\hat{t}_{ij}$. We assume that a service time of τ is incurred when a driver picks up a rider.

We consider a connected public transit network with a fixed cyclic timetable. The timetable describes the departure and arrival times of each train at each station. (We refer to the transit vehicles as trains but these could also be buses, trams, ferries, or metros.) Let S denote the set of transit stations. Let $\bar{t}_{ij}$ be the (shortest) travel time by train between station $i \in S$ and $j \in S$, where traveling between two stations in the network can involve one train or multiple trains with transfers. In case the shortest route involves transfers, then transfer and waiting times are included in the travel time. Note that for a cyclic timetable with identical trains, the travel time $\bar{t}_{ij}$ is not dependent on the departure time from station i. (Modeling a more general setting in which the timetable is not cyclic would require a travel time that depends on a specific departure k, i.e. $\bar{t}_{ij}^{k}$.) We denote by $\theta_s^-(t)$ the last train departure at station $s \in S$ at or before time t and by $\theta_s^+(t)$ the first train departure at station $s \in S$ at or after time t.

For presentational convenience, we only consider trips into the city center, i.e. the morning commute. It is straightforward to extend the concepts and technology to accommodate more extensive travel patterns. Consequently, we assume for each station $s \in S$ an access time, τ_s, representing the time needed to walk from the drop-off location to the platform. Further, we assume that a subset of the transit stations, $S_P \subseteq S$, have a park-and-ride facility and for each station $s \in S_P$ a park time, $\tilde{\tau}_s$, representing the time needed to park and walk from the parking facility to the platform. Finally, we assume that when a rider (or a driver) uses public transport he will always travel to the station that is closest to his final destination. Let s_r be the closest station to the destination of rider r. Hence, using public transit is only feasible for rider r if s_r is within the maximum acceptable walking distance of his destination, i.e. when $d_{s_r d_r} \leq M_r$. For assessing the feasibility of matches involving public transport, it is convenient to calculate the latest time, l_r^s, at which a rider r can arrive at the platform at station $s \in S$ and reach his destination at or before l_r, i.e. $l_r^s = \theta_s^-(l_r - \hat{t}_{s_r d_r} - \bar{t}_{s s_r})$.

We allow matches in which a driver picks up two riders for a drop-off at a transit station (i.e. from two different pickup locations). However, to minimize the inconvenience to the driver the two riders will be dropped off at the same transit station.

Thus, the transit service provider offers these types of matches:

- *A rideshare match:* a match between a rider and the driver in which the driver transports the rider from his origin to his destination;
- *A transit match:* a match between a rider and the driver in which the driver transports the rider to a transit station. Subsequently, the driver drives to his destination while the rider takes public transit to reach his final destination.
- *A park-and-ride match:* a transit match in which the driver parks his car and then uses public transport to reach his final destination.

A rideshare match always involves one rider, but both a transit match and a park-and-ride match can involve one or two riders. At this stage, we do not consider matches with more than two rider pickups. The main reason is that it would increase the inconvenience to the driver (it will increase the driver's journey time and increase the risk of delays on the driver's journey). Note that we do allow multiple riders to travel together between the same origin and destination. A secondary reason is that it facilitates our presentation. Finally, we observe that, conceptually, it is straightforward to extend our algorithm to handle three or more rider pickups.

Table 3.1: A summary of the notation used

Sets	
D	Set of driver trip announcements
R	Set of rider trip announcements
A	Set of trip announcements, $A = D \cup R$
S	Set of transit stations
S_P	Set of transit stations with a park-and-ride facility, $S_P \subseteq S$
Locations	
o_a, d_a	Origin and destination location of trip announcement $a \in A$
s_r	The closest station to the destination of rider $r \in R$, $s_r \in S$
d_{ij}	Distance from location i to j
Time parameters	
σ_a	Announcement time of trip announcement $a \in A$
e_a	Earliest departure time from the origin of trip announcement $a \in A$
l_a	Latest arrival time at the destination of trip announcement $a \in A$
l_r^s	Latest time at which a rider r can arrive at the platform at station $s \in S$ and reach his destination at or before l_r
T_a	Maximum acceptable trip duration for trip announcement $a \in A$
M_a	Maximum acceptable walking distance from a transit station for trip announcement $a \in A$
W_a	Maximum acceptable waiting time at a transit station for trip announcement $a \in A$
t_{ij}	Travel time from location i to j by car
$\hat{t}_{ij}$	Travel time from location i to j by foot
$\bar{t}_{ij}$	Shortest travel time between station $i \in S$ and $j \in S$ by train
τ	Service time at the pickup location of a rider
τ_s	Service time for the rider to walk from to drop-off location to the platform at station $s \in S$
$\tilde{\tau}_s$	Service time for the rider to park and walk from the parking facility to the platform at station $s \in S_P$
$\theta_s^+(t), \theta_s^-(t)$	First and last train departure at station $s \in S$ at or before time t

3.3.1 Assessing the feasibility of a rideshare match

A rideshare match involves a driver $i \in D$ and a rider $j \in R$. The departure time e_i^j of driver i matched with rider j depends on the announcement submission time of rider j, the earliest departure time of rider j, and the trip duration from o_i to o_j, and is set as follows $e_i^j = \max(e_i, \sigma_j, e_j - t_{o_io_j})$. This assures that driver i does not wait for rider j at o_j. The pickup time, e_j^i, of rider j is $e_i^j + t_{o_io_j}$ and the arrival time, l_j^i, at the rider's destination is $e_j^i + \tau + t_{o_jd_j}$. Driver i will arrive at his destination at time $l_i^j = l_j^i + t_{d_jd_i}$. The rideshare match is only feasible if the trip time duration for driver i, i.e. $l_i^j - e_i^j$, is less than or equal to T_i. Further, driver i and rider j must arrive at their destinations before their latest arrival times l_i and l_j, respectively.

3.3.2 Assessing the feasibility of a transit match

A single-rider transit match involves a driver $i \in D$, a rider $j \in R$, and a transit station $s \in S$. The driver's departure time, e_i^j, and the rider's pickup time, e_j^i, are computed in the same way as for a rideshare match. The arrival time at station s is $e_j^i + \tau + t_{o_js}$. The driver arrives at his destination at time $l_i^j = e_j^i + t_{o_js} + \tau + t_{sd_i}$. The rider arrives at his destination at time $l_j^i = \theta^+(e_j^i + \tau + t_{o_js} + \tau_s) + \bar{t}_{ss_r} + \hat{t}_{s_rd_r}$. The transit match is feasible if the rider arrives at the platform of the transit station in time, i.e. $e_j^i + \tau + t_{o_js} + \tau_s \leq l_j^s$, if the driver arrives at his destination in time, i.e. $l_i^j \leq l_i$, and the trip durations of the rider, i.e. $l_j^i - e_j^i$, and the driver, i.e. $l_i^j - e_i^j$, are less than or equal to T_j and T_i, respectively.

A two-rider transit match involves a driver $i \in D$, riders $j, k \in R$ $(j \neq k)$, and a transit station $s \in S$. Without loss of generality, we assume that rider j is picked up before rider k. To avoid waiting at o_j and o_k, the driver departs at $e_i^{jk} = \max(e_i, a_j, e_j - t_{o_io_j}, a_k, e_k - (t_{o_io_j} + \tau + t_{o_jo_k}))$. The pickup time e_j^i of rider j is $e_i^{jk} + t_{o_io_j}$ and the pickup time e_i^k of rider k is $e_j^i + \tau + t_{o_jo_k}$. The arrival time at station s is $e_k^i + \tau + t_{o_ks}$. The driver arrives at his destination at time $l_i^j = e_k^i + \tau + t_{o_js} + t_{sd_i}$. The transit match is only feasible if both riders arrive at the platform at the transit station in time, if the driver arrives at his destination in time, and if the trip durations of the driver and the riders are less than the maximum acceptable trip durations.

Note that in the above discussion we have not explicitly considered the waiting time of a rider at the platform, e.g. $w_j^i = \theta^+(e_j^i + \tau + t_{o_js} + \tau_s) - (e_j^i + \tau + t_{o_js} + \tau_s)$ in the case of a single-rider transit match. However, it is easy to see that if the transit match is feasible, the waiting time at the transit station can be minimized by having the driver depart later. The delay in the driver's departure is bounded by $\min(w_j^i, l_i - l_j^i, l_j^s - e_j^i - (\tau + t_{o_js} + \tau_s))$, i.e. the current waiting time, the remaining driver flexibility, and the remaining rider flexibility.

3.3.3 Assessing the feasibility of a park-and-ride match

Assessing the feasibility of a park-and-ride match is similar to assessing the feasibility of a transit match. We only have to account for the fact that, after dropping off the rider(s) at the transit station, the driver continues using public transport rather than by using his car. The first difference occurs in calculating the arrival time at the station platform: rather than using the access time τ_s, we use the park time $\tilde{\tau}_s$ to account for the fact that the driver parks his car and then the driver and rider(s) walk to the platform from there. The second difference is calculating the time for the driver to arrive at his destination, which now involves public transport.

3.3.4 The matching problem

As in the models introduced in Chapter 2 and Chapter 3, we create a node for each driver $i \in D$ and each rider $j \in R$ and an edge connecting node i and j if there is a feasible match between driver i and rider j. We also introduce nodes that represent pairs of riders (j,k), where $j,k \in R$ and $j \neq k$. We add an edge connecting node i and (j,k) if there is a feasible match between driver i and rider pair (j,k). Each edge e has two weights: the number of riders in the match, ν_e, and the additional driving distance for driver δ_e (note that this value may be negative in park-and-ride matches because the driver uses public transport to reach his destination from the park-and-ride station). Note that a transit match between a driver and a rider (or a driver and a pair of riders) may not be unique since a number of feasible transit matches involving different stations and different departures may exist. Similarly, a park-and-ride match between a driver and a rider (or a driver and a pair of riders) may not be unique. We only consider the match with the shortest driving distance for that combination of the driver and rider(s).

Let E represent the set of all edges in the bipartite graph and let the binary decision variable x_e for edge $e \in E$ indicate whether the edge is in an optimal matching ($x_e = 1$) or not ($x_e = 0$). Further, let E_i and E_j represent the set of edges in E associated with driver i and rider j, respectively. Then, our rideshare matching problem with the objective of maximizing the number of riders who are matched can be formulated as the following integer program:

$$\max z_1 = \sum_{e \in E} \nu_e x_e \tag{3.1}$$

subject to

$$\sum_{e \in E_i} x_e \leq 1 \quad \forall i \in D, \tag{3.2}$$

$$\sum_{e \in E_j} x_e \leq 1 \quad \forall j \in R, \tag{3.3}$$

$$x_e \in \{0,1\} \quad \forall e \in E. \tag{3.4}$$

Objective function (3.1) maximizes the number of matched riders. Constraints (3.2) and (3.3) assure that each driver and each rider is only included in at most one match in an optimal matching, respectively.

To obtain a matching that minimizes the total increase in driving distance for all drivers, the objective should be replaced by:

$$\min z_2 = \sum_{e \in E} \delta_e x_e. \tag{3.5}$$

Since both objectives, i.e. maximizing the number of matched riders and minimizing the total trip time increase for all participants, are relevant, we take them both into account in a hierarchical fashion where we consider z_1 as the primary objective and z_2 as the secondary objective. We first solve (3.1) subject to (3.2) - (3.4). Let z_1^* be the number of matched riders. We then solve (3.5) subject to (3.2) - (3.4) plus the additional constraint $\sum_{e \in E} v_e x_e \geq z_1^*$.

3.4 Solution approach

Our ride-matching algorithm consists of a match identification phase and an optimization phase in which the optimal matching is determined (based on the set of feasible matches). Matches are identified separately for each match type, i.e. rideshare matches, transit matches, and park-and-ride transit matches. To identify feasible matches with two riders, we take advantage of the property that a match between driver i and riders j and k, $j \neq k$, can be time-feasible (but not necessarily) only if the matches between driver i and rider j and driver i and rider k are both time-feasible.

3.4.1 Determining feasible rideshare matches

Identifying a single rideshare match is a relatively simple task. In this study, we perform a straightforward enumeration of all possible driver-rider pairs and identify those that are feasible by checking the conditions outlined in Section 3.3.1.

3.4.2 Determining feasible transit matches

Identifying feasible transit matches starts with several preprocessing procedures. For each announcement $a \in A$, we determine the closest public transit station $s \in S$ to the destination location d_a. This is done by performing a query in a $k-d$ tree in which all transit station locations are stored. Let s_a denote the closest public transit station to the destination location for announcement a. The set of rider announcements $R_p := \left\{ j \in R \mid d_{s_j d_j} \leq M_j \right\}$ is considered in the determination of potential transit matches and the set of driver announcements $D_p := \left\{ i \in D \mid d_{s_i d_i} \leq M_i \right\}$ is considered in the determination of potential park-and-ride transit matches. Note that all driver announcements $d \in D$ are considered when determining potential transit matches.

The next preprocessing procedure serves to find for each rider $j \in R_p$ the set of feasible origin transit stations S_j and the set of feasible departures at each feasible origin station $s \in S_j$. For a rider $j \in R_p$, a train departure time t at origin station $s \in S_j$ is feasible if $e_j + \tau + t_{o_j s} \leq t$ and $t \leq l_j - \hat{t}_{s_j d_j} - \bar{t}_{s s_j}$, i.e. the earliest possible arrival time at the station is less than t and the earliest arrival at the destination departing at or after t is less than l_j. All feasible departures are stored for each individual rider separately.

In another preprocessing step, we find for each driver $i \in D$ the set of feasible transit stations S_d and the set of feasible departures at each feasible transit station $s \in S_d$ that he can visit. For a driver $i \in D$, a transit station s is feasible if he can visit this station on his way from his origin to his destination location without exceeding his maximum acceptable trip duration. A station departure t is feasible if the driver's arrival occurs at or before $t - \tau_s$. To identify such departures, we build an implied time window $[e_s^i, l_s^i]$ at station s with the earliest arrival time $e_s^i = e_i + t_{o_i s} + \tau$ and the latest departure time $l_s^i = l_i - t_{sd_i}$. All departures t for which $t - \tau_s \in [e_s^i, l_s^i]$ are feasible. A lookup table is constructed that enables one to query potentially feasible drivers based on origin station and departure. Feasible single-rider transit matches are identified for each rider by inspecting all feasible stations and feasible train departures for that rider and retrieving all potentially compatible drivers. The feasibility of a given match is checked by examining the conditions that are outlined in Section 3.3.2.

3.4.3 Determining feasible park-and-ride matches

To identify feasible park-and-ride matches, a slightly different preprocessing procedure for the drivers is performed compared to that described in the previous section. The (only) difference is that the latest departure time l_s^i is calculated using the transit trip duration from s to s_i, the destination station of the driver, i.e. $l_s^i = l_i - \widehat{s_i d_i} - \bar{t}_{ss_i} - \tau_s^p$. Based on the preprocessed feasible departures of the riders and drivers, feasible single-rider park-and-ride matches are identified for each rider by inspecting all feasible stations and feasible train departures for that rider and retrieving all potentially compatible drivers. The feasibility of a particular park-and-ride match is checked by examining all conditions that are outlined in Section 3.3.3.

3.4.4 Determining feasible transit and park-and-ride matches with two riders

Once all feasible single-rider transit (or park-and-ride) matches are known, we group all matches of this type using a hash map on driver-station combinations. We examine each individual driver–station combination and construct all potential pairs of riders. We evaluate all possible combinations of departures for a driver–station–rider pair combination to identify and record the best feasible one according to total trip time increase for the participants in a match. Once all feasible transit and park-and-ride matches are known, we determine the best match for a particular driver–rider pair match.

It is straightforward to extend this logic to iteratively identify matches with more than two riders. To identify all feasible matches involving n riders, one needs to examine each feasible grouping of a driver–station combination and a set of $n-1$ riders and construct and examine the feasibility of all possible groupings of the driver–station combination and n riders (where the initial $n-1$ riders form a subset of the n riders). The number of groupings that one needs to examine during this process is typically small because the earliest and latest arrival time constraints as well as the maximum detour and maximum trip duration constraints rule out most of the possible groupings.

3.5 A computational study

In this section, we report the results of our computational study designed to assess the possible benefits and synergies that can arise from integrating a ridesharing system and a public transit system.

3.5.1 Generation of instances

We focus on a rectangular metropolitan area 20 by 10 miles in size that features a circular urban center with a radius of 2.5 miles and a sprawling suburban area.

We consider a stylized transit network that captures the key features of a real-world commuter train network, such as the Bay Area Rapid Transit (BART) network in the San Francisco region.

The radial route network consists of two commuter train lines and four urban rapid transit lines. Transfers between the different train lines are only possible at the transfer hub, which is located in the center of the urban area. The inter-station distance is approximately 2.25 mi on commuter lines and 0.75 mi on the urban transit lines. Figure 3.2 illustrates the transit lines, stations, and the urban and suburban regions. The figure also indicates the eight park-and-ride stations.

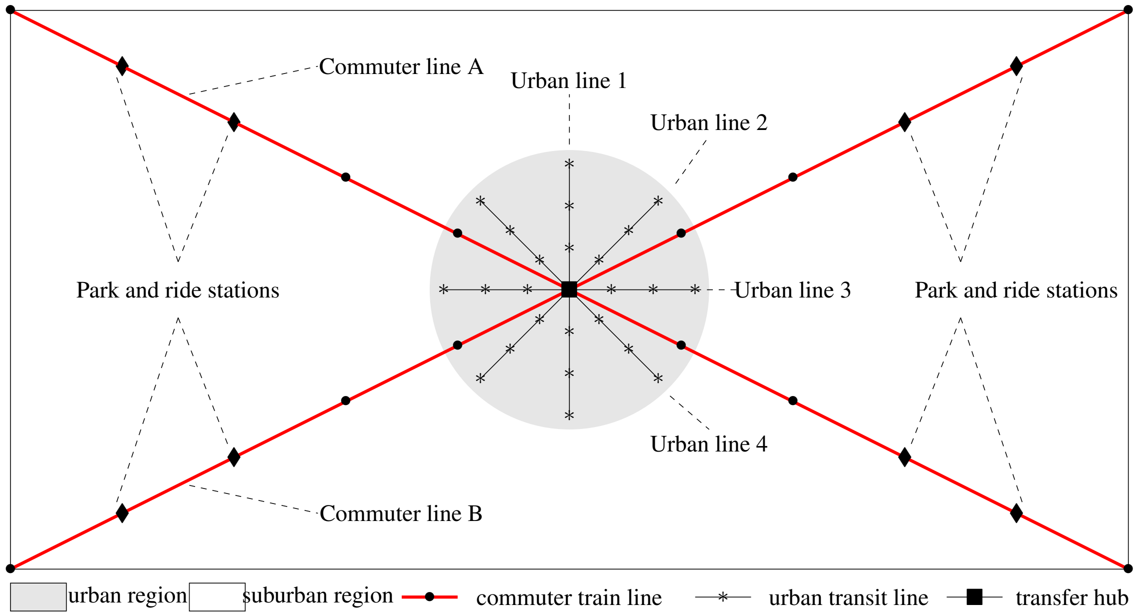


Figure 3.2: Representation of a public transit network and urban/suburban regions

We generate n participant trips in which the probability of a driver or rider trip is equally likely. For each trip announcement, we generate an earliest departure time based on a truncated normal distribution with a standard deviation of 30 minutes. To create commuter trips, we randomly draw origins from the complete region and draw destinations from the urban center. Since we assume that it is unlikely participants with very short trips will participate, we use rejection sampling to filter out trips that are shorter than 1 mile.

Travel times on roads are calculated using Euclidean distances with a 30% uplift. We assume an average driving speed of 20 mph, which represents the travel speed in a congested urban area. We consider a pickup time of two minutes for each rider who is picked up by a driver. We assume a walking speed of 4 feet per second (LaPlante and Kaeser, 2004). The maximum walking distance for the rider to or from a transit station is 0.5 miles, which corresponds to 11 minutes of walking at this speed.

We assume a cyclic schedule with the same departure frequency for all transit lines, with a default departure frequency of 15 minutes. The line schedules are synchronized at the hub location so that all transfers between lines take the same amount of time. The average speed of the commuter train is 40 mph and the average speed of the urban transit train is 20 mph. All trains have a dwell time of one minute for regular stations and three minutes at the hub station to allow for transfers. We assume that it takes two minutes to enter and leave a transit station which captures walking to and from the platform, with an additional two minutes of service time for all park-and-ride matches to account for the time needed to park the vehicle. The characteristics of the base case instances are summarized in Table 3.2.

Table 3.2: Characteristics of the base case instances

Trip pattern:	suburb to center
Avg. number of participants:	1000
Driver-rider ratio:	0.5
Matching flexibility:	20 min
Scheduling flexibility:	15 min
Driver detour flexibility:	25%
Rider flexibility:	50%
Maximum number of driver stops:	3
Avg. driver trip distance:	8.0 mi
Avg. driver trip duration:	24.1 mi
Max. walk distance to transit station:	0.5 mi
Walk speed:	4 ft/s
Car speed:	20 mi/h
Suburban train speed:	40 mi/h
Urban train speed:	20 mi/h
Vehicle capacity:	2 seats
Pickup time per rider:	2 min
Transfer time street to platform:	2 min
Additional transfer time for park-and-ride:	2 min
Train dwell time:	1 min
Train hub dwell time:	3 min
Frequency of train departures:	15 min
Number of stations:	41
Number of park-and-ride stations:	8

3.5.2 Computational results

The main aim of this research is to analyze and quantify the benefits of integrating ridesharing with public transit and to determine what value this can create for different stakeholders, e.g. public transport agencies, rideshare providers, and system participants. We use the optimization results to compute and evaluate a number of metrics to gain insight into the system performance and the potential of ridesharing to enhance mobility and increase the use of public transport. In all experiments, we use either the base case setting or a setting in which one of the characteristics is varied in order to assess the sensitivity of the system performance to this characteristic.

The algorithm for generating feasible matches is implemented in C and the simulation framework is implemented in Python 3.4. CPLEX 12.6 is used for solving matching problems. All base case instances are solved within a few seconds, with solution times of up to a minute for the largest instances with 2,000 participants.

3.5.3 Benefits of an integrated system

To evaluate the benefits of integrating ridesharing and public transit, we analyze the matching rates for several different settings. As a benchmark, we consider a setting in which only door-to-door rideshare matches are generated, denoted by RS. Next, we consider settings in which both rideshare and transit matches are generated, where TRS1 denotes the setting in which only single-rider transit matches are generated and TRS2 denotes the setting in which transit matches with one or two riders are generated. Finally, PTRS denotes the setting in which, in addition to the matches considered in TRS2, a driver may opt to take public transport after dropping off the riders at a transit station, i.e. park-and-ride matches. The results for the base case can be found in Table 3.3 where we report averages over 10 randomly generated instances. When we report the percentage of transit and P+R matches, it is relative to the number of matches, i.e. if 80 out of 100 riders are matched and 36 of them are dropped off at a transit station by drivers who then drive to their final destinations, and 4 of them were dropped off at a transit station by drivers who then take public transport to their destinations, the matching rate is 80%, the transit matching rate is 45%, and the P+R matching rate is 5%. Further, when we report travel time increases for the riders, it is relative to the travel time they would have needed if they had driven themselves.

We observe that integrating ridesharing and public transit can have significant benefits. The average number of matched riders (our primary objective) increases from 66.8% to 83.8%. Interestingly, the average length of the driver detour (our secondary objective) in an integrated system is smaller than in a rideshare-only system, i.e. 7.2% compared to 8.4%. Thus, not only is mobility enhanced but the negative externalities associated with car travel, such as emissions and congestion, are reduced. This reduction of the average length of the driver's detour is due, in part, to drivers deciding to use public transport. Perhaps equally important is the fact that the average increase in travel time for riders is relatively small, i.e. 7.5%. This may encourage more people to consider public transport as a viable alternative to traveling by car.

Another important observation is that to achieve a high matching rate, it is critical that a driver is willing to pick up and drop off more than one rider (in our setting at most two are

Table 3.3: Results for different rideshare settings (avg. over 10 random base case instances)

	RS	**TRS1**	**TRS2**	**PTRS**
Riders				
Matching rate (%):	*66.8*	*74.0*	*83.7*	*83.8*
Transit matches (%):	*0*	*32.4*	*37.2*	*33.9*
P+R matches (%):	*0*	*0*	*0*	*4.0*
Δ travel time (%):	*0*	*7.3*	*7.4*	*7.5*
Drivers				
Matching rate (%):	*68.3*	*75.5*	*73.8*	*74.2*
Transit matches (%):	*0*	*32.4*	*27.3*	*25.1*
P+R matches (%):	*0*	*0*	*0*	*3.3*
Δ travel time (%):	*19.1*	*17.1*	*21.6*	*21.9*
Δ distance (%):	*8.4*	*7.0*	*10.4*	*7.2*

dropped off at the same transit station). If drivers are only willing to pick up and drop off a single rider, the matching rate is 74% whereas if drivers are willing to pick up and drop off two riders the matching rate is 83.7%. Of course, this increase in the matching rate comes at the expense of a longer average driver detour (from 7.0% to 10.4%).

In Figure 3.3, we provide more details on the matches found for riders by showing the number of riders in the different types of matches for the four settings. We again see that transit matches involving two riders are critical to achieving high matching rates in the settings that allow them (TRS2 and PTRS). We also see that the number of riders in park-and-ride matches is small (only 17 on average).

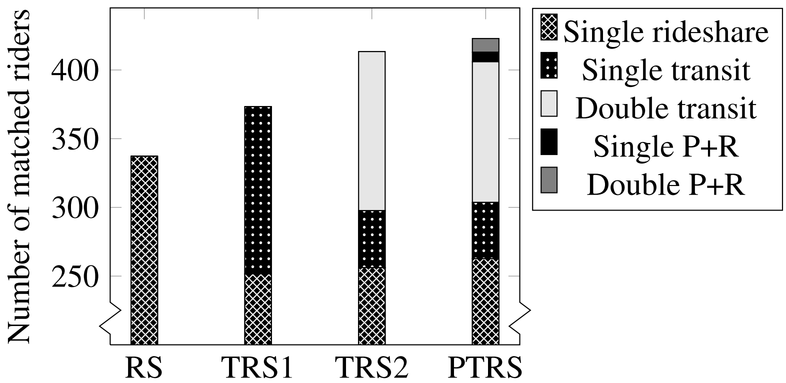


Figure 3.3: Number of matched riders in different match types

The fact that allowing park-and-ride matches results in an increase in the matching rate may, at first, seem surprising as it mostly impacts drivers. However, some transit matches are time-feasible only when the driver uses public transport because the driver can reach his destination faster using public transport than by car. Another factor contributing to the small number of drivers choosing to use public transport is the use of hierarchical optimization, i.e. maximizing the matching rate followed by minimizing the (total) driver distance subject to the constraint that the matching rate cannot decrease. The constraint that the matching rate cannot decrease is restrictive and leaves little flexibility for reducing the detour distance. If we are willing to accept slightly lower matching rates, it is possible that we would see many more drivers opting for public transport. Finally, the location of the transit stations offering park-and-ride, which in our transit network are relatively far from the urban center, (negatively) impacts the benefits of the park-and-ride option.

To provide further insight into the opportunities offered by an integrated system, we look at the trips associated with the transit and park-and-ride matches generated for a single instance. Figure 3.4 shows the paths from the riders' origins to their transit stations. For matches involving two riders, the trip from the first rider's origin to the second rider's origin is represented by a dotted line and the trip from the second rider's origin to the transit station is represented by a solid line. Interestingly, we see that only about 25% of the riders is dropped off at the station that is closest to their origin. It is often more convenient for a driver to use a transit station closer to the urban center. As a consequence, the public transport trips for riders tend to be relatively short. This also suggests that park-and-ride facilities at transit stations located close to the urban center may significantly increase drivers' use of public transport.

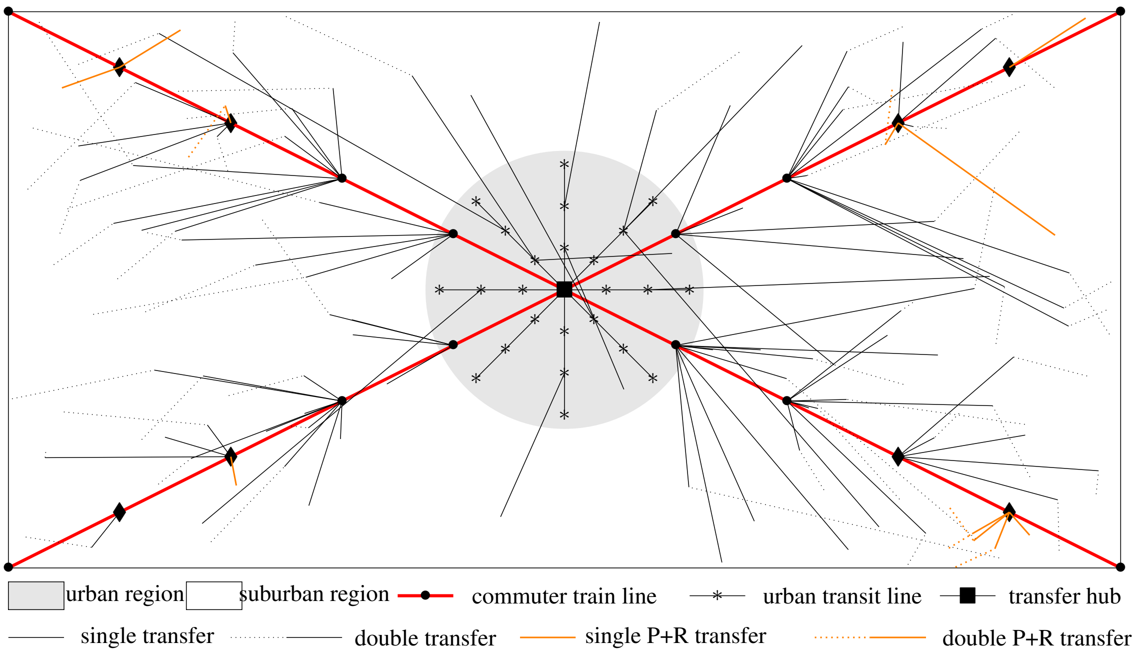


Figure 3.4: Map of rider paths to transit stations for one of the base case instances for the PTRS setting

In the next subsections, we evaluate the impact of varying instance characteristics. We will report results only for the most and the least restricted rideshare settings, i.e. RS and PTRS.

3.5.4 Impact of driver matching flexibility

In this section, we evaluate the impact of the drivers' matching flexibility, i.e. their willingness to extend the latest arrival time at their destination. Table 3.4 shows, unsurprisingly, that lower driver matching flexibility (i.e. 10 min) limits the value of integrating ridesharing with public transit because fewer two-rider transit matches are feasible (there is simply not enough time to accommodate more than one rider).

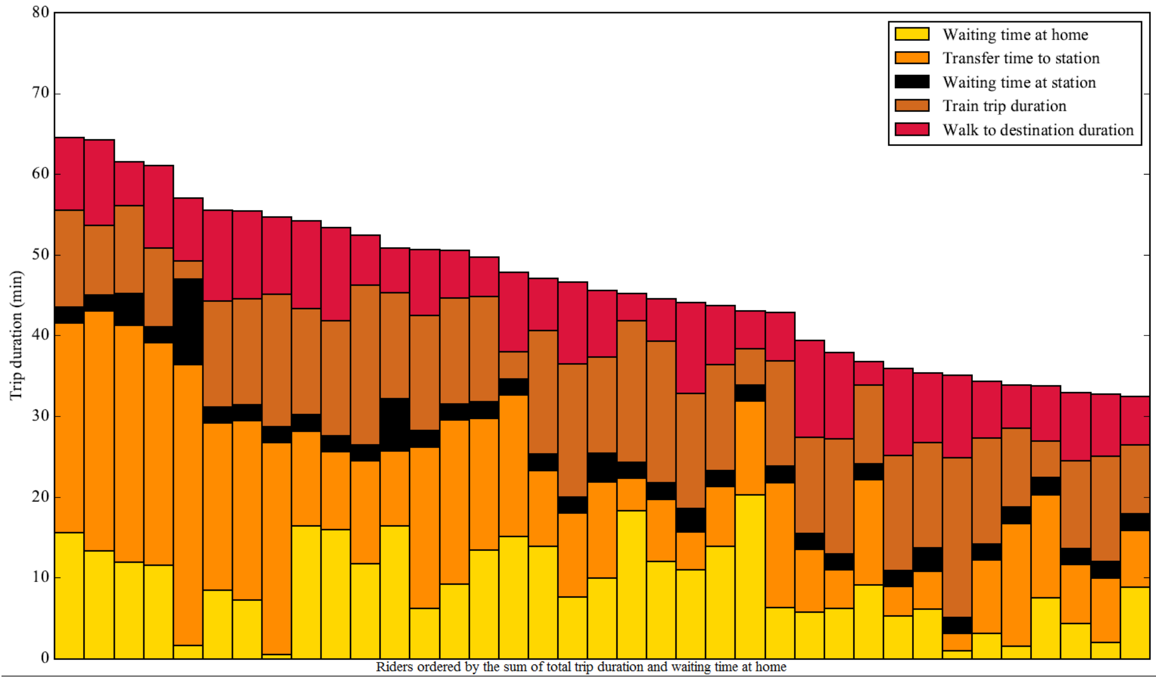


Figure 3.5: Breakdown of trip durations and waiting time at home for all riders matched in single transit matches in the optimal solution for one of the base case instances – an individual bar in the pane represents the breakdown of the itinerary for a rider in a match (riders are ordered based on the sum of total trip duration and the waiting time at home)

Table 3.4: Results for different driver matching flexibilities (Avg. over 10 random instances)

	10 min		15 min		20 min	
	RS	**PTRS**	**RS**	**PTRS**	**RS**	**PTRS**
Riders						
Matching rate (%):	*64.9*	*78.8*	*66.8*	*83.8*	*66.3*	*84.0*
Transit matches (%):	*0*	*34.3*	*0*	*33.9*	*0*	*38.7*
Drivers						
Matching rate (%):	*65*	*70.8*	*68.3*	*74.2*	*66.7*	*73.7*
P+R matches (%):	*0*	*3.0*	*0*	*3.3*	*0*	*3.8*
Δ distance (%):	*8.1*	*6.4*	*8.4*	*7.2*	*8.5*	*7.0*

3.5.5 Impact of density

In this section, we investigate how the number of system participants affects the performance. Table 3.5 shows, as expected, that the matching rate increases significantly when the number of system participants rises; the net improvements in the rider matching rate are 14.4 to 17.6 percentage points.

Table 3.5: Results for different numbers of participants in the system (Averaged over 10 random instances)

	500		1000		2000	
	RS	**PTRS**	**RS**	**PTRS**	**RS**	**PTRS**
Riders						
Matching rate (%):	53.5	67.9	66.8	83.8	76.0	93.7
Transit matches (%):	0	39.0	0	33.9	0	36.5
Drivers						
Matching rate (%):	56.3	63.6	68.3	74.2	74.1	78.7
P+R matches (%):	0	4.6	0	3.3	0	4.3
Δ distance (%):	7.1	4.7	8.4	7.2	8.8	6.7

3.5.6 Impact of transit system parameters

In this section, we study the impact of transit line characteristics on the system performance. We consider three speed scenarios for the suburban trains: 30 mph, 40 mph (base case), and 50 mph and three departure frequencies for all transit lines: every 5, 15 (base case), or 25 minutes. Figure 3.6 shows the rider matching rates for the nine combinations. As expected, we see that more frequent departures and faster trains improve the performance of the system and create more rider matches. The results also suggest that the departure frequency is more important than the speed, i.e. only a marginal benefit arises from increasing the suburban train speed from 40 mph to 50 mph.

We also see that more drivers and riders use public transit (i.e. the number of drivers and riders who are matched in transit and park-and-ride matches increases). Also as expected, less frequent departures and slower trains tend to negatively impact the system's performance in terms of both the percentage of matched riders and the percentage of riders and drivers who use public transit. In the worst case, with a train frequency of 25 minutes and speeds of 30 mph, 77.75% of riders are matched on average and 22.11% are matched in transit and park-and-ride matches. In the best case, 86.86% of riders are matched and 35.45% of riders are matched in transit and park-and-ride matches, on average.

In the second experiment, we examine the interplay between the frequency of train departure and the driver matching flexibility. This interplay is interesting because the matching flexibility determines which train departures can be accommodated by a driver – if the matching flexibility is low and departures are infrequent, few, if any, departures can be accommodated by a driver and vice versa. The results in Table 3.7 confirm our expectations. Combinations of low matching flexibility (10 min) and infrequent departures (25 min) result in poor performance – only approximately 20% of riders are matched in transit matches of park-and-ride matches and less than 2% of drivers are matched in park-and-ride matches. Further, combinations of high matching flexibility and a high frequency of train departures

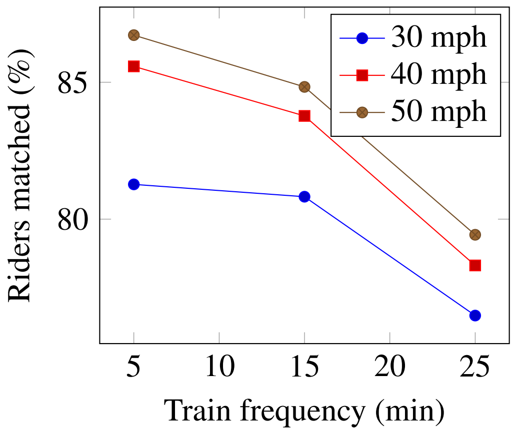


Figure 3.6: Rider match rates for different transit line characteristics

Table 3.6: Sensitivity analysis: Train departure frequency and Suburban train speed

Frequency / Speed		**30 mph**		**40 mph**		**50 mph**	
		Drivers	Riders	Drivers	Riders	Drivers	Riders
5 min	Matched (%):	*73.27*	*83.19*	*73.22*	*83.03*	*73.37*	*86.86*
	PT users (%):	*2.25*	*29.97*	*3.44*	*33.56*	*4.48*	*35.45*
15 min	Matched (%):	*72.25*	*81.03*	*74.2*	*83.77*	*73.12*	*83.38*
	PT users (%):	*1.48*	*27.34*	*2.41*	*31.71*	*3.83*	*34.9*
25 min	Matched (%):	*70.4*	*77.75*	*70.48*	*78.79*	*69.8*	*79.2*
	PT users (%):	*1.13*	*22.11*	*2.28*	*26.08*	*2.6*	*26.2*

create more opportunities to create (transit) matches and produce a high rider matching rate (90.07%) and a large percentage of riders matched in transit matches (34.76%).

It is perhaps surprising that the shares of park-and-ride and transit matches in the setting with the lowest matching flexibility and least frequent train departures are still quite high. A possible explanation for this is the fact that numerous transit stations and transit lines are considered for a drop-off for each match. This creates greater opportunities to find a feasible drop-off station and a feasible drop-off time than if just the nearest station (or a smaller set of stations) were to be considered. The findings of this section lead us to conclude that: (1) the integration of ridesharing and public transit creates value across a wide range of speed and frequency properties of a public transport network, but that (2) it is (of course) beneficial for the system if the trains are fast, the departures are frequent, and the drivers are flexible in their schedule.

Table 3.7: Sensitivity analysis: Train departure frequency and Driver matching flexibility

Frequency / Match flex.		**10 min**		**20 min**		**30 min**	
		Drivers	Riders	Drivers	Riders	Drivers	Riders
5 min	Matched (%):	*71.29*	*79.24*	*74.68*	*87.28*	*76.43*	*90.07*
	PT users (%):	*3.36*	*30.76*	*3.76*	*34.1*	*4.31*	*34.76*
15 min	Matched (%):	*68.74*	*78.68*	*73.34*	*84.75*	*76.38*	*87.66*
	PT users (%):	*1.86*	*28.38*	*3.41*	*32.17*	*3.46*	*32.9*
25 min	Matched (%):	*68.82*	*72.57*	*72.5*	*84.32*	*76.67*	*85.84*
	PT users (%):	*1.94*	*19.78*	*3.04*	*28.86*	*3.31*	*28.89*

3.6 Conclusion

Our study has shown that the integration of a ridesharing system and a public transit system can significantly enhance mobility, increase the use of public transport, and reduce the negative externalities associated with car travel.

We found that driver willingness to pick up and drop off more than one rider is critical to the system performance. We investigated only the simplest and most convenient (from a driver's perspective) variant, i.e. driver willingness to pick up two riders and drop them off at the same transit station. Further investigation of more flexible variants is warranted. We also observed the potential of park-and-ride facilities, but more experimentation is needed to fully understand the benefits, i.e. the location and number of transit stations with park-and-ride facilities as well as the use of different objective hierarchies when generating rideshare and transit matches. Other opportunities for further research include analyzing more complex transit systems, the use of meeting points in order to streamline the process of picking up riders, and the integration of (free) bike-sharing systems.

4 ON THE EFFICIENT GENERATION OF FEASIBLE DRIVER RIDER MATCHES IN RIDESHARING SYSTEMS

4.1 Introduction

Ridesharing platforms enable people with similar itineraries and time schedules to share rides. Platforms seeking to provide high-quality automated matching need to efficiently solve ride-matching problems: Given a set of riders and a set of drivers with associated itineraries, schedules, and other preferences, one needs to find an optimal matching with respect to the total distance savings or an alternative objective. An exact solution approach requires that all feasible matches be found by checking each possible driver–rider pair. The resulting feasible matches are then used to construct a matching problem, which can be solved very efficiently using a state-of-the-art LP solver.

We study how to identify all feasible driver–rider matches more efficiently. This can have an important effect on the total runtime of the algorithm because, typically, only a very small fraction of the possible matches are feasible, meaning it is possible to do much better if we do not have to fully evaluate all pairs. We explore two ideas: (1) direct drive times from origin to destination can be used to efficiently identify those riders who have sufficiently small drive times to be matched with a particular driver, and (2) rider time windows can be stored in a memory structure that allows one to find riders with time windows that overlap with the time window of a driver in sub-linear time. We develop and test a data structure that combines optimizations (1) and (2) and test its performance.

To the best of our knowledge, the only related research on the efficient generation of driver–rider matches was undertaken by Geisberg et al. (2010). They develop an approach based on contraction hierarchies to efficiently compute detours to match ridesharing offers and requests in real road networks. In this chapter, we look at how to efficiently identify matches with compatible trip times.

4.2 Problem description

We are provided with a set of trip announcements S. Each trip announcement $s \in S$ is associated with an origin location o_s and a destination location d_s as well as an earliest departure time e_s and a latest arrival time l_s. We assume the participants' departure times are somewhat flexible so that the difference $l_s - e_s$ is greater than the travel time from origin to destination. The set of announcements S can be partitioned into $D \subset S$, the set of trip announcements by the drivers, and $R \subset S$, the set of trip announcements by the riders. Each driver $i \in D$ also specifies a maximum trip duration T_i, which implies the extra time the driver has available to accommodate a rideshare. We will denote the travel time between locations i and j with t_{ij}.

This chapter is based on Stiglic et al. (2015b).

The data structure we are designing needs to support one query type: *find and return all riders $j \in R$ with a travel time that is compatible with the travel time of the given driver $i \in D$.* A rider's and a driver's travel time are compatible if: (1) the time windows of the driver and the rider overlap by at least the shortest time needed to complete the trip from the rider's origin to the rider's destination; and (2) the trip duration from the rider's origin to his destination is smaller or equal to the sum of the trip duration from the driver's origin to his destination and the maximum detour duration the driver is willing to incur.

To check if a match is feasible, we construct an implied time window within which driver $i \in D$ can pick up rider $j \in R$ at o_j. For a match to be feasible, this time window needs to overlap with the implied pickup time window of rider j. This constraint can be expressed in the following form:

$$\max(e_i + t_{o_i o_j}, e_j) \leq \min(l_i - t_{d_j d_i} - t_{o_j d_j}, l_j - t_{o_j d_j}) \tag{4.1}$$

Also, the trip of driver $i \in D$ that is associated with this match must be smaller than T_i, i.e. the sum of the travel times associated with the three legs of the trip must not exceed T_i:

$$t_{o_i o_j} + t_{o_j d_j} + t_{d_j d_i} \leq T_i \tag{4.2}$$

4.3 Meeting points

The problem described above does not take into account the concept of meeting points introduced in Chapter 2. However, this case can easily be addressed by constructing implied time windows. Consider Figure 4.1. Let t_j^{max} denote the time needed to drive distance d_j^{max}, the longest distance rider j is willing to walk to and from a meeting point. Thus, the minimum time the rider and driver will share is $t_j^{min} = t_{o_j,d_j} - 2t_j^{max}$. Therefore, we redefine the time window for rider j to be $[e_j + t_j^{max}, l_j - t_j^{max}]$.

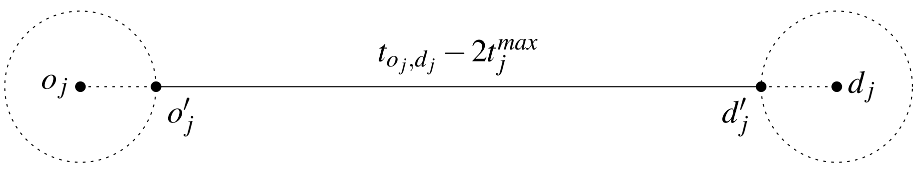


Figure 4.1: Minimum shared ride time for rider j

4.4 Basic approach

We may consider drivers one by one and find all time- and cost-feasible matches for each driver. A straightforward enumeration algorithm with runtime complexity O(nm) finds all feasible single-rider matches, where n is the number of drivers and m is the number of riders. When the number of participants is large, it becomes computationally expensive to determine the time- and cost-feasible single matches in this way, especially if one considers that for each driver–rider pair one needs to fully evaluate one must also find the shortest path for the shared trip.

4.5 Interval tree

Interval trees are designed to efficiently find intervals that overlap. It takes $O(log(m))$ time to find an overlapping interval in a balanced interval tree, where m is the number of intervals in the tree. Unfortunately, in the ridesharing setting, as many as $1/2$ of the rider time windows may overlap with a driver time window on average. Hence, if we denote the expected ratio of overlapping rider intervals by r_{exp}, we cannot expect the performance to be better than $O(r_{exp} \cdot log(m))$ with a standard interval tree. In practice, the performance of an interval tree might even be inferior to a linear search due to issues with locality of reference.

A query in a standard interval tree returns all overlapping intervals – including those with a very small overlap. However, the time window of a driver and a rider has to overlap by at least the time needed to complete the shared part of their trip. Consider Figure 4.2. It shows three possible ways in which the time window of a rider j can overlap with the time window of a driver i.

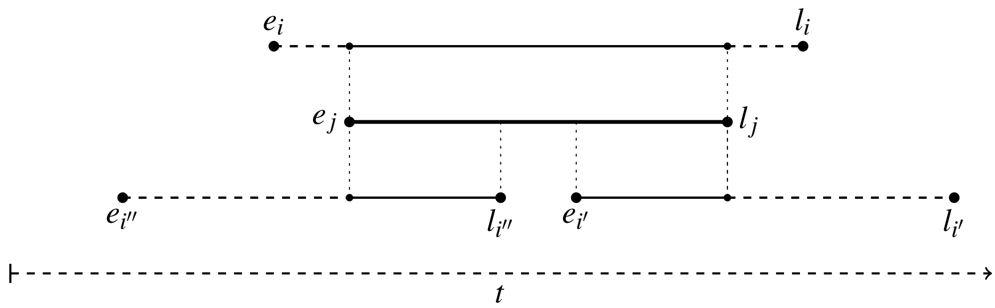


Figure 4.2: Overlaps of the time interval of rider j with queries corresponding to three drivers i,i',i''

The interval of the rider is stored in the memory structure, and the three intervals related to drivers i, i', and i'' represent three possible queries. The overlap can be from the right of interval $[e_j, l_j]$, as with $[e_{i'}, l_{i'}]$, it can be from the left of interval $[e_j, l_j]$, as with $[e_{i''}, l_{i''}]$, or it can span the entire interval $[e_j, l_j]$, as with $[e_i, l_i]$.

Using this simple observation, we see that if the overlap in the query has to be at least t_j^{min}, then we can shorten the rider intervals $[e_j, l_j]$ in our memory structure by t_j^{min}. We can shorten the rider intervals from the left *and* from the right. That is, we shorten the intervals by adding t_j^{min} to the lower end and by subtracting t_j^{min} from the upper end, i.e. $[e'_j, l'_j] = [e_j + t_{o_j,d_j} - t_j^{max}, l_j - t_{o_j,d_j} + t_j^{max}]$. This may result in: (1) $e'_j < l'_j$ or (2) $e'_j \geq l'_j$. For each of these two cases, we analyze the three possible types of driver queries.

Figure 4.3 depicts Case (1): the intervals of drivers i, i', and i'', represent all possible types of overlap.

Queries in an interval tree with reduced rider intervals for drivers i' and i'' will not return rider j as a compatible rider. The query for driver i, on the other hand, will return rider j

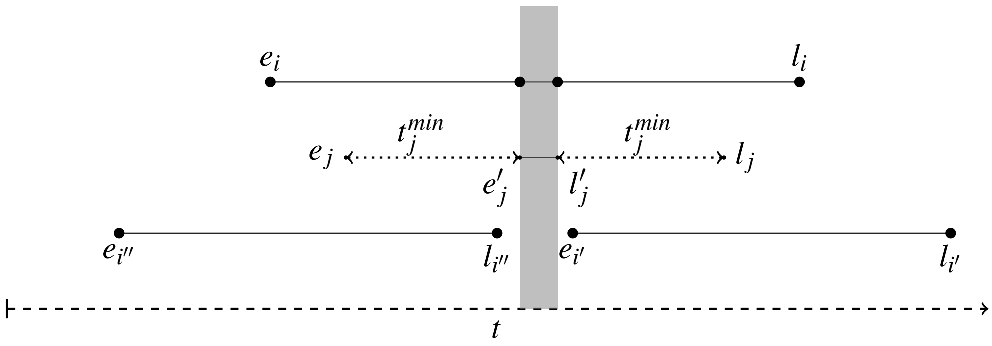


Figure 4.3: Shortening the time interval of rider j and overlaps with queries corresponding to drivers i, i', and i''

since the overlap is greater than t_j^{min}. Thus, using reduced intervals in a standard interval tree produces the desired results. Let us now consider Case (2). If $e'_j \geq l'_j$ (as in Figure 4.4), we have an inverted interval $[l'_j, e'_j]$. In such a situation, there is sufficient overlap only if interval $[l'_j, e'_j]$ is a sub-interval of interval $[e_i, l_i]$.

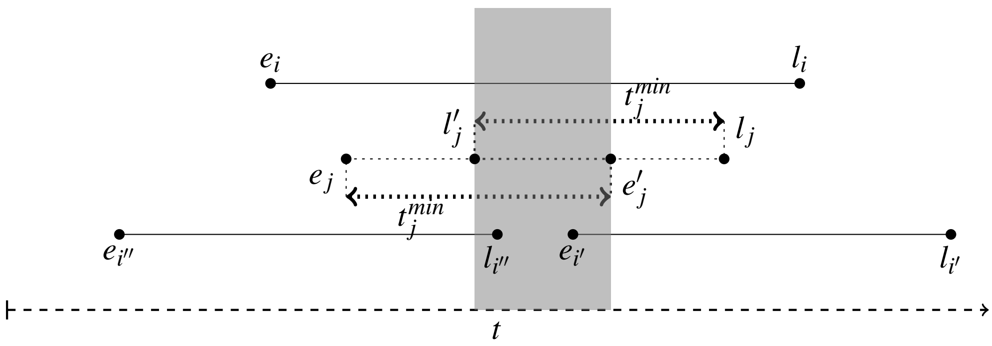


Figure 4.4: Inverted time interval of rider j and overlaps with queries corresponding to drivers i, i', and i''

The intervals of drivers i' and i'' have some overlap with the inverted interval $[l'_j, e'_j]$, but only the interval of driver i has sufficient overlap.

Queries using a standard interval tree with reduced time windows will return all time-feasible riders for a driver but, unfortunately, may also return some time-infeasible riders (i.e. the driver's interval is too small to accommodate the rider). This disadvantage is outweighed by the fact that the intervals in the interval tree are much smaller and, thus, the queries are much faster. However, the number of riders returned can be quite large. Below, we propose an alternative approach that uses sorted lists.

4.6 Sorted interval list

We maintain a list of intervals, each with a lower end l and an upper end u. We first sort the intervals in non-decreasing order of their lower end. We then recursively compute an auxiliary upper end u^* for each interval in the sorted list. The auxiliary upper end is defined as $u_i^* = max(u_{i-1}^*, u_i)$ for an interval in position i in the list (for all positions $i \geq 1$) and $u_0^* = u_0$ for the interval in position 0. Intervals for which $u = u^*$ are super-intervals – they

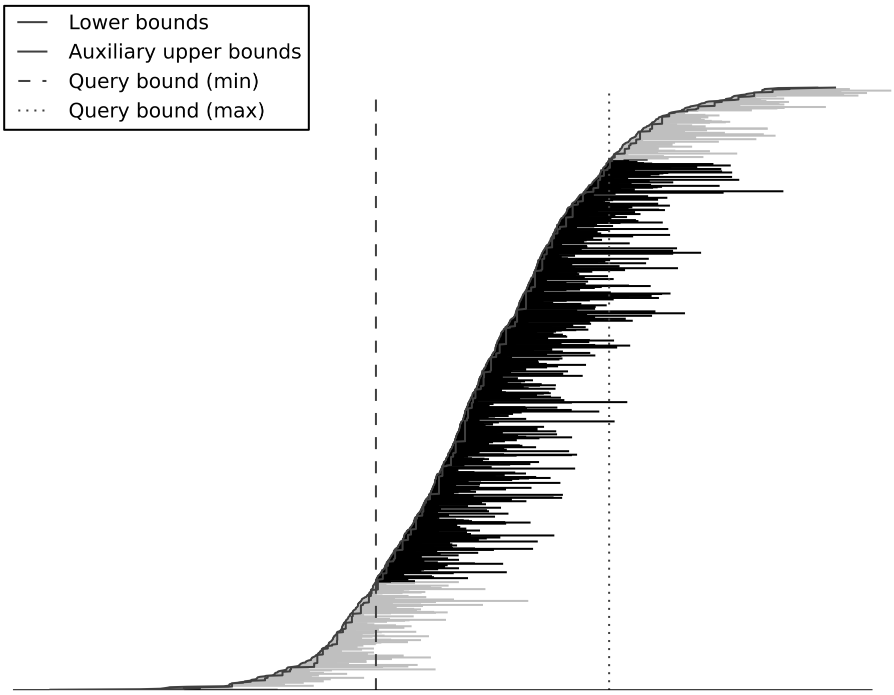


Figure 4.5: Visualization of a query in an interval list

are not a sub-interval of any other interval in the list. Intervals for which $u < u^*$ are sub-intervals – they are a subset of at least one interval in the list.

If we are in Case (1), we find all intervals that potentially overlap with $[t_{low}, t_{high}]$ by finding the position of the leftmost interval in the sorted list with $t_{u*} \geq t_{low}$ and the position of the rightmost interval in the sorted list with $t_l > t_{high}$ using binary search. The two positions define the sublist we want. For Case (2), the query is even simpler: we first find the position of the leftmost interval in the sorted list with $t_l \geq t_{low}$ and the position of the rightmost interval in the sorted list with $t_l > t_{high}$. These positions define the sublist that contains all potential sub-intervals of $[t_{low}, t_{high}]$. We add the two resulting lists together and further refine them. Figure 4.5 visualizes how the intervals are sorted and how a query works (i.e. which intervals it returns).

The speed of the search may be further improved by breaking up the sorted list into sublists. For each super-interval, we create a sublist containing all its sub-intervals. We only keep the super-intervals in the original list. Each sublist may be further broken down to a desired level of granularity (i.e. the minimum number of intervals in a sublist). We know that all the intervals in a sublist of an interval are within its bounds so we may use this information to explore only the relevant sublists. This approach can improve the search in large data sets, especially when the super-intervals are large compared to the sub-intervals.

4.7 Interval container

We finally propose an approach using a container with n bins. The positions of the bins in the container are ordered with respect to the riders' trip durations. By using bins to group riders with similar ride durations, we limit the search to relevant subsets of bins. Each bin features an interval container as described in Section 4.6, but we do not break the interval lists up into sub-lists. By grouping intervals into bins, we ensure the intervals within each bin have a similar length. This helps in performing efficient queries inside the interval lists.

Let us first discuss the construction of the bins. We compute $T_{min} = \min_{j \in R} T_j$ and $T_{max} = \max_{j \in R} T_j$. We then compute a step size $h = (T_{max} - T_{min})/n$, which is used to determine a critical value for each bin. The critical value for a bin in position k is $v_k = T_{min} + hk, \forall k \in [1, ..., n]$. We then sort all riders into bins so that all riders in the bin at position $k \in [2, ..., n]$ have a ride time smaller than or equal to v_k and greater than v_{k-1}. Riders with ride times smaller than v_1 are sorted into the bin in position 0.

We proceed by creating an interval container in each bin. We first shorten all rider intervals in a given bin as suggested in the previous section. We may shorten the interval of a rider by adding T_j to e_j and subtracting T_j from l_j. Hence, $e'_j = e_j + T_j$ and $l'_j = l_j - T_j$. If $e'_j \leq l'_j$, we have a normal interval $[e', l']$, and we have an inverted interval $[l', e']$ otherwise. Next, we create a list of normal intervals and a list of inverted intervals. Each interval has a lower end l and an upper end u. We first sort the intervals in non-decreasing order of their lower end. We then recursively compute an auxiliary upper end u^* for each interval in the sorted list. The auxiliary upper end is defined as $u^*_i = \max(u^*_{i-1}, u_i)$ for an interval in position i in the list (for all positions $i \geq 1$) and $u^*_0 = u_0$ for the interval in position 0.

Queries are performed as follows. For driver i, with time interval $[t_{low}, thigh]$, we first identify the bins for which, $v_k \leq T_i, k \in [1, ..., n]$, using bisection search. For each bin we identify, we explore the interval container to find and return all riders with overlapping intervals. We perform one query in the part that contains normal intervals and one query in the part that contains inverted intervals of the interval container. With normal intervals, we find the position of the leftmost interval in the sorted list by using binary search. The two positions define the sub-list we want. With inverted intervals, these two positions are the leftmost interval in the sorted list with $t_l \geq t_{low}$ and the rightmost interval in the sorted list with $t_l \geq t_{high}$. We add all the resulting sub-lists together and further refine them using inequalities (1) and (2).

4.8 Heuristic approach

Finally, we also propose a heuristic strategy. Instead of identifying the bins for which $v_k \leq T_i, k \in [1, ..., n]$, for driver $i \in D$, we propose to only consider those for which $v_k \leq t_{o_i d_i}$. By doing so, we ignore all riders who have ride times larger than the drive time of the driver. In the instances we have experimented with, this means we overlook 2% of the possible matches. However, these are typically matches that are less attractive to a driver because the origin, the destination, or both are normally not well-aligned with the driver's original route. Such a strategy may decrease query times by an additional 25% on average. Optionally, one could also decide to cut-off riders with short rides compared to that of the driver because the distance saving associated with such a match will be low. Thus, one could add several of such rules to the query, speed up the search, and also have full control over what kind of matches will be identified.

4.9 Performance

In order to test the performance, we generate several ridesharing instances of various sizes using the same approach as in Chapter 3 (based on the travel demand model for the metropolitan Atlanta region). We use the same instance characteristics as in the base case instances, but do not consider meeting points.

We generate eight instances that have between approximately 1,000 and 12,000 participants. We implement four different enumeration techniques in Python 2.7: brute force enumeration, enumeration with interval lists (as in Section 4.6), enumeration with interval bins (as in Section 4.7) and the heuristic (as in Section 4.8). We test on a single core of an i5-3360M machine. Figure 1 presents the test results. The horizontal axis represents the number of possible driver–rider pairs and the vertical axis represents the runtimes for the four techniques.

As can be seen from Figure 4.6, the approach based on interval containers performs best on all instances. The runtime is more than 51% less than for brute force on average. Compared with the interval lists, the best performing approach takes 20% less time to complete. These results were stable over the instances.

4.10 Conclusion

In this section, we presented an approach for identifying all feasible single driver–single rider matches more efficiently. We showed that it can decrease the duration of the enumeration phase by more than 50% compared to a brute force search and 20% compared to interval lists. We also presented a heuristic strategy to speed up the search even further.

One possible direction for future research could be to consider how to use the spatial data associated with the participants' trips in order to partition the data structure even further and speed up the queries even more.

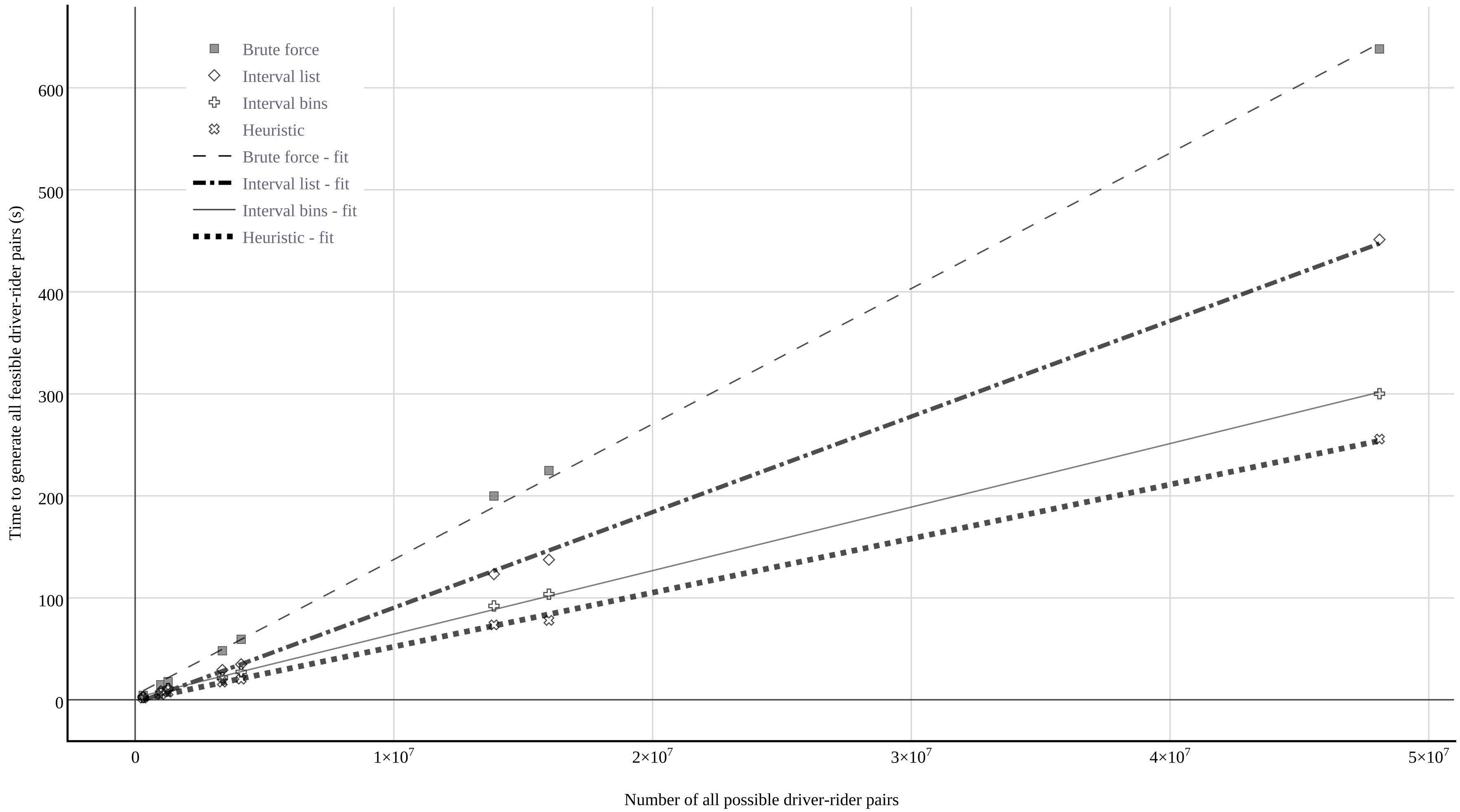


Figure 4.6: Performance of different match-generation methods

5 A MATHEMATICAL FORMULATION FOR THE SINGLE DRIVER–MULTI RIDER MATCHING PROBLEM IN RIDESHARING SYSTEMS

5.1 Introduction

In this section, we present a mixed-integer linear programming model for the single driver–multiple rider matching problem that arises in certain types of ridesharing systems. The problem has many similarities with the dial-a-ride problem (DARP), which is a pickup and delivery problem with ride time constraints. Given the similarities, we borrow certain types of constraints from the DARP literature.

The model we devise allows us to match a single driver with several riders. It also permits participants to opt for driver or rider roles or, alternatively, to let the model determine what is the most desirable role from a system perspective. If there is no match for a rider or a driver, the model is able to add this opportunity cost to the objective function value. The model minimizes the cost of all the trips that have to be performed to move the participants from their origin to their destination nodes.

The model we present can be used to plan rides for a certain period of time (typically a day) for a ridesharing platform (accessed by participants through a web or smartphone application). We assume that on a specific day there is a group of participants that have to make a trip from a specific origin to a specific destination. Each participant can opt to offer a ride to potential riders (a), or ride in someone else's car (b), or alternatively may leave the decision regarding their role to the system. Participants enter their ride requests/offers in the system at least by noon (12.00) for the next day. The following information is available:

- exact origins and destinations of all participants (coordinates and addresses);
- time windows for all participants' trips (earliest time of departure and latest time of arrival);
- is participant a driver, rider, or is he willing to let the system determine what is more desirable;
- number of free seats in a particular vehicle;
- transportation network data (distance matrix, travel time matrix).

The model we present is based on the prior work of Cordeau, Laporte, and Ropke as well as Herbawi and Weber (Cordeau and Laporte, 2003; Cordeau, 2006; Cordeau and Laporte, 2007; Ropke et al., 2007; Ropke and Cordeau, 2009; Herbawi and Weber, 2011c, 2012a,b, 2011b,a, 2012c). It is a generalization of the pickup and delivery problem with time windows or of the dial-a-ride problem (if one also considers passenger convenience). The reduction is very straightforward: in the ride-matching problem, assume that drivers have no time constraint or (alternatively) have homogenous or heterogeneous working times. Further, assume that drivers do not have individual origins and a destination, but have one

This chapter is based on Stiglic and Gradisar (2014).

common origin and destination depot or that there are only a few of such depots. Also assume that the drivers' vehicle capacities are homogenous or, alternatively, that there are few different vehicle types in the fleet, each with a specific capacity respectively. Then, such a special case is in fact a pickup and delivery problem with time windows.

5.2 Mathematical model

We adapt the dial-a-ride problem (DARP) model formulation of Cordeau (2006). Let n denote the number of ride requests and m the number of drive offers. The problem may be defined on a complete directed graph $G = (N,A)$, where $N = P \cup D \cup O \cup T$, $P = \{1,...,n\}$, $O = \{n+1,...,2n\}$,$D = \{n+m+1,...,2n+m\}$, $T = \{2n+m+1,...,2n+2m\}$. Subsets P and D represent the pickup and drop-off nodes of the riders, while subsets O and T represent the origin nodes and the terminal nodes of the drivers. Each ride request i is associated with an origin node i and a destination node $i+n+m$. Similarly, each drive offer is associated with an origin node i and a destination node $i+n+m$. Flexible ride requests have origins in P and destinations in D like normal ride requests. Let K represent the set of pseudovehicles. Each ride request and each ride offer corresponds to exactly one pseudovehicle. Each pseudovehicle $k \in K$ has a capacity of Q_k and a cost per km of c_k. Capacity is set to 1 for pseudovehicles corresponding to riders and to the number of seats in a participant's vehicle for pseudovehicles corresponding to drivers and flexible riders. Each pseudovehicle also has a cost per km c_k. In the case of pseudovehicles corresponding to riders, the cost represents the cost the rider will incur if he is not assigned to any driver (e.g. the cost of using public transport). By modelling the problem in this way, we assure that the costs of unassigned riders are taken into account in the objective function, thus implicitly taking care of minimizing the number of riders left without a ride.

Each node $i \in N$ is associated with a load change q_i and a non-negative service duration d_i such that $q_i = -q_{i+n+m}$ $(i = 1,...,n+m)$. A time window $[e_i, l_i]$ is associated with each node $i \in N$, where e_i and l_i represent the earliest and latest time, respectively, at which the service may begin at node i. Each arc $(i, j) \in A$ is associated with a distance d_{ij} and a travel time t_{ij}.
For each arc $(i, j) \in A$ and each pseudovehicle $k \in K$, let $x^k_{i,j} = 1$ if vehicle k travels from node i to node j and 0 otherwise. For each node $i \in N$ and each vehicle $k \in K$, let B^k_i be the time at which the vehicle k begins service (i.e. visits) at node i, and Q^k_i be the load of vehicle k after visiting node i. Finally, for each participant i, let L^k_i be the ride time of participant i in vehicle k and L_i the maximum acceptable ride time for that participant. We formulate the problem as follows:

$$\min \sum_{i\in N}\sum_{j\in N}\sum_{k\in K} x^k_{i,j} d_{i,j} c_k \tag{5.1}$$

Subject to

$$\sum_{k\in K}\sum_{j\in N} x^k_{i,j} = 1 \quad \forall i \in P\cup O, \tag{5.2}$$

$$\sum_{k\in K}\sum_{i\in N} x^k_{i,j} = 1 \quad \forall j \in D\cup T, \tag{5.3}$$

$$\sum_{k\in K}\sum_{i\in N} x^k_{i,j} \leq 1 \quad \forall j \in N, \tag{5.4}$$

$$\sum_{j\in N} (x^k_{j,i} - x^k_{i,j}) = 0 \quad \forall i \in P\cup O,\ \forall k \in K,\ k \neq i, \tag{5.5}$$

$$\sum_{j\in N} (x^k_{j,i+n+m} - x^k_{i+n+m,j}) = 0 \quad \forall i \in P\cup O,\ \forall k \in K,\ k \neq i, \tag{5.6}$$

$$\sum_{\substack{j\in N\\ j\neq i+n+m\\ j\neq i}} (x^k_{i,j} - x^k_{j,i+n+m}) = 0 \quad \forall i \in P\cup O,\ \forall k \in K,\ k \neq i, \tag{5.7}$$

$$\sum_{j\in P} x^i_{i,j} + x^i_{i,i+n+m} = 1 \quad \forall i \in O, \tag{5.8}$$

$$x^k_{k,k+n+m} + x^k_{i,j} \leq 1\ \forall i \in N,\ \forall j \in N,\ \forall k \in K, k \neq i, \tag{5.9}$$

$$x^i_{i+n+m,j} = 0 \quad \forall i \in P\cup O,\ \forall j \in N, \tag{5.10}$$

$$x^i_{j,i} = 0 \quad \forall i \in P\cup O,\ \forall j \in N, \tag{5.11}$$

$$B^k_j \geq (B^k_i + d_i + t_{i,j}) x^k_{i,j} \quad \forall i \in N,\ \forall j \in N, \forall k \in K, \tag{5.12}$$

$$L_i + B^k_i + d_i - B^k_{i+n+m} = 0 \quad \forall i \in N,\ \forall i \in P\cup O, \forall k \in K, \tag{5.13}$$

$$Q^k_j \geq (Q^k_i + q_i) x^k_{i,j} \quad \forall i \in N,\ \forall j \in N, \forall k \in K, \tag{5.14}$$

$$e_i \leq B^k_i \leq l_i \quad \forall i \in N,\ \forall k \in K, \tag{5.15}$$

$$t_{i,i+n+m} \leq L^k_i \leq L_i \quad \forall i \in P\cup O,\ \forall k \in K, \tag{5.16}$$

$$x^k_{i,j} \in \{0,1\} \quad \forall i \in N, \forall j \in N,\ \forall k \in K, \tag{5.17}$$

$$B^k_i \in \mathbb{R}^+_0 \quad \forall i \in N,\ \forall k \in K, \tag{5.18}$$

$$L^k_i \in \mathbb{R}^+_0 \quad \forall i \in N,\ \forall k \in K, \tag{5.19}$$

$$Q^k_i \in \mathbb{Z}^+_0 \quad \forall i \in N,\ \forall k \in K, \tag{5.20}$$

The objective function (1) minimizes the total costs of all the trips including the costs incurred by unassigned riders. Constraints (2)–(4) are global constraints that govern the assignment of participants to trips. Constraints (2) and (3) assure that each participant leaves from his origin node and arrives at his destination node exactly once. Constraints (3) assure that a pseudovehicle arrives at a particular node not more than once. Constraints (5) to (7) are flow conservation constraints that enforce the consistency of the routes. Constraints (5) and (6) enforce that the vehicle which enters a particular node is also the one the leaves that node. Constraints (7) ensure that the same vehicle that picks up a particular rider is also the one that drops him at his destination. Driver origin and destination nodes are exceptions in all flow conservation constraints. Constraint (8) enforces that a driver's vehicle will definitely be used in the solution, while constraint (9) ensures that a pseudovehicle that is going straight from its origin to destination cannot interfere with other trips. Constraints (10) make sure that a driver's vehicle does not arrive at its own origin node. Finally, constraints (12)–(16) are well-known DARP constraints assuring the consistency of the load and service time variables. Note that (12) and (14) are in fact nonlinear constraints and need to be linearized – we use the approach suggested by Cordeau (2006).

5.3 Preprocessing phase

We implemented a preprocessing phase to speed up the runtime of the solver. In the preprocessing phase, we analyzed each possible pairing of two participants i and j, where $i \neq j$, by solving $(n+m)^2-(n+m)$ a small linear program in the following form:

$$\min t_i \tag{5.21}$$

Subject to

$$t_j - t_i \geq t_{i,j} + d_i \tag{5.22}$$

$$t_{j+N} - t_j \geq t_{j,j+N} + d_j \tag{5.23}$$

$$t_{j+N} - t_{i+N} \geq t_{j+N,i+N} + d_{j+N} \tag{5.24}$$

$$t_i \geq e_i \tag{5.25}$$

$$t_j \geq e_j \tag{5.26}$$

$$t_{i+n+m} \leq l_{i+n+m} \tag{5.27}$$

$$t_{j+n+m} \leq l_{j+n+m} \tag{5.28}$$

$$t_{j+n+m} - t_j \leq L_j \tag{5.29}$$

$$t_{i+n+m} - t_i \leq L_i \tag{5.30}$$

$$t_k \in \mathbb{R}_0^+,\ k \in \{i, j, i+n+m, j+n+m\} \tag{5.31}$$

By solving each linear program, we find whether the pairing of rider j with driver i is feasible considering all the time windows and ride time constraints. If such a pairing is infeasible, we fix all the corresponding $x^i_{j,s}$ and $x^i_{s,j+n+m}$, where $\forall s \in N$ variables to zero.

5.4 Testing

We implemented the model using the Python CPLEX 12.6 application programming interface and ran experiments on an Intel i5 quad core computer with 4 GB of memory. We have tested the model on a number of randomly generated instances to determine whether the model is correctly implemented. After this step, we constructed several instances that model a more realistic problem of matching riders and drivers traveling between Slovenia's two biggest cities. In the instances, 20 participants need to drive from Maribor to Ljubljana. We calculated a travel time matrix using Google Maps tools.

We experiment how the clustering of time windows (if the time windows of all participants are less or more aligned), the participants' maximum ride time and the number of flexible participants influence the number of matches, total costs, and the complexity of the problem reflected in the time the solver spent solving the instance. We used a cost of .25 EUR per vehicle kilometer for all participants. The value of the objective function hence reflects the total costs associated with the planned itineraries. We report the results in the table below.

We found that preprocessing significantly reduces solution times, e.g. for instance 8, we reduced the time CPLEX spent on the main model from 1,546 to 12 seconds. Still, we could not solve instances 6, 10, 14 within the limit of 1000 s.

5.5 Results

What we observe from the results in the table below is that simply by varying the clustering of the time windows, the maximum ride times and the number of users that are flexible, one may significantly increase or decrease the complexity of the problem. It can clearly be seen that the homogeneity of time windows, the ride time flexibility, and the number of flexible riders (0, 4, 12, 20) all have a very positive effect on the number of matches found, the value of the objective function, but also can have drastic effects on the solution times. The star, for instance 12, denotes there are two double matches in the solution. All other instances only have single driver–single rider matches.

5.6 Conclusion

In this chapter, we introduced a new ride-matching model for ridesharing systems that is capable of matching individual drivers with multiple riders. We also devised a preprocessing phase and demonstrated its usefulness. The tests we performed clearly demonstrate that simply by varying the clustering of the time windows, the maximum ride times and the number of users that are flexible, one may significantly increase or decrease the complexity of a single instance of the problem. It is also quite clear that larger, real-life instances of this problem would require a more sophisticated solution approach. This is an obvious possibility for future research.

Table 5.1: Results of single driver–multi rider experiments with MIP model

Number of flexible users	Time windows	Departure time flexibility	Instance ID	Optimal value	Time needed (s)	Number of matches
0	Homogenous	10 min	1	581.875	0.15	2
		20 min	2	334.125	456.23	10
	Scattered	10 min	3	581.875	0.12	2
		20 min	4	395.625	1.07	8
4	Homogenous	10 min	5	549.125	0.15	3
		20 min	6	N/A	N/A	N/A
	Scattered	10 min	7	549.125	0.15	3
		20 min	8	363.125	12.01	9
12	Homogenous	10 min	9	487.25	0.15	4
		20 min	10	N/A	N/A	N/A
	Scattered	10 min	11	517.25	0.15	4
		20 min	12	301	947.63	10*
20	Homogenous	10 min	13	394.75	290.97	8
		20 min	14	N/A	N/A	N/A
	Scattered	10 min	15	426.125	2.42	7
		20 min	16	N/A	N/A	N/A

CONCLUSION

Our aim in this thesis was to gain a better understanding of how ridesharing systems function and to investigate several strategies and possible design choices for ridesharing system providers.

In Chapter 1, we identified and defined three different types of participant flexibility relevant in the dynamic ridesharing context. We quantified the impact of these types of flexibility on system performance by conducting an extensive computational study. Finally, we investigated the level of additional flexibility that is required to improve the effectiveness of a rideshare system.

In Chapter 2, we designed and implemented an algorithm that optimally matches drivers and riders in large-scale ridesharing systems. We performed an extensive simulation study in order to understand how meeting points affect the number of matched participants as well as the system-wide driving distance savings. We concluded by performing sensitivity analysis for a wide range of potential factors to understand the robustness of the effects of introducing meeting points into a ridesharing system.

In Chapter 3, we designed and implemented an algorithm to optimally create single or multi-modal rideshare matches. We then conducted an extensive simulation study to quantify the benefits of integrating ridesharing and public transit. Like in the other two chapters, we also performed sensitivity analysis for a wide range of potential factors to understand the robustness of the observed effects.

In Chapter 4, we studied how to identify feasible driver–rider matches more efficiently. We explored two ideas: (1) direct drive times from origin to destination can be used to efficiently identify those riders who have sufficiently small drive times to be matched with a particular driver; and (2) rider time windows can be stored in a memory structure that allows one to find riders with time windows that overlap with the time window of a driver in sub-linear time. We developed and tested a data structure that combines optimizations (1) and (2) and tested its performance.

In Chapter 5, we presented a new mixed-integer linear programming model for the single driver–multiple rider matching problem that arises in certain types of ridesharing systems. The model we devised allows users to opt for driver or rider roles or, alternatively, to let the model determine what is best. The model minimizes the cost of all the trips that have to be performed to move the users from their origin to their destination nodes. We developed a preprocessing procedure and tested its usefulness. As a final step, we performed simulations on different instances constructed according to the ridesharing practice between Slovenia's two largest cities.

Discussion of Practical and Theoretical Contributions

The main contributions of Chapter 1 are that we introduced and defined three different types of participant flexibility relevant in the dynamic ridesharing context, i.e. matching flexibility, detour flexibility, and scheduling flexibility; we quantified the impact of these flexibility types on system performance by conducting an extensive computational study; and we investigated the level of additional flexibility required to improve a rideshare system's effectiveness. While previous studies have looked at some aspects of participant flexibility in isolation, this is the first study to explicitly and extensively investigate the interaction between system density, level of flexibility, and type of flexibility.

The primary contributions of Chapter 2 are that we introduced a new and relevant problem that considers the use of meeting points in ridesharing systems, we discussed the design and implementation of an algorithm that optimally matches drivers and riders (based on an extension of the traditional bipartite matching formulation) in large-scale ridesharing systems with meeting points, and we performed an extensive simulation study (based on real-world traffic patterns) to assess the benefits of meeting points.

The biggest contributions of Chapter 3 are that we introduced a new and relevant problem that considers the integration and synchronization of ridesharing and scheduled public transit, presented a solution approach to optimally create single or multi-modal rideshare matches, and conducted an extensive numerical study on artificial instances that capture the main characteristics of many real-world transit settings and quantify the benefits of integrating ridesharing and public transport.

The main contributions of Chapter 4 are that we presented several alternative data structures designed to find and return all riders with a travel time that is compatible with the travel time of a given driver. We showed that the most advanced method we developed can reduce the duration of the enumeration phase by more than 50% compared to an exhaustive enumeration. The key contributions of Chapter 5 are that we introduced a new and relevant optimization problem for the single driver–multiple rider matching problem that emerges in certain types of ridesharing systems and presented a special preprocessing procedure for this specific problem.

Summary of the most important insights

We would first like to highlight two important insights of this thesis as a whole:

- **Methodological insight:** In Chapter 2, we observed that a match between a driver i and a set of riders $J \subseteq R$ with $|J| \geq 2$ is time-feasible if the match between driver i and subset of riders $J' \subseteq J$ is time-feasible for all $J' \subseteq J$. Hence, for a match of one driver and two riders to be time-feasible, the match of the driver with each of these two riders must also be time-feasible. Similarly, for a match of one driver and three riders to be time-feasible, the match of the driver with each of the possible rider pairs must be time-feasible as well, and so forth. We believe this is an important property that can help efficiently solve a wide variety of ride-matching problems. We used it in three different implementations and it has also been applied by Arslan et al. (2016) to solve a related matching problem.
- **Operational insight:** All of our results consistently show that if rideshare participants are not motivated and/or willing to be flexible and if the system operates in isolation, then it is very challenging to guarantee a high matching rate for participants (e.g. $\geq 75\%$). Our results also show that these problems can potentially be mitigated by motivating drivers to make longer detours and by motivating riders to

accept multi-modal matches in which part of the trip may be performed by riding on public transport, riding on a bike, or walking.

The most important insights for each chapter are summarized as follows:

- The results presented in Chapter 1 clearly demonstrate (and quantify) the impact of participant flexibility on the performance of a single driver–single rider ridesharing system (in terms of the matching rate achieved). The study shows that participant flexibility plays a key role in easing the matching process, especially in systems with low participation rates. In order for dynamic ridesharing to work, drivers and riders need to be flexible in terms of their departure and arrival times (at least 10 to 15 minutes depending on the locations of origin and destination) but, most importantly, drivers need to be flexible in terms of the detour they are willing to make. Another key finding is that when the number of trip announcements in the system is small, participants must be flexible in their departure times to find a match, the extent to which drivers are willing to make detours is critical to the success of a ridesharing system, and the flexibility required to be matched can vary significantly for system participants.

- In Chapter 2, we showed that introducing meeting points into a ridesharing system can substantially improve several critical performance metrics, i.e. percentage of matched riders, percentage of matched participants, and mileage savings. The price to be paid to achieve these performance increases is minor: riders may have to walk a short distance and may have to plan their time more carefully so as to ensure they arrive on time at the meeting point where they are to be picked up (it is unlikely that drivers will be willing to wait for a rider at a pickup point for more than a minute or two). The observed increases in performance of a ridesharing system resulting from introducing meeting points may even be greater when the meeting points are chosen carefully based on observed travel patterns. Even though the number of possible matches increases significantly when meeting points are introduced, our computational experiments demonstrated that all feasible matches can be generated efficiently with a carefully designed and implemented algorithm.

- The results presented in Chapter 3 show the integration of a ridesharing system and a public transit system can significantly enhance mobility, increase the use of public transport, and reduce the negative externalities associated with car travel. We found that driver willingness to pick up and drop off more than one rider is critical to the system performance.

- In Chapter 4, we demonstrated that it is possible to significantly speed up the generation of feasible rideshare matches by exploiting the idea that direct drive times from origin to destination can be used to efficiently identify those riders who have sufficiently small drive times to be matched with a particular driver.

- Finally, in Chapter 5 we introduced a new model and a preprocessing procedure.

We hope the insights this thesis generates will valuably contribute to the current body of knowledge on ridesharing systems and their operation and they will inform ridesharing system providers on how to design applications, matching algorithms and incentive schemes as well as form alliances with other systems such as public transport agencies and bike-sharing system providers.

Opportunities for Future Research

Ridesharing as a research topic is receiving more and more attention. While the number and variety of studies on the design and operation of these systems is growing, there are still important opportunities for future research.

A clear opportunity for future research is the analysis of complex ridesharing systems integrated with large metropolitan public transport systems, bike-sharing systems and/or other modes of scheduled or on-demand transportation.

A different research direction could entail the design and testing of cost-sharing and incentive schemes to motivate drivers and riders to share rides. One interesting option is the blending of genuine ridesharing in which a driver performs only a small detour to pick up a rider and only expects the rider to remunerate a certain part of the trip costs with the ridesharing model popularized by Lyft and Uber (in which drivers act as taxi drivers). It seems that a combination of these two systems could ensure a high level of service at an affordable cost while also reducing congestion-related problems in urban areas.

A topic that still needs to be understood better is the influence of different commitment strategies on the performance of a ridesharing system in a dynamic system. A more empirical study could also focus on user preferences regarding commitment strategies and matching rules as such.

Finally, in most of this thesis we investigated only some of the most convenient and practicable types of matches. The main reason for this is that in genuine ridesharing drivers share rides on trips that are already planned. In such a setting, drivers only accept rideshare tasks that are convenient for them. Clearly, there are settings like long-distance ridesharing in which more complex matches can be interesting (see Chapter 5). One potential research topic in long-distance ridesharing is certainly multi-hop ridesharing in which a rider can transfer between several vehicles before arriving at his final destination. Another possible extension is the integration with other modes of scheduled or on-demand transport. Riders may also want to book a return trip back home, making the ride-matching problem even more complex and hard to solve.

REFERENCES

Agatz, N. A. H., Erera, A., Savelsbergh, M. P. W., & Wang, X. (2011). Dynamic ride-sharing: A simulation study in metro Atlanta. *Transportation Research Part B*, *45*(9), 1450–1464.

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

Aissat, K. & Oulamara, A. (2014). A priori approach of real-time ridesharing problem with intermediate meeting locations. *Journal of Artificial Intelligence and Soft Computing Research*, *4*(4), 287–299.

Amey, A. (2011). Proposed methodology for estimating rideshare viability within an organization: Application to the MIT community. *Transportation Research Board Annual Meeting 2011*, Paper 11-2585.

Arslan, A., Agatz, N., Kroon, L. G., & Zuidwijk, R. A. (2016). Crowdsourced delivery: A dynamic pickup and delivery problem with ad-hoc drivers. *Technical report*, Erasmus University, Rotterdam, The Netherlands.

Baldacci, R., Maniezzo, V., & Mingozzi, A. (2004). An exact method for the car pooling problem based on Lagrangean column generation. *Operations Research*, *52*(3), 422–439.

Bentley, J. L. (1990). K-d trees for semidynamic point sets. In *Proceedings of the Sixth Annual Symposium on Computational Geometry, SCG '90*, (pp. 187–197), New York, USA.

Bertrand, W. M. J. & Fransoo, J. C. (2002). Operations management research methodologies using quantitative modeling. *International Journal of Operations & Production Management*, *22*(2), 241–264.

Bruglieri, M., Ciccarelli, D., Colorni, A., & Luè, A. (2011). Poliunipool: A carpooling system for universities. *Procedia - Social and Behavioral Sciences*, 20, 558–567.

Calvo, R. W., de Luigi, F., Haastrup, P., & Maniezzo, V. (2004). A distributed geographic information system for the daily car pooling problem. *Computers and Operations Research*, *31*(13), 2263–2278.

Cayford, R. & Yim, Y. B. Y. (2004). Personalized demand-responsive transit service. *Technical report*, University of California, Berkley.

Coltin, B. J. & Veloso, M. (2013). Towards ridesharing with passenger transfers. In *Proceedings of the 2013 International Conference on Autonomous agents and Multi-agent Systems* (pp. 1299–1300). International Foundation for Autonomous Agents and Multiagent Systems.

Cordeau, J. F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, *54*(3), 573–586.

Cordeau, J. F. & Laporte, G. (2003). The dial-a-ride problem (DARP): Variants, modeling issues and algorithms. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, *1*(2).

Cordeau, J. F. & Laporte, G. (2007). The dial-a-ride problem: Models and algorithms. *Annals of Operations Research*, *153*(1), 29–46.

Drews, F. & Luxen, D. (2013). Multi-hop ride sharing. In *Proceedings of the Sixth Annual Symposium on Combinatorial Search* (pp. 71–79). AAAI Press.

Drexl, M. (2012). Synchronization in vehicle routing–A survey of VRPs with multiple synchronization constraints. *Transportation Science*, *46*(3), 297–316.

EEA (2010). Occupancy rates of passenger vehicles. *Technical report*, European Environment Agency.

Epperson, K. (2015). Santa Barbara dynamic ridesharing pilot program. *Technical report*, SBCAG Traffic Solutions, Santa Barbara, CA, USA.

Furuhata, M., Dessouky, M., Ordonez, F., Brunet, M., Wang, X., & Koenig, S. (2013). Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, *57*(0), 28 – 46.

Geisberg, R., Luxen, D., Neubauer, S., Sanders, P., & Volker, L. (2010). Fast detour computation for ride sharing. In *ATMOS 2010 - 10th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems* (pp. 88–99). Leibniz-Zentrum fuer Informatik.

Goel, A. & Meisel, F. (2013). Workforce routing and scheduling for electricity network maintenance with downtime minimization. *European Journal of Operational Research*, *231*(1), 210–228.

Herbawi, W. & Weber, M. (2011a). Ant colony vs. genetic multiobjective route planning in dynamic multi-hop ridesharing. In *Tools with Artificial Intelligence (ICTAI) 2011* (pp. 282–288).

Herbawi, W. & Weber, M. (2011b). Comparison of multiobjective evolutionary algorithms for solving the multiobjective route planning in dynamic multi-hop ridesharing. In *Evolutionary Computation* (CEC)2011 (pp. 2099–2106.)

Herbawi, W. & Weber, M. (2011c). Evolutionary multiobjective route planning in dynamic multi-hop ridesharing. In Merz, P. & Hao, J.-K. (Eds.), *Evolutionary Computation in Combinatorial Optimization* (pp. 84–95). Springer.

Herbawi, W. & Weber, M. (2012a). A genetic local search algorithm for multiobjective time-dependent route planning. In *IEEE Congress on Evolutionary Computation* (pp. 1-7). Brisbane, Queensland.

Herbawi, W. & Weber, M. (2012b). The ridematching problem with time windows in dynamic ridesharing: A model and a genetic algorithm. In *Evolutionary Computation* (CEC) 2012 (pp. 1–8).

Herbawi, W. & Weber, M. (2012c). *Solving the ridematching problem in dynamic ridesharing* (doctoral thesis), Ulm University, Ulm.

Hosni, H., Naoum-Sawaya, J., & Artail, H. (2014). The shared-taxi problem: Formulation and solution methods. *Transportation Research Part B: Methodological*, 70, 303–318.

Ignizio, J. P. (1976). *Goal programming and extensions*. Lexington Books, Lexington, MA.

Jacobson, S. H. & King, D. M. (2009). Fuel saving and ridesharing in the US: Motivations, limitations, and opportunities. *Transportation Research Part D: Transport and Environment*, *14*(1), 14–21.

Kaan, L. & Olinick, E. V. (2013). The vanpool assignment problem: Optimization models and solution algorithms. *Computers and Industrial Engineering*, *66*(1), 24–40.

Kamar, E. & Horvitz, E. (2009a). Collaboration and shared plans in the open world: Studies of ridesharing. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence, IJCAI '09*, (pp. 187–194). Palo Alto, California, USA. AAAI.

Kamar, E. & Horvitz, E. (2009b). Generating transportation plans under varying preferences: Ridesharing models and mechanisms. *Technical report*, Microsoft Research.

Kleiner, A., Nebel, B., & Ziparo, V. (2011). A mechanism for dynamic ride sharing based on parallel auctions. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, IJCAI '11,* (pp. 266–272). Palo Alto, California, USA. AAAI.

Koffman, D. (2004). Operational experiences with flexible transit services. *Technical report*, Transportation Research Board.

LaPlante, J. & Kaeser, T. (2004). The continuing evolution of pedestrian walking speed assumptions. *ITE Journal*, *74*(9), 32–40.

Lee, A. & Savelsbergh, M. (2017). An extended demand responsive connector. *EURO Journal on Transportation and Logistics*, *6*(1), 25 – 50.

Lee, A. & Savelsbergh, M. (2015). Dynamic ridesharing: Is there a role for dedicated drivers? *Transportation Research Part B: Methodological*, 81, 483 – 497.

Li, X. & Quadrifoglio, L. (2010). Feeder transit services: Choosing between fixed and demand responsive policy. *Transportation Research Part C: Emerging Technologies*, *18*(5), 770 – 780.

Li, X. & Quadrifoglio, L. (2011). 2-vehicle zone optimal design for feeder transit services. *Public Transport*, *3*(1), 89–104.

McKenzie, B. (2010). Public transportation usage among U.S. workers: 2008 and 2009. *Technical report*, U.S. Census Bureau.

Meisel, F. & Kopfer, H. (2014). Synchronized routing of active and passive means of transport. *OR Spectrum*, *36*(2), 297–322.

Murphy, C. (2016). Shared mobility and the transformation of public transit. *Technical report*, Transportation Research Board.

Murray, G., Chase, M., Kim, E., & McBrayer, M. (2012). Ridesharing as a complement to transit: A synthesis of transit practice. *Technical report*, Transportation Research Board.

Nguyen, D. T. (2013). *Fair cost sharing auction mechanisms in last mile ridesharing* (doctoral thesis), Singapore Management University, Singapore.

Park, J. & Kim, B. (2010). The school bus routing problem: A review. *European Journal of Operational Research*, *202*(2), 311–319.

Polzin, S. & Pisarski, A. (2013). Commuting in America 2013. *Technical report*, American Association of State Highway and Transportation Officials.

Quadrifoglio, L. & Xiugang, L. (2009). A methodology to derive the critical demand density for designing and operating feeder transit services. *Transportation Research Part B: Methodological*, *43*(10), 922 – 935.

Riera-Ledesma, J. & Salazar-Gonzalez, J. J. (2013). A column generation approach for a school bus routing problem with resource constraints. *Computers and Operations Research*, *40*(2), 566–583.

Ropke, S. & Cordeau, J.-F. (2009). Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, *43*(3), 267–286.

Ropke, S., Cordeau, J.-F., & Laporte, G. (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, *49*(4), 258–272.

Schittekat, P., Kinable, J., Sorensen, K., Sevaux, M., Spieksma, F., & Springael, J. (2013). A metaheuristic for the school bus routing problem with bus stop selection. *European Journal of Operational Research*, *229*(2), 518–528.

Schrank, D., Eisele, B., & Lomax, T. (2012). TTI's 2012 urban mobility report. *Technical report*, Texas A&M Transportation Institute.

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

Stiglic, M., Agatz, N., Savelsbergh, M., & Gradisar, M. (2015b). On the efficient generation of feasible driver rider matches in ride-sharing systems. In Kljajic, M. & Lasker, G. E. (Eds.), *Advances in simulation-based decision support & business intelligence, volume V*, 27th International Conference on Systems Research, Informatics and Cybernetics, Baden-Baden, Germany.

Stiglic, M., Agatz, N., Savelsbergh, M., & Gradisar, M. (2016a). Enhancing urban mobility: Integrating ride-sharing and public transit. *Technical report*, Erasmus Research Institute of Management, Rotterdam.

Stiglic, M., Agatz, N., Savelsbergh, M., & Gradisar, M. (2016b). Making dynamic ride-sharing work: The impact of driver and rider flexibility. *Transportation Research Part E: Logistics and Transportation Review*, 91, 190 – 207.

Stiglic, M. & Gradisar, M. (2014). Mixed-integer linear programming model for the single driver multiple rider matching problem in ridesharing systems. In Kljajic, M. & Lasker, G. E. (Eds.), *Advances in simulation-based decision support & business intelligence, Volume IV*, 26th Conference on System Research, Informatics and Cybernetics, Baden-Baden, Germany.

Teodorovic, D. & Dell' Orco, M. (2008). Mitigating traffic congestion: Solving the ride-matching problem by bee colony optimization. *Transportation Planning and Technology*, *31*(2), 135–152.

Wang, X., Agatz, N. A. H., & Erera, A. (2014). Stable matching for dynamic ride-sharing systems. *Technical report*, Erasmus University Rotterdam, Rotterdam, The Netherlands.

Wang, X., Dessouky, M., & Ordonez, F. (2016). A pickup and delivery problem for ridesharing considering congestion. *Transportation Letters*, *8*(5), 259–269.

Xu, H., Ordóñez, F., & Dessouky, M. (2015a). A traffic assignment model for a ridesharing transportation market. *Journal of Advanced Transportation*, *49*(7), 793–816.

Xu, H., Pang, J.-S., Ordóñez, F., & Dessouky, M. (2015b). Complementarity models for traffic equilibrium with ridesharing. *Transportation Research Part B: Methodological*, 81, 161–182.

Yan, S. & Chen, C.-Y. (2011a). A model and a solution algorithm for the car pooling problem with pre-matching information. *Computers and Industrial Engineering*, *61*(3), 512–524.

Yan, S. & Chen, C.-Y. (2011b). An optimization model and a solution algorithm for the many-to-many car pooling problem. *Annals of Operations Research*, *191*(1), 37–71.

APPENDIX

Summary in Slovenian language / Daljši povzetek v slovenskem jeziku

Uvod

Deljenje prevoza (ang. *ride-sharing*) je družbeni pojav, pri katerem si posamezniki z ujemajočimi se prevoznimi potrebami delijo eno vozilo in celotno pot ali nekatere dele svojih poti opravijo skupaj. Voznik vozila ima določeno pot ob določenem času in je v zameno za povračilo stroškov prevoza pripravljen po poti pobirati ter odlagati sopotnike. Pot voznika je lahko fiksna ali pa se lahko prilagaja posameznim lokacijam vstopa in izstopa potnikov – obstaja veliko možnosti oz. variacij. Razen povračila stroškov prevoza vozniku, ki sicer močno prevladuje, so možni tudi drugi dogovori, kot npr. da vsak udeleženec pelje enkrat na teden (tipično v t. i. Carpoolingu).

Koristi deljenja prevoza za posameznika so nižji stroški transporta, možnost uporabe pasov za vozila z več potniki (tiṗasovi HOV v ZDA) in manjša utrujenost sopotnikov. V primerih, ko sopotniki nimajo osebnega vozila, lahko govorimo tudi o skrajšanem času potovanja v primerjavi z javnim transportom.

Z deljenjem prevozov se lahko pomembno zmanjša število osebnih vozil, ki so potrebna za zadovoljitev potreb posameznikov po mobilnosti. Zlasti v regijah, v katerih mobilnost posameznikov temelji na uporabi osebnih vozil, je z deljenjem prevozov mogoče znižati število vozil v prometu in s tem razbremeniti prometno omrežje. Deljenje prevozov ima lahko torej pozitivne učinke na promet in okolje v smislu zmanjševanja pojavnosti gneče ter zastojev v prometu. Pravtako lahko vpliva na zmanjševanje onesnaženosti, zmanjševanje porabe goriv oz. energije ter posledično primarnih virov.

S pojavom interneta in mobilnih tehnologij so se pojavile raznovrstne, napredne oblike deljenja prevoza. Danes obstaja širok spekter različnih sistemov: od preprostejših e-oglasnih desk za deljenje prevoza do bolj kompleksnih storitev, do katerih se lahko dostopa preko spletnih in mobilnih aplikacij in ki ponujajo avtomatsko tvorjenje skupin, načrtovanje poti ter elektronska plačila. V tej disertaciji se osredotočamo na napredne sisteme, ki omogočajo deljenje prevozov preko pametnih telefonov in drugih sodobnih naprav ter avtomatsko povezujejo voznike in potnike, tj. tvorijo prevozne skupine. Primer takšnega ponudnika je npr. nemško podjetje Flinc (`https://flinc.org/`). V primerih, ko je vodilni čas od najave prevoza v sistemu do dejanske izvedbe kratek (npr. 10-30 minut) in je malo časa za koordinacijo med vozniki ter potniki, govorimo o t. i. dinamičnem deljenju prevozov.

Tako izkušnje iz prakse kot akademske študije kažejo, da je snovanje in upravljanje sistemov za deljenje prevoza zahtevno. Predvidevati je treba, kako se bo sistem obnašal v različnih razmerah, in ves čas spremljati, kaj se dogaja na strani ponudbe prevozov ter kaj na strani povpraševanja po prevozih. To je zlasti pomembno ob vzpostavitvi tovrstnih sistemov. Ob zagonih novih sistemov za deljenje prevoza je število uporabnikov nizko, zaradi česar je pogosto težko najti ujemanja. Kamar in Horwitz (2009) sta pokazala, da je število dopustnih rešitev (ujemanj) za uporabnika močno odvisno od celotnega števila aktivnih uporabnikov. Zato je ob zagonu verjetnost, da se bo neki uporabnik ujemal s katerim drugim, majhna. To se dogaja tudi v uveljavljenih sistemih na določenih relacijah ob določenih urah oz. dnevih. Nezmožnost najti prevoz ali potnike, posebej nekajkrat zaporedoma, lahko uporabnika odvrne od nadaljnje uporabe. Gre torej za resen problem, ki je relevanten tako za sisteme v zagonu kot tudi za uveljavljene sisteme.

Vsak voznik, ki ponuja prevoz v sistemu za deljenje prevoza, ima določeno pot. Da bi našel sopotnike, s katerimi bi si delil stroške prevoza, je pripravljen narediti nekaj dodatnih postankov, potrebnih za vstop v vozilo in izstop iz njega. Pogosto mora voznik napraviti tudi obvoz in se s tem približati izvoru ter cilju posameznega potnika. Največja dolžina obvoza in število dodatnih postankov sta odvisna od voznikove pripravljenosti, da podaljša potovalni čas. To pomembno razlikuje sisteme za deljenje prevoza od drugih sorodnih sistemov, v katerih so vozniki dejansko taksisti in se popolnoma prilagajajo želji stranke. Primer takšnega sistema je npr. Uber (`https://www.uber.com`). Nivo storitve v tovrstnem sistemu je lahko bistveno višji zaradi visoke fleksibilnosti voznika – to se odraža tudi v ceni prevoza, ki je bistveno višja kot v pravem sistemu za deljenje prevoza. Z izjemo taksijev, v katerih si več, med seboj nepovezanih strank deli taksi prevoz, takšne storitve tudi nimajo nujno pozitivnih učinkov na zgoščenost prometa ter okolje.

Omejena pripravljenost voznikov za izvedbo dodatnih postankov in daljših obvozov zelo otežuje deljenje prevozov. Če je voznik nefleksibilen, je namreč izredno težko najti ustreznega sopotnika. Posledica tega je lahko, da velikega deleža voznikov in potnikov, ki bi sicer želeli sodelovati v sistemu, ni mogoče povezati. Da bi tovrstne probleme preprečili ali jih vsaj pomembno omejili, je treba sisteme za deljenje prevoza skrbno načrtovati in uporabiti primerne tehnologije ter algoritme, vendar, kot smo pokazali v disertaciji, to še ni zadosten pogoj za uspešno delovanje.

Ta disertacije je osredotočena na lajšanje procesa iskanja ujemanj v sistemih deljenja prevoza. V ta namen smo zasnovali in preizkusili več različnih modelov, algoritmov ter mehanizmov, ki bi lahko zagotovili visoko verjetnost ujemanja za uporabnike (maksimizacija števila ujemanj). Opravili smo veliko število simulacij, na podlagi katerih je možno sklepati o pomembnih lastnostnih sistemov za deljenje prevoza in tudi o potencialnih učinkih določenih strateških odločitev upravljavca sistema (npr. integracija sistema deljenja prevoza s sistemom javnega transporta ali vključitev nabornih točk v sistem deljenja prevoza).

V sklopu disertacije smo razvili in testirali večje število metod. Te metode so zapisane v obliki matematičnih modelov in v psevdokodi ter implementirane v visokonivojskih programskih jezikih (Python, Cython, C, Optimization Programming Language [OPL]). Pomembna dela razvoja sta bila testiranje in končna validacija. Po tem koraku smo primerjali različne mehanizme in algoritme med seboj s pomočjo simulacij na naključno generiranih podatkih. Ti podatki so bili generirani na podoben način kot v preteklih primerljivih študijah (npr. Agatz et al., 2011; Herbawi and Weber, 2012a; Kleiner et al., 2011). V simulacijah za drugo poglavje smo uporabili tudi podatke, ki so jih že uporabili raziskovalci v preteklih študijah (podatki za Atlanto, ZDA, v Agatz et al., 2011).

V prvem poglavju disertacije proučujemo vpliv fleksibilnosti uporabnikov sistema deljenja prevoza na število ujemanj med vozniki in potniki, ki jih je možno vzpostaviti. Ta vpliv smo proučili tako, da smo zgradili posebno simulacijsko okolje, ki simulira sistem deljenja prevoza, v katerem se en voznik lahko poveže z največ enim potnikom. Definirali smo tri tipe fleksibilnosti, ki igrajo pomembno vlogo pri dinamičnem deljenju prevozov, in skozi veliko število simulacij kvantificirali vpliv posamičnega tipa fleksibilnosti na število možnih ujemanj. Tipi fleksibilnosti, ki smo jih definirali so fleksibilnost prirejanja (ang. *matching flexibility*), fleksibilnost razporejanja (ang. *scheduling flexibility*) in fleksibilnost obvoza (ang. *detour flexibility*). Naredili smo tudi simulacijo, pri kateri smo proučili, koliko dodatne fleksibilnosti (in katere) je potrebno, da se odstotek povezanih uporabnikov poveča za določen odstotek.

V drugem poglavju razvijemo nov model za tvorjenje prevoznih skupin, ki vsebuje naborne točke. Do zdaj so modeli v literaturi predpostavljali, da potniki vstopajo v vozilo in izstopajo iz njega na svojih dejanskih izvorih ter ponorih. Kakorkoli, ta predpostavka je precej omejujoča, saj se mora samo ena stran prilagajati drugi. Z vključitvijo standardnih nabornih mest v sistem je omogočeno, da se potnik iz svojega izvora premakne na najugodnejše naborno mesto oz. da se z določene ugodne naborne točke pomakne proti svojemu ponoru z različnimi transportnimi modalitetami. Takšne naborne točke se uporabljajo v veliko sistemih deljenja prevozov, vendar še niso bile omenjene oz. uporabljene v optimizacijski literaturi. Algoritem, ki smo ga zasnovali za rešitev tega problema, skuša optimirati ujemanja z vidika dveh kriterijev: maksimiranje števila uparjenih uporabnikov in maksimiranje število prihranjenih prevoženih kilometrov v celotnem sistemu.

V tretjem poglavju predstavljamo nov model in algoritem za tvorjenje prevoznih skupin, ki omogoča integracijo s sistemom javnega transporta. Na ta način je omogočeno, da voznik potnika zapelje na njegovo končno destinacijo ali pa ga zapelje na postajo javnega transporta, od koder se z vlakom, avtobusom ali drugim prevoznim sredstvom pelje do svoje končne destinacije. Sistem poskuša sinhronizirati poti in čase odhoda in prihoda voznikov in potnikov z urnikom javnega transporta ter upošteva veliko število omejitev glede preferenc voznikov in potnikov (najhitrejši čas odhoda, najpoznejši čas prihoda, najdaljše trajanje poti ipd.). Podobno kot v prejšnjem poglavju tudi v tem algoritem, ki smo ga zasnovali za rešitev problema, skuša optimirati ujemanja z vidika dveh kriterijev: maksimiranje števila uparjenih uporabnikov in maksimiranje število prihranjenih prevoženih kilometrov v celotnem sistemu.

V četrtem poglavju predstavljamo krajšo metodološko diskusijo na temo generiranja vseh dopustnih ujemanj med vozniki in potniki. Predstavljamo, kako je mogoče izkoristiti lastnosti problema za izboljšanje učinkovitosti metode generiranja vseh dopustnih ujemanj. Informacije o najkrajšem možnem trajanju poti posamičnega potnika in voznika izkoristimo pri grajenju podatkovne strukture, katere namen je omogočiti učinkovito poizvedbo o tem, kateri potniki so potencialno dopustni za dotičnega voznika.

V zadnjem, petem poglavju predstavimo matematični model za tvorjenje prevoznih skupin v sistemih za deljenje prevoza, ki omogoča, da posamičnega voznika povežemo z večjim številom potnikov. Gre za model, ki je primeren za deljenje prevoza med večjimi mesti. Model tudi dopušča, da se uporabnik ne opredeli glede vloge in mu jo sistem sam določi na podlagi razmerja med ponudbo ter povpraševanjem. Predstavimo tudi rutino za predprocesiranje problema. Izvedemo nekaj manjših simulacij.

V nadaljevanju povzemamo glavne ugotovitve prvih treh poglavij.

Vpliv fleksibilnosti voznikov in potnikov na delovanje dinamičnih sistemov deljenja prevoza

O dinamičnem deljenju prevoza govorimo, ko se iščejo ujemanja za posamičnega voznika znotraj zelo kratkega časovnega okna. Z drugimi besedami to pomeni, da je vodilni čas od najave prevoza v sistemu do dejanske izvedbe kratek (npr. 10-30 minut) in je malo časa za koordinacijo med vozniki ter potniki. Takšni sistemi so primerni za deljenje prevozov v urbanih okoljih, kjer je gostota prometa velika in zato obstaja dobra možnost, da bo se za dotičnega voznika ali potnika našlo ujemanje. Lahko gre za poti v službo in nazaj ali pa za pot na prostočasne aktivnosti, kot je npr. nakupovanje.

Dinamično deljenje prevoza uporabniku dopušča možnost, da prevoz ponudi ali poišče v trenutku, ko ve, kdaj točno bi se želel nekam odpraviti. To pomeni, da je uporabnik lahko neodvisen od strogo določenega urnika, kot npr. pri javnem transportu ali navadnem deljenju prevozov, kjer je čas odhoda fiksno določen že vsaj nekaj ur vnaprej. Po drugi strani ravno pomanjkanje fleksibilnosti sistemu povzroča težave pri vzpostavljanju ujemanj, saj je izredno težko ob vsakem trenutku zagotavljati, da bo mogoče najti primernega sopotnika ali voznika.

Jasno je, da je za sistem bolje, da se prevoz najavi bistveno pred najhitrejšim časom odhoda ali da je voznik pripravljen narediti večji obvoz na svoji poti, ali spet da je potnik pripravljen odriniti kadarkoli znotraj širokega časovnega okna. Kar ni jasno, je, kako vsak izmed teh tipov fleksibilnosti oz. prilagodljivosti vpliva na obnašanje sistem, tj. kateri tip fleksibilnost je bolj pomemben, ali je bolj pomembna fleksibilnost potnikov ali voznikov ipd. S temi vprašanji se ukvarja to poglavje.

V tem poglavju proučujemo, kako različni tipi fleksibilnosti, ki so relevantni v kontekstu dinamičnega deljenja prevozov, vplivajo na delovanje sistema kot celote. Ugotavljamo, kako določene vrednosti nekega tipa fleksibilnosti vplivajo na število ujemanj, ki jih je mogoče vzpostaviti. Ugotavljamo, kako povečati število ujemanj v sistemu, s tem da se spremeni fleksibilnost določenega števila uporabnikov, in ne nazadnje opazujemo, kako samo število uporabnikov v sistemu in razmerje med vozniki ter potniki vplivata na obnašanje sistema in na število ujemanj, ki jih je mogoče vzpostaviti.

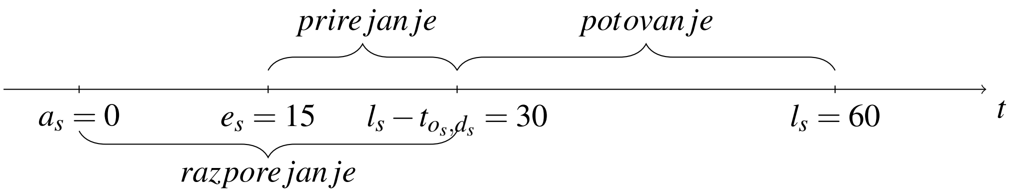
Simuliramo sistem, v katerem je mogoče enega voznika povezati z največ enim potnikom. V študiji definiramo tri tipe fleksibilnosti: fleksibilnost prirejanja (ang. *matching flexibility*), fleksibilnost razporejanja (ang. *scheduling flexibility*) in fleksibilnost obvoza (ang. *detour flexibility*). Fleksibilnost obvoza je relevantna samo za voznike, medtem ko sta fleksibilnost razporejanja in fleksibilnost prirejanja relevantni tako za voznike kot za potnike.

Fleksibilnost prirejanja je pripravljenost uporabnika, da se na pot odpravi hitreje ali pozneje, kot je idealni čas, in sicer z namenom, da bi si povečal možnost, da se zanj najde ujemanje. Fleksibilnost ujemanja je v tej študiji definirana kot razlika med najhitrejšim časom odhoda in najpoznejšim časom prihoda, ki se ji odšteje direktni potovalni čas med izvorom ter ponorom uporabnika. V tej študiji predpostavljamo, da uporabniki fiksno določijo najhitrejši čas odhoda in da večja fleksibilnost prirejanja pomeni, da je uporabnik pripravljen na cilj prispeti pozneje. Omeniti je treba, da ta fleksibilnost (tako kot je definirana) postavlja zgornjo mejo na najdaljši obvoz, ki ga je še pripravljen narediti voznik.

Fleksibilnost razporejanja definira čas, ki je na voljo, da sistem razporedi potnike in voznike.

Za posameznega uporabnika to pomeni, koliko časa je njegova najava v sistemu in na voljo za iskanje najboljšega ujemanja. Fleksibilnost razporejanja je sestavljena iz vodilnega časa od najave v sistem do najhitrejšega časa odhoda.

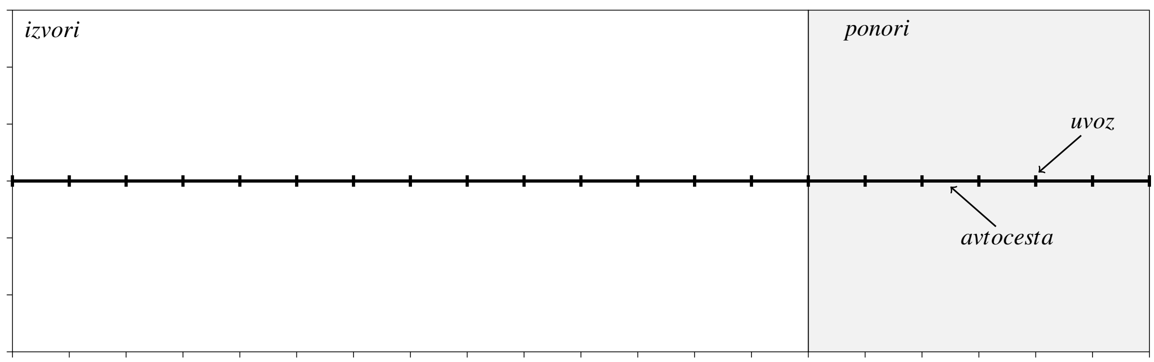
Slika 1 na časovni premici vizualizira zgoraj omenjene tipe fleksibilnosti za najavo. Vodilni čas najave je 15 minut, fleksibilnost razporejanja je 30 minut, fleksibilnost prirejanja pa je 15 minut. Direktni potovalni čas med izvorom in ponorom uporabnika je 30 minut.



Slika 1: Fleksibilnost prirejanja in razporejanja za najavo $s \in S$

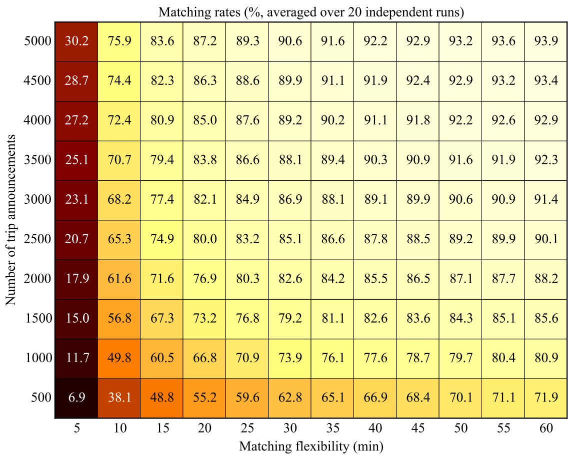
Kot smo že izpostavili, je fleksibilnost obvoza relevantna samo za voznike. Gre za razmerje med potovalnim časom med izvorom in ponorom voznika ter najdaljšim potovalnim časom, ki ga je še pripravljen sprejeti, da bi se prilagodil potniku in si z njim delil stroške. V tej študiji to fleksibilnost definiramo kot funkcijo potovalnega časa voznika. Predpostavljamo, da je fleksibilnost obvoza δ_i voznika $i \in D$ funkcija trajanje njegove direktne poti, tj. $\delta_i = c_f t_{o_i d_i} + \tau$, kjer je c_f parameter fleksibilnosti obvoza, τ pa čas, potreben za izvedbo postanka.

Osnovne eksperimente smo naredili za dve med seboj različni geografski razporeditvi. Prva predstavlja koridor, v katerem se uporabniki iz suburbanega območja vključujejo na avtocesto oz. hitro cesto in potujejo v urbani center. Tak model ima značilnosti dnevnih migracij na delovna mesta iz okolice večjih mest v center mesta. Takšen model je relevanten, ker predstavlja zelo pogost prometni vzorec, ki je pogosto tudi povezan s pomembno zgoščenostjo prometa in z zastoji na vpadnicah v mesta. Slika 2 prikazuje osnovne značilnosti te geografske razporeditve. Druga geografska razporeditev je bolj naključna in predstavlja bolj naključne poti na prostočasne aktivnosti znotraj večjega mesta. Izvori in ponori so generirani popolnoma naključno znotraj večjega kvadrata.

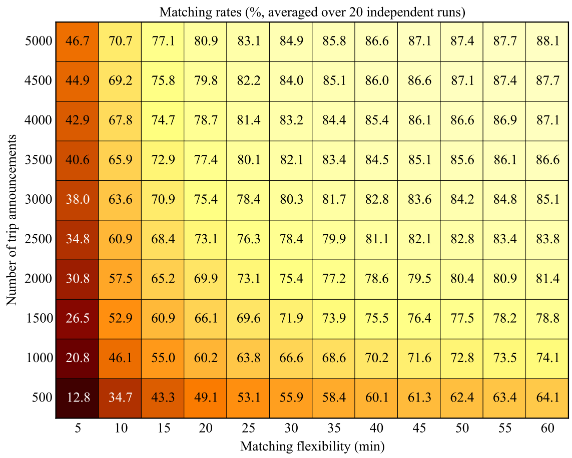


Slika 2: Prikaz geografske razporeditve v koridorju.

Za omenjeni geografski razporeditvi smo naredili večje število simulacij. V prvem tipu simulacij smo proučevali, kako število uporabnikov v sistemu in njihova fleksibilnost (ki je enaka za vse uporabnike) vplivata na število ujemanj oz. stopnjo ujemanja. Stopnja ujemanja je definirana kot odstotek vseh uporabnikov, ki so bili uparjeni. V prvem eksperimentu smo za 20 različnih naključno generiranih primerov izračunali stopnjo ujemanja za različna števila uporabnikov v sistemu (med 500 in 5000 uporabnikov) in fleksibilnostjo prirejanja (med 5 in 60 minut). Slika 3 prikazuje rezultate za prvi eksperiment.



(a) Geografija - koridor.



(b) Geografija - center mesta.

Slika 3: Povprečne stopnje ujemanje (Matching rate) za različne kombinacije sistemske fleksibilnosti prirejanja (System-wide matching flexibility) in števila najav uporabnikov (Number of trip announcements). Rezultati so povprečja 20 neodvisnih ponovitev.

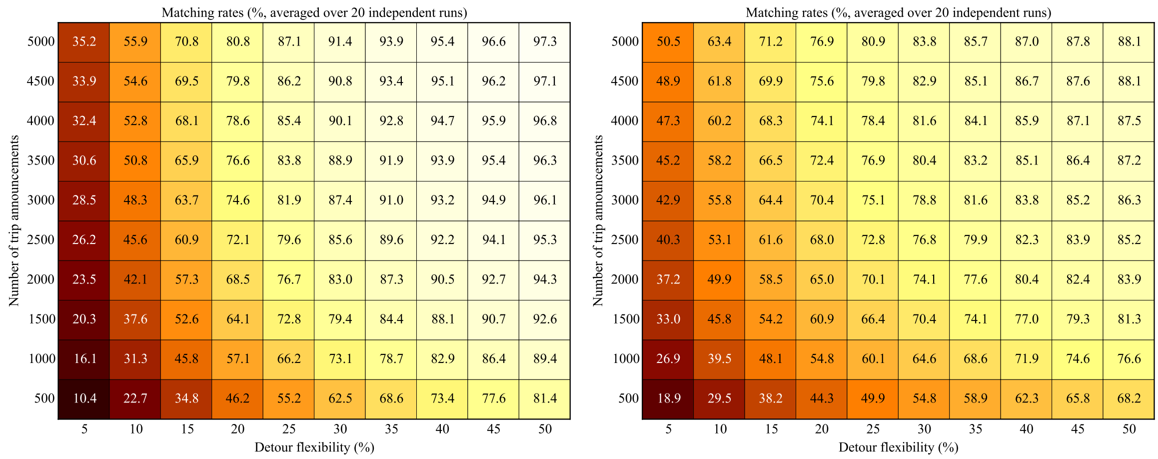
Rezultati kažejo, da je težko zagotoviti visoko stopnjo ujemanja v sistemu deljenja prevoza, če uporabniki niso pripravljeni sprejeti fleksibilnosti prirejanja med 15 in 20 minut. Rezultati tudi kažejo, da je fleksibilnost potrebna zlasti, ko je število uporabnikov majhno. To hkrati pomeni, da je kritična masa, tj. okvirno število uporabnikov, pri katerem sistem postane vzdržen, očitno močno odvisna od samega obnašanja uporabnikov, tj. njihove fleksibilnosti.

Naslednji eksperiment je podoben prvemu: za 20 različnih naključno generiranih primerov smo izračunali stopnjo ujemanja za različna števila uporabnikov v sistemu (med 500 in 5000 uporabnikov) in fleksibilnostjo obvoza (med 5% in 50%). Odstotek fleksibilnosti pomeni odstotek, za katerega je voznik pripravljen podaljšati svojo pot, da bi se prilagodil potniku. Ta fleksibilnost ne zajema časa, ki je potreben za izvedbo dveh postankov – ta čas je fiksiran na 2 minuti –, predpostavljamo, da je voznik vedno pripravljen narediti ta postanek.

Rezultati kažejo, da je fleksibilnost voznikov izjemnega pomena pri vzpostavljanju ujemanj. Na sliki 4 so očitne velike razlike med rezultati za nizke vrednosti fleksibilnosti in majhne vrednosti fleksibilnosti. Očitno je, da je eden od ključev za uspešno delovanje sistema najti primerne spodbude, da bi vozniki bili pripravljeni narediti daljše obvoze.

Zadnji eksperiment v prvi seriji proučuje vpliv fleksibilnosti razporejanja na število ujemanj. V nasprotju s prejšnjima dvema tukaj predstavljamo rezultate samo za primer deljenja prevozov znotraj mest (geografija tipa 2). Razlog je v tem, da so rezultati za koridor praktično popolnoma enaki. Nadalje proučujemo samo dve števili uporabnikov v sistemu, in sicer 500 ter 2000 uporabnikov. Vodilni čas variiramo med 0 in 30 minut, fleksibilnost prirejanja pa tudi med 0 in 30 minut. Izračunamo rezultate za vse možne kombinacije. Naj spomnimo, da smo v tej študiji fleksibilnost razporejanja definirali kot seštevek vodilnega časa in fleksibilnosti prirejanja.

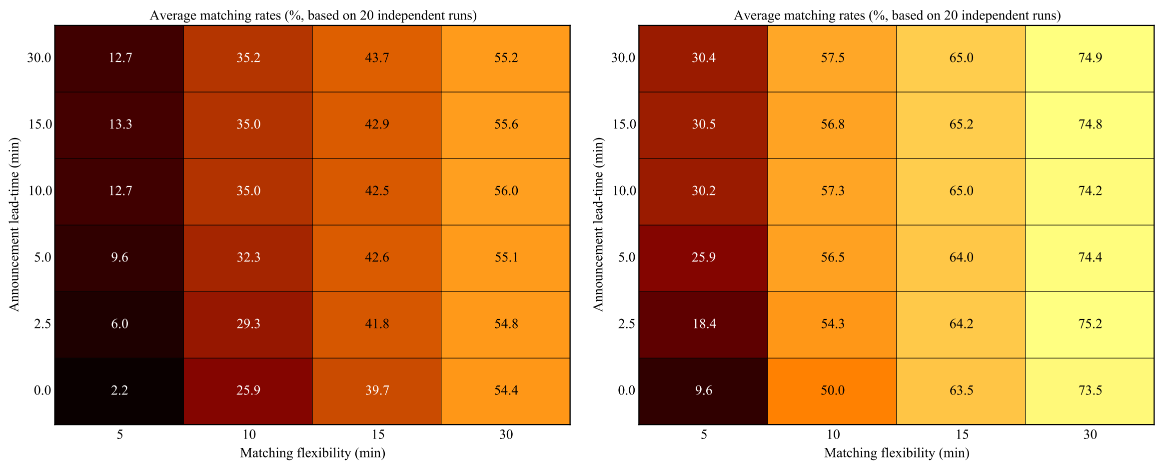
Rezultati na sliki 5 kažejo, da vodilni čas pomembno vpliva na stopnjo ujemanja pri nizkih vrednostih fleksibilnosti prirejanja. Kombinacije kratkega vodilnega časa (≤ 5 min) in nizke



(a) Geografija - koridor.

(b) Geografija - center mesta.

Slika 4: Povprečne stopnje ujemanje (Matching rate) za različne kombinacije sistemske fleksibilnosti obvoza (system-wide detour flexibility) in števila najav uporabnikov (Number of trip announcements). Rezultati so povprečja 20 neodvisnih ponovitev.



(a) 500 najav uporabnikov.

(b) 2.000 najav uporabnikov.

Slika 5: Povprečne stopnje ujemanje (Matching rate) za različne kombinacije sistemske fleksibilnosti prirejanja (system-wide matching flexibility) in vodilnega časa najave (Announcement lead-time).

fleksibilnosti prirejanja ($\leq$ 10 min) zelo omejujejo delovanje sistema za deljenje prevoza. Opozoriti je treba, da bi v realnem okolju ti rezultati po vsej verjetnosti bili še slabši, saj v simulaciji ne upoštevamo vseh mogočih neučinkovitosti, ki lahko pri tako kratkih vodilnih časih pomembno vplivajo na komunikacijo med sistemom in uporabniki.

V nadaljevanju smo naredili še vrsto drugih eksperimentov, ki so bolj ali manj variacije eksperimentov, ki smo jih predstavili v tem povzetku. Na podlagi rezultatov eksperimentov lahko potegnemo naslednje zaključke:

- fleksibilnost uporabnikov igra ključno vlogo pri lajšanju procesa iskanja ujemanj v sistemih deljenja prevoza,
- da bi sistem lahko uspešno deloval, morajo vozniki in potniki biti fleksibilni pri izbiri časa odhoda ter prihoda (vsaj 10 do 15 minut),

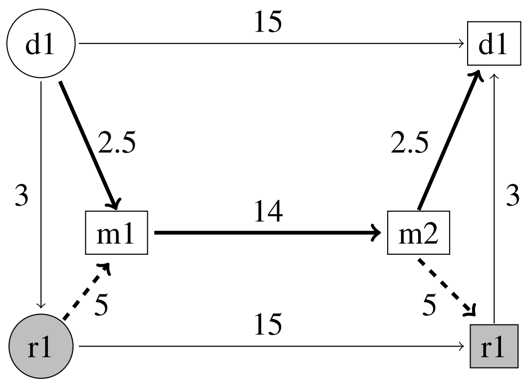
- izjemno pomembno za dobro delovanje sistema je, da so vozniki pripravljeni delati daljše obvoze – rezultati kažejo, da je ravno s povečevanjem pripravljenosti voznikov, da podaljšajo svoj vozni čas, mogoče doseči največja povečanja števila ujemanj.

Rezultati simulacij kažejo na potencial vpeljave spodbud za povečevanje fleksibilnosti uporabnikov in lahko služijo kot strokovna podlaga pri njihovem snovanju ter vpeljavi.

Koristi zbornih mest v sistemih deljenja prevoza

Do zdaj so modeli v literaturi predpostavljali, da potniki vstopajo v vozilo in izstopajo iz njega na svojih dejanskih izvorih ter ponorih. Kakorkoli, ta predpostavka je precej omejujoča, saj se mora samo ena stran prilagajati drugi. Z vključitvijo standardnih nabornih mest v sistem je omogočeno, da se potnik iz svojega izvora premakne na najugodnejše naborno mesto oz. da se z določene ugodne naborne točke pomakne proti svojemu ponoru z različnimi transportnimi modalitetami. Takšne naborne točke se uporabljajo v veliko sistemih deljenja prevozov, vendar še niso bile omenjene oz. uporabljene v optimizacijski literaturi. Algoritem, ki smo ga zasnovali za rešitev tega problema, skuša optimirati ujemanja z vidika dveh kriterijev: maksimiranje števila uparjenih uporabnikov in maksimiranje števila prihranjenih prevoženih kilometrov v celotnem sistemu.

Graf na sliki 6 prikazuje primer deljenja prevoza med voznikom $d1$ in potnikom $r1$ ter dvema zbornima točkama $m1$ in $m2$. Številka nad povezavo predstavlja potovalni čas med vozliščema. Brez uporabe zbornih točk ni možno povezati voznika in potnika, saj mora voznik podaljšati trajanje svoje poti za 6 minut, da bi se lahko prilagodil potniku, česar ni pripravljen storiti. Če pa je potnik pripravljen hoditi dodatnih pet minut do naborne točke in od nje, pa je deljenje prevoza možno, saj se je obvoz za voznika ustrezno znižal.



Slika 6: Potnik (siva barva) in voznik (bela barva) potujeta od izvora (krog) do ponora (kvadrat) preko zbornih točk.

Da bi omogočili avtomatsko povezovanje voznikov in potnikov v sistemu deljenja prevoza, je treba definirati optimizacijski problem in razviti primeren algoritem za njegovo rešitev. V tej študij smo problem definirali na način, da lahko voznik naredi največ dva postanka, in sicer enega, da potnika oziroma potnike pobere, in enega, da potnike odloži. Model, ki smo ga definirali, dopušča, da vozniku priredimo večje število potnikov. V tej študiji smo to število omejili na največ tri potnike, ker je to največje število odraslih potnikov, ki se še lahko udobno peljejo v večini osebnih vozil. Voznik tako lahko na enem zbornem mestu pobere in na drugem zbornem mestu odloži do tri potnike. Model upošteva večje število omejitev, ki so vezane na najhitrejši čas odhoda iz izvora in najpoznejši prihod na ponor, najdaljši še sprejemljivi čas potovanja, najdaljše trajanje hoje do zbornega mesta in z njega ipd.

Algoritem, ki smo ga razvili za rešitev optimizacijskega problema, deluje v dveh fazah: v prvi fazi generiramo vsa možna ujemanja med vozniki in potniki, v drugi fazi pa na podlagi identificiranih dopustnih ujemanj skonstruiramo celoštevilski linearni optimizacijski problem prirejanja potnikov oz. kombinacij potnikov voznikom.

V prvi fazi na učinkovit način generiramo vsa možna ujemanja med potniki in vozniki. Pri tem uporabljamo lastnost, da je dopustno ujemanje med voznikom in n potnikov možno le, če obstaja dopustno ujemanje za vsako izmed možnih kombinacij teh potnikov ter voznika. To pomeni, da lahko ujemanja generiramo rekurzivno, tako da v prvi fazi za določenega voznika generiramo samo ujemanja tega voznika s posamičnimi potniki. Ko generiramo ujemanja s pari potnikov, je treba za posamičnega voznika proučiti samo tiste potnike, ki so bili dopustna posamična ujemanja. Enaka logika velja tudi za trojice.

V drugi fazi gre za razširjen problem prirejanja, ki voznikom prireja posamične voznike oz. pare ali trojice voznikov. Problem je sicer NP-težek, vendar je zaradi svojih lastnosti hitro rešljiv s programom IBM CPLEX. Učinkovito lahko rešimo tudi velike primere problema prirejanja z več tisoč uporabniki.

V drugem delu študije smo nato naredili večje število simulacij na podlagi prometnega modela za širše območje mesta Atlanta v ZDA (Metro Atlanta). Namen teh simulacij je bil kvantificirati vpliv vključitve nabornih točk v sistem deljenja prevozov. Parametri, ki smo jih prevzeli v simulacijah, so povzeti v tabeli 2.

Tabela 2: Lastnosti osnovnih primerov deljenja prevoza v simulaciji.

Prometni vzorec:	migracija z obrobja v center
Povprečno število najav uporabnikov:	2849.4
Povprečno število najav voznikov:	1425.8
Povprečno število najav potnikov:	1423.6
Povprečna dolžina poti voznika:	7.58 mi
Povprečna dolžina poti potnika:	7.64 mi
Povprečno trajanje poti voznika:	30.34 min
Povprečno trajanje poti potnika:	30.56 min
Najdaljša razdalja do zborne točke:	0.5 mi
Hitrost pešca:	4 ft/s
Najdaljši čas hoje do zborne točke:	11 min
Hitrost vozila:	15 mi/h
Fleksibilnost obvoza voznika:	0.25
Fleksibilnost prirejanja voznika:	20 min
Kapaciteta vozil:	3 prosti sedeži

V tabeli 3 so povzeti rezultati za šest različnih scenarijev. Stolpec 0 prikazuje rezultate za sistem, v katerem ni nabornih točk. Stolpci 1-4 prikazujejo rezultate za različne stopnje gostote nabornih točk v sistemu, v stolpcu 1 imamo eno naborno točko na TAZ, v stolpcu 4 pa štiri naborne točke na TAZ. Mesto je v modelu namreč razdeljeno na manjše regije, imenovane TAZ (TAZ – travel analysis zone). V povprečju TAZ obsega površino 4,1 kvadratne milje. Scenarija 4* in 4** prikazujeta rezultate za najvišjo gostoto nabornih točk, s tem da so v scenariju 4* možna samo prirejanja med enim voznikom in enim potnikom, v scenariju 4** pa je možna uporaba samo najbližje naborne točke do potnikovega izvora oz. ponora. Kot je razvidno, lahko uporaba nabornih točk v sistemu bistveno poveča število ujemanj med vozniki in potniki, ki jih je možno vzpostaviti. V primerjavi z rezultati brez uporabe nabornih točk se je stopnja ujemanja povečala za 6,8 odstotne točke. Razvidno je tudi, da stopnja ujemanja ni bistveno nižja, če upoštevamo le ujemanja, v katerih sta en voznik in en potnik (povprečna stopnja ujemanja se zmanjša le za 0,7 odstotne točke). Po drugi strani je jasno vidno, da je za dobro delovanje sistema koristno, da je nabornih točk v sistemu veliko (z večanjem gostote točk v sistemu se pomembno veča tudi stopnja ujemanja) in da so potniki ter vozniki fleksibilni glede tega, na kateri naborni točki bodo vstopili ali izstopili (stopnja ujemanja za scenarij 4** je bistveno nižja kot za scenarij 4).

Tabela 3: Rezultati za različne gostote porazdelitve nabornih točk in različne tipe ujemanj.

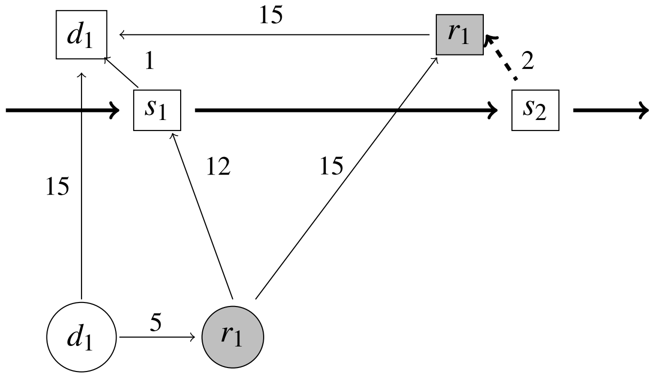
	0	**1**	**2**	**4**	**4***	**4****
Sistem:						
Stopnja ujemanja (%)	68.00	71.14	72.90	**74.83**	74.13	69.71
Prihranek prevoženih milj (%)	27.39	28.36	28.93	**29.63**	29.24	27.59
Vozniki:						
Stopnja ujemanja (%)	67.96	70.93	72.45	**74.08**	74.08	69.65
Povečanje trajanja poti (%)	25.45	25.98	26.31	**26.41**	26.19	25.77
Potniki:						
Stopnja ujemanja (%)	68.11	71.43	73.43	**75.65**	74.26	69.84
Povečanje trajanja poti (%)	13.09	19.27	22.74	**26.54**	16.43	16.42
Trajanje hoje (min:sek)	-	8:06	8:28	**8:56**	8:45	5:06

V nadaljevanju študij smo naredili še večje število simulacij, ki so bile namenjene preveritvi robustnosti rezultatov simulacije in vpliva določenih pomembnih parametrov. Zaključek tega poglavja je, da je vpeljava nabornih točk v sistem vsekakor koristna. Največje koristi je možno doseči, ko so tako potniki kot vozniki fleksibilni glede izbire primernih nabornih točk. Pomemben učinek je možno doseči tudi, ko potnik lahko doseže bolj oddaljene naborne točke, npr. z uporabo kolesa ali javnega transporta. Pomemben učinek ima tudi pripravljenost voznikov za delanje obvozov (kar smo že pokazali v prejšnjem poglavju).

Integracija sistema deljenja prevoza s sistemom javnega transporta

V tem poglavju predstavljamo nov model in algoritem za tvorjenje prevoznih skupin, ki omogoča integracijo s sistemom javnega transporta. Na ta način je omogočeno, da voznik potnika zapelje na njegovo končno destinacijo ali pa ga zapelje na postajo javnega transporta, od koder se z vlakom, avtobusom ali drugim prevoznim sredstvom pelje do svoje končne destinacije. Sistem poskuša sinhronizirati poti voznikov in potnikov z urnikom javnega transporta ter upošteva veliko število omejitev glede preferenc voznikov in potnikov (najhitrejši čas odhoda, najpoznejši čas prihoda, najdaljše trajanje poti ipd.). Podobno kot v prejšnjem poglavju tudi v tem algoritem, ki smo ga zasnovali za rešitev problema, skuša optimirati ujemanja z vidika dveh kriterijev: maksimiranje števila uparjenih uporabnikov in maksimiranje števila prihranjenih prevoženih kilometrov v celotnem sistemu.

Graf na sliki 7 prikazuje smisel integracije sistema za deljenje prevoza s sistemom javnega transporta. Gre za primer deljenje prevoza med voznikom $d1$ in potnikom $r1$ ter dvema postajama javnega transporta $s1$ in $s2$. Številka nad povezavo predstavlja potovalni čas med vozliščema. Brez uporabe javnega transporta ni možno povezati voznika in potnika, saj mora voznik podaljšati trajanje svoje poti za celih 20 minut, da bi se lahko prilagodil potniku, česar ni pripravljen storiti. Če pa je potnik pripravljen uporabiti tudi javni transport in hoditi dodatni dve minuti do končne destinacije, pa je deljenje prevoza možno, saj se je obvoz za voznika ustrezno skrajšal.



Slika 7: Potnik (siva barva) in voznik (bela barva) potujeta od izvora (krog) do ponora (kvadrat)

Da bi omogočili avtomatsko povezovanje voznikov in potnikov v sistemu deljenja prevoza, je treba definirati optimizacijski problem in razviti primeren algoritem za njegovo rešitev. V tej študij smo problem definirali na način, da lahko voznik naredi največ tri postanke, in sicer do dva postanka, da pobere do dva potnika, in enega, da potnika odloži na postaji javnega transporta. Dodatno v modelu tudi dopuščamo, da voznik potnika pobere in ga prepelje direktno do njegove destinacije, če je to dopustno in smiselno. Model upošteva večje število omejitev, ki so vezane na najhitrejši čas odhoda z izvora in najpoznejši prihod na ponor, najdaljši še sprejemljivi čas potovanja, najdaljše trajanje hoje do zbornega mesta in z njega ipd.

Algoritem, ki smo ga razvili za rešitev optimizacijskega problema, deluje v dveh fazah: v prvi fazi generiramo vsa možna ujemanja med vozniki in potniki, pri čemer upoštevamo urnik javnega transporta, v drugi fazi pa na podlagi identificiranih dopustnih ujemanj skonstruiramo celoštevilski linearni optimizacijski problem prirejanja potnikov oz. kombinacij potnikov voznikom.

V prvi fazi na učinkovit način generiramo vsa možna ujemanja med potniki in vozniki. Pri tem uporabljamo lastnost, da je dopustno ujemanje med voznikom in n potnikov možno le, če obstaja dopustno ujemanje za vsako izmed možnih kombinacij teh potnikov in voznika. To pomeni, da lahko ujemanja generiramo rekurzivno, tako da v prvi fazi za določenega voznika generiramo samo ujemanja tega voznika s posamičnimi potniki. Ko generiramo ujemanja s pari potnikov, je treba za posamičnega voznika proučiti samo tiste potnike, ki so bili dopustna posamična ujemanja. Enaka logika velja tudi za trojice.

V drugi fazi gre za razširjen problem prirejanja, ki voznikom prireja posamične voznike oz.

pare ali trojice voznikov. Problem je sicer NP-težek, vendar je zaradi lastnosti hitro rešljiv s programom IBM CPLEX. Učinkovito lahko rešimo tudi velike primere problema prirejanja z več tisoč uporabniki.

V drugem delu študije smo nato naredili večje število simulacij, s katerimi smo želeli kvantificirati vpliv integracije sistema za deljenje prevozov in sistema javnega transporta. Simulacija predpostavlja metropolitansko območje, ki meri 20 krat 10 milj. V sredini območja je urbani center. Izvore uporabnikov generiramo enakomerno naključno v celotnem pravokotnem območju, njihove ponore pa znotraj kroga, ki predstavlja center mesta. Sistem javnega transporta sestavlja šest linij, ki se križajo v centru. Tukaj je možno prestopiti na drugo linijo. Rdeči črti predstavljata hitrejša vlaka, ki povezujeta periferijo s centrom (glej sliko 3.2). Črne črte predstavljajo urbane linije znotraj mesta. Potovalna hitrost teh linij je v primerjavi s tistimi na periferiji nižja, postaje so postavljene bolj na gosto. V določenih scenarijih dopuščamo tudi uporabo parkirišč na obrobju mesta (t. i. koncept park-and-ride). Osnovni parametri simulacije so povzeti v tabeli 4.

Tabela 4: Lastnosti osnovnih primerov deljenja prevoza.

Prometni vzorec:	migracija z obrobja v center mesta
Povprečno število najav uporabnikov:	1000
Razmerje med vozniki in potniki:	0.5
Fleksibilnost prirejanja:	20 min
Fleksibilnost razporejanja:	15 min
Fleksibilnost obvoza:	25%
Fleksibilnost potnika:	50%
Največje število postankov voznika:	3
Povprečna dolžina poti voznika:	8.0 mi
Povprečno trajanje poti voznika:	24.1 mi
Največja razdalja do postaje javnega transporta:	0.5 mi
Hitrost pešca:	4 ft/s
Hitrost vozila:	20 mi/h
Hitrost regionalne linije:	40 mi/h
Hitrost urbane linije:	20 mi/h
Kapaciteta vozila:	2 sedeža
Čas pobiranja potnika:	2 min
Čas, potreben za transfer med avtom in vlakom:	2 min
Dodatno trajanje transferja za park-and-ride:	2 min
Čas postanka vlaka na postaji:	1 min
Čas postanka vlaka na osrednji postaji:	3 min
Frekvenca odhodov vlakov:	15 min
Število postaj javnega transporta:	41
Število postaj park-and-ride:	8

Rezultati osnovne simulacije so povzeti v tabeli 5.

Scenariji od leve proti desni so:

- RS – normalno deljenje prevoza (ni integracije z javnim transportom);
- TRS1 – deljenje prevozov z možnimi transferji enega potnika do postaje javnega

Tabela 5: Rezultati za različne primere deljenja prevoza (povp. 10 neodvisnih ponovitev).

	RS	**TRS1**	**TRS2**	**PTRS**
Potniki				
Stopnja ujemanja (%):	*66.8*	*74.0*	*83.7*	*83.8*
Transferji na postaje (%):	*0*	*32.4*	*37.2*	*33.9*
Transferji P+R (%):	*0*	*0*	*0*	*4.0*
Δ potov. čas (%):	*0*	*7.3*	*7.4*	*7.5*
Vozniki				
Stopnja ujemanja (%):	*68.3*	*75.5*	*73.8*	*74.2*
Transferji na postaje (%):	*0*	*32.4*	*27.3*	*25.1*
Transferji P+R (%):	*0*	*0*	*0*	*3.3*
Δ potov. čas (%):	*19.1*	*17.1*	*21.6*	*21.9*
Δ razdalja (%):	*8.4*	*7.0*	*10.4*	*7.2*

transporta;

- TRS2 – deljenje prevozov z možnimi transferji do dveh potnikov do postaje javnega transporta;
- PTRS – deljenje prevozov z možnimi transferji do dveh potnikov do postaje javnega transporta in možnostjo, da voznik svoje vozilo parkira na postaji javnega transporta ter pot nadaljuje z javnim transportom.

Kot je razvidno iz rezultatov, je integracija obeh sistemov lahko zelo koristna. V primerjavi z osnovnim scenarijem se stopnja ujemanja poveča s 66,8% na 83,7%. Pomemben rezultat je tudi, da vpeljava ujemanj, ko voznik na postajo prepelje do dva potnika, pomembno prispeva k večji stopnji ujemanja (povečanje stopnje ujemanja za cca 10 odstotnih točk). Vidimo tudi, da manjši odstotek voznikov (3,3%) uporabi tudi možnosti park-and-ride.

Zaključek

Upamo, da bodo metode, dognanja in rezultati, predstavljeni v tej disertaciji, koristno prispevali k boljšemu razumevanju, upravljanju ter delovanju sistemov za deljenje prevoza. Zlasti upamo, da bodo rezultati simulacij upravljavce sistemov za deljenje prevoza spodbudili k uvedbi raznovrstnih spodbud oz. motivacijskih shem za voznike in potnike ter k proučitvi možnosti integracije z javnim transportom in drugimi sistemi, kot so npr. sistemi deljenja koles.