UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Cristiano Martins Monteiro

Computational Aspects of Carsharing Planning Without Relocation Operations

Belo Horizonte
2022

Cristiano Martins Monteiro

Computational Aspects of Carsharing Planning Without Relocation Operations

Final Version

Dissertation presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Clodoveu Augusto Davis Jr.

Belo Horizonte
2022

Monteiro, Cristiano Martins.

M775c Computational aspects of carsharing planning without relocation operations [recurso eletrônico] / Cristiano Martins Monteiro – 2022.
1 recurso online (146 f. il, color.) : pdf.

Orientador: Clodoveu Augusto Davis Júnior.
Tese (Doutorado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f. 124 -131

1. Computação – Teses. 2. Mobilidade urbana – Teses. 3. Programação linear – Teses. 4. Simulação (Computadores) – Teses. I Davis Júnior, Clodoveu Augusto. II. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. III. Título.

CDU 519.6*61(043)

Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa
CRB 6/1510 Universidade Federal de Minas Gerais - ICEx

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

COMPUTATIONAL ASPECTS OF CARSHARING PLANNING WITHOUT RELOCATION OPERATIONS

CRISTIANO MARTINS MONTEIRO

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. Clodoveu Augusto Davis Júnior - Orientador
Departamento de Ciência da Computação - UFMG

Prof. José Alberto Quintanilha
Departamento de Engenharia de Transportes - Escola Politécnica da USP

Prof. Flávio Vinícius Cruzeiro Martins
Departamento de Computação - CEFET-MG

Prof. Cristiano Arbex Valle
Departamento de Ciência da Computação - UFMG

Prof. Vinícius Fernandes dos Santos
Departamento de Ciência da Computação - UFMG

Prof. Humberto Torres Marques Neto
Departamento de Computação - ICEI - PUC Minas

Belo Horizonte, 15 de dezembro de 2022.

Documento assinado eletronicamente por **Clodoveu Augusto Davis Junior**, **Professor do Magistério Superior**, em 16/01/2023, às 10:54, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Vinicius Fernandes dos Santos**, **Professor do Magistério Superior**, em 16/01/2023, às 11:50, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Cristiano Arbex Valle**, **Professor do Magistério Superior**, em 16/01/2023, às 12:14, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **José Alberto Quintanilha**, **Usuário Externo**, em 16/01/2023, às 13:22, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Humberto Torres Marques Neto**, **Usuário Externo**, em 30/03/2023, às 03:25, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2018650** e o código CRC **FF4A3486**.

I dedicate this work to my father Geraldo Monteiro da Silva, and my mother Creuzoni Martins Monteiro.

Acknowledgments

I am grateful to all acquaintances who helped or inspired me along my academic career. This includes family, research colleagues and professors.

Thanks to my mother Creuzoni M. Monteiro, my wife Janaina O. Souza, my brother Otaviano M. Monteiro and my stepfather Adelmo A. Gomes for cheering me up along all these years.

Thanks to the colleagues Michelle B. Pinheiro, Alexandre F. Faria, Gabriel P. de Oliveira, Mariana O. Lage, Danilo B. Seufitelli, Luci A. Nicolau, and Pedro V. Brum for being always available to clarify doubts.

Thanks to my advisor Clodoveu A. Davis Jr. for guiding me since before I become a regular student in the PhD program. I am also thankful to José A. Quintanilha and Cláudia A. S. Machado for discussing research ideas even after the project had been finished. And thanks to Vinícius F. dos Santos for helping me to prove the NP-Hardness of the Spacing Shared Mobility Stations problem.

Finally, I acknowledge the financial support received by Engineering Technological Development Foundation (Fundação para o Desenvolvimento Tecnológico da Engenharia – FDTE) and Coordination for the Improvement of Higher Education Personnel (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES). And I am thankful to God for enabling all about this to be possible.

"*Any endeavor without goals is bound to fail.*"
(Jain [1991])

Resumo

Deslocamentos são tarefas rotineiras das pessoas que vivem em áreas urbanas. Os serviços de mobilidade compartilhada visam oferecer diferentes opções a essa rotina, proporcionando melhor conforto e viagens mais rápidas que os meios de transporte público convencionais, além de evitar os custos dos clientes de possuírem um veículo particular. Um serviço de carsharing devidamente planejado pode ser atraente inclusive para quem possui e dirige um veículo particular mas consideraria deixar de possuí-lo se um meio de transporte mais barato e sustentável estivesse disponível. Um aluguel de carsharing de baixo custo pode ser alcançado posicionando adequadamente a frota ao longo da cidade e aproveitando ao máximo os veículos compartilhados de acordo com uma seleção prévia de qual subconjunto de demandas de viagens podem ser atendidas. Esta seleção prévia escolheria quais demandas têm origem e destino combinando, o que permite que esses clientes utilizem um mesmo veículo mas em momentos distintos, não exigindo que a empresa carsharing realoque a frota entre as estações devido às diferentes demandas ao longo do dia e da semana. Este trabalho contextualiza os desafios operacionais e computacionais do planejamento de um serviço de carsharing; prova a NP-Completude de otimizar os locais para estações de mobilidade compartilhada; propõe uma formulação de Programação Linear Inteira-Mista para este problema original e uma outra formulação de Programação Linear Inteira-Mista que produz boas localizações para estações; e aplica uma formulação linear de tempo polinomial para simular e comparar o desempenho de três diferentes modelos de negócio de carsharing de acordo com dados históricos de mobilidade da Região Metropolitana de São Paulo. Resultados mostram que é possível oferecer um serviço de carsharing de baixo custo e lucrativo sem realizar relocações de veículos. Porém, somente um subconjunto das viagens são atendidas e é necessário que clientes sejam flexíveis para caminhar até algum veículo disponível na região. Resultados também demonstram que as viagens selecionadas para serem servidas são similares entre os diferentes modelos de negócios; estão concentradas na região central de São Paulo; são mais curtas que a média das viagens, mas tem padrões similares ao restante das viagens; e a falta de vagas de estacionamento pode ser um risco para empresas de carsharing.

Palavras-chave: Mobilidade de Baixo Custo, Programação Linear Inteira-Mista, Simulação

Abstract

Commuting is a routine task of people living in urban areas. Shared mobility services aim to offer different options to this routine by providing better comfort and faster trips than conventional public transport means, along with avoiding the clients' costs of owning a private vehicle. A properly planned carsharing service can be attractive even for who owns and drives a private vehicle but would consider not owning it anymore if a cheaper and more sustainable transport mean is available. Low-cost carsharing rentals can be achieved by suitably positioning the fleet along the city and by making the most of the shared vehicles according with a previous selection of which subset of trip demands can be served. This previous selection would choose which demands have a combining origin and destination, allowing these clients to use a same vehicle but in different moments, not requiring the carsharing company to relocate the fleet among stations due to different demands along the day and week. This work contextualizes the operational and computational challenges in planning a carsharing service; proves the NP-Completeness of optimizing the locations for shared mobility stations; proposes a Mixed-Integer Linear Programming formulation for this original problem, and another Mixed-Integer Linear Programming formulation which yields good locations for stations; and applies a polynomial time linear formulation to simulate and compare the performance of three different carsharing business models according with historical mobility data from the São Paulo Metropolitan Area. Results show that it is possible to offer a profitable low-cost carsharing service without performing vehicle relocations. However, only a subset of trips are served and clients must be flexible enough to walk to get to an available vehicle nearby. Results also demonstrate that trips selected to be served are similar among the different business models; are concentrated on São Paulo's downtown region; are shorter than the average trip, but otherwise behave in a similar way as compared to the complete set of trips; and the lack of parking slots may be a risk to the carsharing company.

Keywords: Low-cost Mobility, Mixed-Integer Linear Programming, Simulation

List of Figures

List of Tables

List of Algorithms

Contents

Chapter 1

Introduction

Daily commuting is among the main routines of people living in urban areas. People commute to work, to school, for shopping or for leisure purposes, and thus spend part of their day by moving around in the city. Even though the bus and subway lines are often the cheapest transportation services, these services can show drawbacks to their passengers, such as:

- few options (or inexistence of) bus and subway lines in certain regions;
- low supply of buses at certain times;
- need to transfer among bus and subway lines;
- lengthy walks to get to a bus stop or subway station;
- uncomfortable trips (in certain city regions and/or peak times).

These drawbacks, together with long periods spent in commuting, intensified by the urban population[1] increase[2], have motivated studies to improve the transportation services. Among these studies are: identifying transport-related social exclusion [Logiodice et al., 2015], suggesting new transportation services [Jorge et al., 2015], building applications to integrate, visualize, and analyze data of public transportation [Alic et al., 2018], improving the accessibility for public transportation [Monteiro et al., 2018], understanding job inaccessibility inequalities [Tomasiello et al., 2020], optimizing the assignment of buses to travels [Veloso-Poblete et al., 2018], reducing energy consumption costs of electric buses [Bartłomiejczyk, 2018], reducing transportation time and costs by integrating ride-sharing to public transportation [Stiglic et al., 2018].

Alternative transportation services can reduce those drawbacks without substantially increasing the involved costs. Shared mobility modalities have grown in urban areas by providing more sustainable [Cohen and Kietzmann, 2014; Machado et al., 2018] and affordable transportation services [Benetti, 2019; Violin, 2021], reducing the urban emissions of polluting gases [Martin and Shaheen, 2011], diminishing parking needs [Tchervenkov

[1] https://data.worldbank.org/indicator/SP.URB.TOTL.IN.ZS

[2] https://data.worldbank.org/indicator/SP.URB.TOTL

et al., 2018], and relieving traffic jams [Luan et al., 2018]. Among the shared mobility modalities, on-demand transport services, together with sustainable awareness and low-cost requirements, are shaping the emergent transportation modes[3] in the cities [Roy et al., 2018; Machado et al., 2018; Benetti, 2019; Aguilera-García et al., 2020].

Amid these on-demand transport services is the sharing of vehicles, usually named as carsharing. Carsharing consists in offering vehicles in an "as-needed" basis. Clients can rent cars for periods as little as some minutes, avoiding the costs of owning a vehicle or of renting it for a whole day [Jorge et al., 2015; Machado et al., 2018]. Since the client is also the driver, there is no need to pay a taxi driver. Therefore, carsharing fares tend to be lower than the fares from taxi or similar services, like Transport Network Companies (TNC) such as Uber and Lyft [Schwieterman and Bieszczat, 2017].

Carsharing services can be classified into three modes: round-trip, one-way, and free-floating. The round-trip and one-way are station-based modes, therefore, the rentals must start and end in the carsharing company's stations. On the round-trip, clients must return the rented car at the same station where the rental has started. On the one-way mode, clients can return the vehicle in a different station. On the free-floating mode there are no stations: vehicles are picked-up or dropped-off in any place within the area of operations [Machado et al., 2018].

Next section presents the motivation involved in the research on shared mobility, more specifically on the carsharing topic.

1.1 Motivation

According to the S.E.U. [2003] from the United Kingdom, transport problems can hamper people's access to services and opportunities, reinforcing their status of social exclusion. The lack of transportation services can also bring difficulties to build social networks and reduce family recreational day trips, being harmful to well-being and mental health. Few mobility options can also restrict people to walking to locally accessible grocery stores, being subject to higher grocery prices and having fewer healthy food options. Besides, the absence of mobility options can hamper opportunities of employment, education, and training that could increase income [Mackett and Thoreau, 2015].

The authors from S.E.U. [2003] defined transport accessibility as people being able to get to key services at a reasonable cost, in reasonable time and with reasonable ease. Also, accessible transport should be reliable, passengers should feel safe using it and people should be financially and physically able to access the transport. Low-accessibility

[3] https://silvio.meira.com/silvio/olha-um-carro-compartilhado-autonomo/

transportation can cause problems for work, learning, healthcare, shopping, social, and sporting activities.

Transport-related social exclusion is often more properly explained by non-transport factors and issues such as economic power, accessibility, and choice. For example, while people with money, time and good health to drive can choose to enjoy driving to an attractive rural environment, inhabitants of these rural areas (lacking services such as healthcare and shopping) can be "forced" to own and maintain a vehicle in order to have access to essential services. Therefore, being "forced" to have a vehicle is a factor of experiencing transport-related social exclusion, differently from choosing to own a vehicle while having other reasonable options of mobility [Pooley, 2016].

Shared mobility modes represent a promising set of solutions for vulnerable groups that are neither properly served by public transport nor can comfortably afford a private vehicle [Viegas and Martinez, 2017]. People who are able to drive may prefer using carsharing instead of ridesourcing services, such as from Uber and Lyft, to either avoid the costs of paying a driver for the trip or not depending on the availability of a driver to perform the trip[4,5]. This latter situation includes Brazil due to the increase in gas prices, demotivating the ridesourcing supply of drivers[6,7,8]. If low-cost carsharing services are available, drivers who own cars may become motivated to become carsharing clients. To do so, low-cost carsharing services must be properly planned to ensure low-cost prices and wide coverage [Correia and Antunes, 2012; Jorge et al., 2014, 2015; Genikomsakis et al., 2017; Lage et al., 2019, 2021]. However, planning such services is not an easy task. The following section discusses the computational challenges on planning and optimizing carsharing services.

1.2 Computational Challenges of Carsharing

Essential efforts of carsharing planning are within the class of the hardest problems in Computer Science. This class is known as NP-Hard [Garey and Johnson, 1979; Cormen et al., 2009] and includes tasks such as selecting places that should have electric vehicle

[4] https://www.washingtonpost.com/technology/2021/05/07/uber-lyft-drivers/

[5] https://www.dailymail.co.uk/news/article-9883145/Struggling-Uber-Pay-row-driver-shortages-multi-apping-causing-nightmare-users.html

[6] https://www.uol.com.br/carros/noticias/redacao/2021/10/04/por-que-os-motoristas-de-aplicativo-estao-cancelando-tantas-corridas.htm

[7] https://www.istoedinheiro.com.br/entenda-porque-motoristas-de-aplicativos-estao-cancelando-corridas/

[8] https://www1.folha.uol.com.br/mercado/2021/12/taxi-fica-mais-vantajoso-com-apagao-de-uber-em-sao-paulo.shtml

chargers [Lam et al., 2014], and organizing routes for multiple workers to relocate shared vehicles between stations [Albinski, 2015].

Solving such problems efficiently is a challenge for carsharing companies, which may constantly depend on such simulation or optimization tasks, since their prices are constrained by the prices of competitors. As an example, potential carsharing clients may prefer using ridesourcing services if the difference of prices is not significant. This is expected because ridesourcing clients neither need to drive, nor to walk to get to an available vehicle or walk from where the vehicle was parked at the end of rental to get to their actual destination. In recent years, the competition between carsharing and ridesourcing prices increased due to the ridesplitting modality [Schwieterman and Bieszczat, 2017]. In this modality, a ridesourcing client may serve other clients during a trip if their trips have overlapping routes. By doing so, ridesplitting clients have their fares split, reducing their costs [Schwieterman and Bieszczat, 2017; Machado et al., 2018]. As a broader example of low-cost competitor, Silva Jr et al. [2018] showed that it is possible to find interesting trade-offs between price and trip duration by integrating public transportation with a possible ridesplitting taxi service in New York.

Therefore, luring clients for carsharing includes ensuring low prices and offering available services nearby their clients' departure and destination places. Although the availability can be achieved by increasing the number of vehicles, stations and covered area, the more resources are included, the more expensive it becomes to the carsharing company. The trade-off between improving accessibility and profits of the service can also be dealt through simulation and optimization techniques. For example, Correia et al. [2014] and Boyacı and Zografos [2019] evaluated how the flexibility of one-way clients on walking short distances to get an available vehicle may improve carsharing dynamics. It is expected that a carsharing client will be flexible to walk short distances because competitors such as public transportation and ridesplitting services may also require it, and even if he/she owns a private car, sometimes he/she will also have to walk hundreds of meters between where the car finally could be parked and his/her actual destination [van der Waerden et al., 2017].

Nevertheless, the more business rules are included in the simulation or optimization techniques, the harder it may become for a computer to run it and obtain optimal solutions in a viable time [Cristian et al., 2019; Mellou et al., 2020]. In this scenario, methods with no optimality guarantee are often used, since they produce good solutions and run faster (usually in polynomial-time) [Ritzinger et al., 2016; Mourad et al., 2019; Murray et al., 2020].

A problem is solved in polynomial-time if, for all applicable instances, the time required to solve it can be expressed by a polynomial in function of the instance size. It differs from NP-Hard problems since, at least so far, the time required to solve them is expressed only by exponential functions regarding the instance size. Therefore, as the instance size increases, the time required to solve a NP-Hard problem increases much more (in an exponential pace) than the time required to solve another problem but in polynomial-time (increases in a polynomial pace) [Garey and Johnson, 1979; Cormen et al., 2009].

Although carsharing simulation and optimization have been extensively studied in the recent years, the computational aspects of such techniques were not widely researched yet. The following section presents the research questions to be answered in this work.

1.3 Research Questions and Objectives

The major research question of this work is:

Is it possible to computationally plan and simulate, for a region as wide as the São Paulo Metropolitan Area, a profitable low-cost one-way carsharing service without vehicle relocation operations and considering the clients' flexibility to walk?

By answering that major research question, minor research questions are also approached:

- How do the profits of a carsharing company vary between the round-trip and one-way modes?
- How does the computational performance improve by changing the optimizing formulation for fleet-sizing?
- How is the vehicle and parking slots availability impaired in an one-way carsharing service without vehicle relocation operations?
- How much flexibility in walking to nearby stations is needed to improve carsharing dynamics?
- Is spacing the location of stations an NP-Hard problem?
- Is it possible to select, in polynomial-time, a subset of drivers whose routine trips match among themselves, allowing an one-way carsharing service without relocation operations?
- Where should stations be located to maximize the possible origin and destination locations for the expected demand?
- Which operational challenges will a carsharing service face?

The main objective of this dissertation is:

Propose useful optimization methods for planning a profitable low-cost one-way carsharing service without relocation operations.

In order to fulfill this main objective and to answer those research questions, some specific objectives must be accomplished:

1. Compare the expected profits of round-trip and one-way modes;
2. Compare the time spent on solving different fleet-sizing Mixed-Integer Linear Programming (MILP) formulations;
3. Evaluate different scenarios of supply and demand to analyze how the vehicle and parking slots availability decreases in an one-way carsharing service when no vehicle relocation operations are performed;
4. Evaluate the increase in clients served when they are flexible to walk to nearby stations to get an available vehicle or parking slot;
5. Prove that optimizing the location of stations is an NP-Hard problem;
6. Propose a Linear Programming (LP) formulation for selecting a subset of drivers whose routine trips match among themselves, allowing an one-way carsharing service without relocation operations;
7. Design and implement a database describing the expected demand of trips and identifying which stations are nearby each drivers' trip demand;
8. Compare the computational performance of solving MILP formulations to find good locations for stations;
9. Use the location of stations obtained on the previous objective to select drivers matching their demand and simulate the one-way carsharing service;
10. Analyze the risks of the one-way carsharing service without relocation operations having an outage of available vehicles or parking slots due to clients occasionally being not able to use the service, disrupting the expected carsharing flow.

The analyses presented here can be immediately useful for carsharing decision-makers, mainly for those operating (or planning to operate) in São Paulo, Brazil, due to the dataset of trip demands used in this work. The next section presents an overview of the remaining chapters of this work.

1.4 Structure of this Work

The next chapter discusses related works and presents our contributions to the literature. Chapter 3 presents the initial and already published results, which inspired the carsharing business model proposed in Section 3.4 and the development of the other chapters. Chapter 4 presents the proposed linear formulations for optimizing the locations of carsharing stations and for simulating the one-way carsharing service; and describes the algorithm for calculating the shortest distances in the pedestrian ways network. Chapter 5 describes the software and hardware resources used, the diagram of the spatial database used, and the algorithm used to prepare the instances for solving the optimization problems. Chapter 6 presents and discusses the simulations made for different carsharing business models. Finally, Chapter 7 concludes this thesis and suggests future works.

Chapter 2

Related Work

Complex transportation problems have recently been solved using simulation and optimization approaches. These approaches are useful since they enable the evaluation of parameter changes, being useful to support "what-if" analyses for the decision-making process [Jorge et al., 2015; Genikomsakis et al., 2017; Cristian et al., 2019; Boyacı and Zografos, 2019; Mellou et al., 2020; Schiffer et al., 2021]. The following works made contributions related to the main subject of this work.

A MILP model is proposed in Correia and Antunes [2012] to maximize the profits of a carsharing company considering all the revenues and costs involved. Since the one-way mode allows clients to deliver the vehicle in another station, the number of vehicles in each station will change along the day due to their different demands. The work aims to optimize the locations for carsharing stations, avoiding fleet unbalancing among the stations on the one-way mode. The authors showed, in a case study in Lisbon, Portugal, the impact of the location of stations for the system performance and considered different scenarios of client behavior.

That work was extended in Correia et al. [2014] to evaluate the impact of one-way clients being flexible to walk to the closest or second closest station to get an available vehicle or parking slot. The authors simulated the walking time between stations rather than the walking distance. Results show that flexible clients knowing which stations have available resources would have to walk about eight minutes to get to those stations. However, scenarios with clients that are flexible to walk can serve twice as many clients than scenarios with clients inflexible to walk. Therefore, the location of stations plays an important role on the carsharing dynamics. The same happens with the spatial distribution of chargers for electric vehicles.

Lam et al. [2014] showed that the problem of optimizing the location of chargers for electric vehicles is NP-Hard. The authors proposed four different methods for solving this problem: two MILP formulations, one greedy heuristic, and a meta-heuristic nature-inspired on Chemical Reaction Optimization. These four methods were applied to five different problem instance sizes and had their run times and solutions assessed. Numerical results based on Hong Kong, China, showed that the methods producing better solutions also required more run time, being suitable to different situations.

Different optimization methods were also evaluated for optimizing the carsharing service. Jorge et al. [2014] compare two methods for the relocation of carsharing vehicles to fix the unbalancing among stations on the one-way mode, in Lisbon, Portugal. The method based on MILP overcame the method built using the AnyLogic[1] environment for simulation, mainly while considering vehicle relocation.

Vehicle relocations are common for one-way mode, since they address two issues:

1. stations unable to serve clients because earlier clients rented all available vehicles and delivered them to other stations;

2. clients unable to deliver the vehicle in the desired station because all existing parking slots are occupied.

Issue (2) can be even more restrictive to the one-way dynamics, because it can happen on the first rentals of the day. In addition, that issue can happen even when the fleet is balanced, with vehicles distributed to all stations. If one-way stations have a low number of parking slots, and the stations are far apart, relocation operations can be expensive and unable to solve issue (2) quickly enough. That happens because stations with a low number of parking slots will more often be subject to issue (2), and due to the fact that the vehicle cannot be used while being relocated. Even with relocations, a necessary approach to avoid issue (2) consists in the carsharing company holding more parking slots than vehicles to rent.

Repoux et al. [2014] optimizes the fleet size of a carsharing service in Nice, France using relocation operations. The authors have considered a carsharing mode where clients can park the rented vehicles somewhere close to the station if no parking slots are available. This carsharing mode was named *partial floating*. Although partial floating could solve issue (2), it depends on available public parking area nearby the stations. Besides, if the carsharing service is based on electric vehicles and if these vehicles need to be often recharged, partial floating will only be effective if there are chargers available in the public parking areas nearby, in order to recharge the idle vehicles. Results showed that relocation operations could reduce the problems caused by issue (2) and avoid high accumulation of vehicles outside stations.

Jorge et al. [2015] propose a MILP model to optimize the design of a carsharing service based on round-trip to also offer one-way rentals in Boston, USA. The results show that adding an optimized one-way mode to the service could increase the earned profit. Even though the one-way mode can support more clients, many carsharing companies do not offer one-way mode due to the effort and costs involved on relocation operations. The vehicle relocation problem was proved to be NP-Hard by Albinski [2015]. The author also showed that this problem is NP-Complete and evaluated branch-and-cut methods for solving it faster.

[1] https://www.anylogic.com/

Böhmová et al. [2016] showed that the carsharing fleet-sizing problem with no relocation operations varies from polynomial-time to NP-Hard depending on which objective is being optimized and how the service works. If the objective is to minimize the number of vehicles used, and clients are served only if all their demands are satisfied, then this problem is solvable in polynomial-time. However, if the objective is to maximize the number of satisfied clients and each client has two demands (move from one station to another and afterwards move back to the initial station), this problem becomes NP-Hard.

Hara and Hato [2017] proposed an one-way carsharing system in which instead of workers performing relocation operations, clients trade a permit to use the vehicle whose price is defined by the market. Therefore, renting prices would vary depending on demand, and these prices could even become negative (people earn to use the cars) when the trip to be made is infrequent and would help the carsharing dynamics similarly to a relocation operation. Fleet-sizing under this business model is said to be NP-Hard, and the authors proposed a polynomial-time LP model for instances in which every client only trade for one permit to use the vehicle. Experimental results were evaluated using randomly generated data.

Du et al. [2018] proved that optimizing the location of chargers for electric vehicles finding a balance between coverage of points of interest and local demand is also NP-Hard. The authors also proposed a greedy approximation algorithm with a ratio $1 - \frac{1}{e}$ and a time complexity of $O(m^2 n)$, where e stands for the amount of candidate locations, m is the number of charging stations, and n is the number of points of interest. Experimental results based on a carsharing dataset from Beijing, China were also presented.

Bruglieri et al. [2018] proposed a Multi-Objective MILP model for the one-way mode in Milan, Italy. The proposed model has three objectives: minimizing the number of workers needed to relocate vehicles; maximizing the number of vehicle relocations to increase the number of served clients; and minimizing the lengthiest relocation route. Approximate heuristic solutions were compared with exact MILP ones observing the computational time spent for both methods. Results show the benefits of using the approximate heuristic methods instead of waiting for the exact optimal solution.

Lu et al. [2018] proposed a stochastic MILP based on Benders decomposition to analyze the number of necessary vehicles, fleet used percentage, vehicle relocation costs, and metrics on the quality of the carsharing service in the Boston-Cambridge area in Massachusetts, USA. The results showed that companies offering both round-trip and one-way modes can have their profits for one-way decreased if the demand of clients is generated by pricing and strategic customer behavior instead of natural market penetration and user adoption. That happens because if the one-way demand is driven by pricing, for example, round-trip clients could prefer making two one-way rentals since it could be cheaper. Therefore, instead of increasing the demand, pricing could rather reduce the company's profit margin since one-way relocations can be costly.

Although relocation is important for balancing the fleet, it may not significantly increase the carsharing company's profit. As simulated in [Santos and Correia, 2019] for a demand of 25% of all potential clients in the city of Lisbon, Portugal, and considering that all relocations would result in new clients served, relocations would increase the company's profit by only 3.2%. It is possible that both issues (1) and (2), mentioned earlier, regarding one-way unbalancing in stations, are better avoided by approaches other than vehicle relocations. If the carsharing investment is complemented by more vehicles and parking slots, carsharing companies can also avoid both issues (1) and (2). Furthermore, the management of one-way rentals can be as simple as the round-trip ones, operation costs can be reduced and the rental prices can be cheaper. Lower rental prices can attract more clients, and also motivate them to walk to another station to get an available vehicle or parking slot.

Similar collaborations among clients were already evaluated in the literature for one-way carsharing with electric vehicles [Cocca et al., 2019; Boyacı and Zografos, 2019; Ströhle et al., 2019]. Cocca et al. [2019] presented benefits derived from clients driving free-floating electric vehicles up to nearby charging stations, instead of parking them in any other place. Simulations based on the cities of Turin, Italy; Milan, Italy; Berlin, Germany; and Vancouver, Canada were applied to define locations to place electric chargers to avoid carsharing vehicles running out of battery. Among the results, it was found that it is better to place chargers in popular areas, such as downtown, instead of in areas where vehicles tend to be parked for long periods.

Boyacı and Zografos [2019] evaluated the collaboration of clients regarding their spatial flexibility (being available to walk to another station) and temporal flexibility (being available to start their rental earlier or later than expected) over a carsharing dataset from Nice, France. Among other results, authors found that spatial flexibility could increase the number of clients served in up to 25%, having a stronger effect on the system's performance than temporal flexibility, which increased the number of clients served by less than 5%.

Ströhle et al. [2019] evaluated the potential benefits of the spatial and temporal flexibility of one-way carsharing clients. Similarly to Boyacı and Zografos [2019], the authors have found that temporal flexibility has a low potential in improving the carsharing dynamics. Even if clients are flexible to postpone their trips by up to four hours, the fleet size could only be reduced by 4%. However, if clients are flexible to walk up to 1 kilometer, the fleet size could be reduced by 12%. This reduction could reach almost 20% if the spatial and temporal flexibility are combined: clients being flexible to walk up to 750 meters and postpone their trips by up to three hours. These results were obtained after optimizing the carsharing system using as input a real dataset with more than 50,000 carsharing reservations from a mid-sized German city. The authors also conducted an online survey to assess the willingness of clients in being flexible. Among the obtained results, the study found that clients agree to be more flexible, in exchange for not having to pay more to be served, if they will perform “long trips”. In the survey, the term “long trips” was used to ask about supposed carsharing rentals with driving distance of 37,5 kilometers, in contrast to “short trips”, whose considered driving distance was 15 kilometers. The implemented fleet sizing MILP is said to solve an NP-Hard problem.

Deza et al. [2020] optimized the locations of chargers for electric vehicles of a carsharing service. The optimization was performed using data from Toronto, Canada, and its surrounding area. The authors proposed a LP formulation which can be solved in polynomial-time. However, in the worst case it may require an exponential number of potential charger locations.

Brandstätter et al. [2020] proved that the problem of optimizing the locations of chargers for electric carsharing vehicles with budget constraints (costs for opening a station and costs per electric charger) and battery capacity constraints is NP-Hard. Besides, the authors showed that if neither budget nor battery capacity constraints are included, the problem can be solved in polynomial-time by modeling it as a minimum-cost flow. The battery levels of each car were explicitly tracked in the proposed MILP formulations, whose computational performance were evaluated over instances from Vienna, Austria. Similarly to Lam et al. [2014], Du et al. [2018] and Deza et al. [2020], the problem proposed in Brandstätter et al. [2020] optimizes the location of stations by selecting candidate points, instead of defining locations directly along the road network.

Schiffer et al. [2021] optimized a carsharing service based in Vancouver, Canada, considering that the relocation operations would be performed by the clients themselves. According to the authors, this problem equals to the k-disjoint Shortest Paths Problem with negative weights. The computational time complexity in the worst case is $O(|\mathbb{A}|(|\mathbb{V}|+k) + k|\mathbb{V}|log|\mathbb{V}|)$ where $|\mathbb{A}|$ stands for the number of arcs (trips being served) in the network, $|\mathbb{V}|$ is the number of vertices (possible vehicle situations after moving by serving trips) and k is the number of vehicles. Results indicate that, after optimized, the number of served requests can increase by 21% and the company's revenue can increase by 10%.

Another polynomial-time method for carsharing fleet-sizing is proposed by Liu et al. [2022]. The authors evaluated the performance of autonomous vehicles used for carsharing and the ridesplitting modality of ridesourcing using data from Langfang, China. Since the vehicles are autonomous, no staff was considered to perform relocations. In that work, trips were modeled as vertices and the trips' temporal order and similarity were modeled as edges. The matching of client pairs to be served together (ridesplitting) was performed by solving the Maximum Weighted Matching Problem. The minimum fleet size for serving the trips was found by solving the Minimum Path Cover Problem on the directed acyclic graph formed by the trips' order. Since the carsharing service works in the free-floating modality and the cars can relocate themselves, the stations' locations were simply defined as areas of high demand. Results showed that one shared autonomous vehicle can replace 7.84 private conventional vehicles if all evaluated clients use the service alone, and replace 15.52 private conventional vehicles if all clients are flexible to split his/her trip with other client. However, the total distance traveled by the shared autonomous vehicles in the scenarios without ridesplitting is expected to be higher than the total distance traveled by private vehicles.

The fleet-sizing problem was also approached by Zhang et al. [2022] with simulations based on Beijing, China. The authors proposed a MILP formulation to optimize the size of a heterogeneous electric fleet. This problem considers vehicle relocations and is said to be NP-Hard. The authors applied Dantzig–Wolfe decomposition to the proposed MILP model and used ant colony optimization into the column generation framework to speed-up the pricing subproblems optimization. It was found that the proposed method is faster than solving the MILP using CPLEX, and usually finds solutions with better objective values when considering large instances. The authors also noticed that not serving some rental requests would be more economically beneficial than serving every demand.

Lai et al. [2022] dealt with the problem of dynamically optimizing the relocation of electric carsharing vehicles. This work was based on Beijing, China; all trips are on-demand (without any scheduling); and it assumes that clients are flexible to walk to be served. The clients' flexibility was used to offer discounts to clients who may walk to another station with available resources. This same rationale was applied into heuristics to avoid using staff to relocate vehicles in all situations. The authors compared three operational methods to organize the relocations: a greedy policy based on using resources to always serve the first trip demands; a proposed Iterated Local Search heuristic; and a version of Particle Swarm Optimization. The Iterated Local Search heuristic outperforms the other two methods evaluated.

Although carsharing planning may resemble traditional combinatorial optimization problems such as Facility Location Problem and Vehicle Routing Problem, the carsharing problems differ from those traditional problems due to the business specific constraints imposed. Therefore, analyses of time complexity are important to better control the efforts spent on optimizing a carsharing service. Table 2.1 indicates how this work's contribution fits in the literature among the related works that evaluate the clients' flexibility or discuss their models' computational time complexity.

This thesis differs from the related works by proposing a MILP formulation to optimize the exact location of shared mobility stations instead of selecting candidate locations, proving it to be NP-Complete, proposing a polynomial-time fleet-sizing LP formulation for carsharing without relocation operations, and considering different levels of spatial flexibility by clients.

Every mark in Table 2.1 indicates that the work of that line approached the feature

[2] Although Deza et al. [2020] proposed an LP model solvable in polynomial-time, it requires an exponential number of variables in the worst case.

Table 2.1: Contributions of this work compared to the related works

Related Works	Clients' Flexibility		Stations Location		Fleet-sizing	
	Spatial	Temporal	NP-Hard	P **cases**	NP-Hard	P **cases**
Correia et al. [2014]	×					
Lam et al. [2014]			×			
Albinski [2015]					×	
Böhmová et al. [2016]					×	×
Hara and Hato [2017]					×	×
Du et al. [2018]			×	×		
Cocca et al. [2019]	×					
Boyacı and Zografos [2019]	×	×				
Ströhle et al. [2019]	×	×			×	
Brandstätter et al. [2020]			×	×		
Deza et al. [2020]				*[2]		
Schiffer et al. [2021]						×
Liu et al. [2022]				×		×
Zhang et al. [2022]					×	
Lai et al. [2022]	×					
This work	×		×			×

mentioned in the column header. Marks regarding the "Client's Flexibility" topic mean that the work analyzed either the spatial or temporal flexibility of carsharing. In this work, only the spatial flexibility is analyzed due to the weak results found by Boyacı and Zografos [2019] and Ströhle et al. [2019] when assessing the benefits of temporal flexibility.

Marks regarding the "Stations Location" and "Fleet-sizing" topic mean that the work approached at least one problem regarding the topic and defined whether the problem is NP-Hard. Some works approached more than one problem about the same topic, and thus, proposed different algorithms with different time complexities for solving them. When the same work approaches an NP-Hard problem and a polynomial-time algorithm (for different problems) the Table 2.1 presents it with two marks in the same topic, one for NP-Hard and another for "P cases".

In this work, only NP-Hard problems are approached for locating stations to assure good stations' locations to the simulated carsharing system. Since the stations are not expected to change their locations frequently, it is reasonable to spend more time looking for better combinations of stations' locations when it is needed. However, it does not apply for the fleet-sizing task. The demand of each station may change more frequently, requiring a faster method for adjusting the supply of vehicles. Therefore, this work do not deal with a NP-Hard fleet-sizing problem, but proposes a polynomial-time fleet-sizing method.

The next chapter presents the initial results of this work, which provided essential insights for developing the following chapters.

Chapter 3

Insights from Initial Results

This chapter presents the initial and already published results about the simulation of carsharing services in the city of São Paulo, and proposes a carsharing business model that fits the local demands.

Section 3.1 aims to maximize profits by choosing between round-trip or one-way modalities. Section 3.2 compares the computational performance of two MILP formulations for maximizing the profits of a carsharing company offering round-trip services. Section 3.3 approaches the fleet-sizing by maximizing the number of clients served. Next, Section 3.4 explains the proposed low-cost carsharing business models.

The three first sections do not directly tackle the research problem of this thesis. These sections focus on simulating the carsharing service by applying different objective functions and constraints to better comprehend their rental dynamics and define the business model. Besides, the results obtained were useful for building a linear formulation for fleet-sizing presented on the next chapter, at section 4.3, which is suitable for the proposed business model.

3.1 Maximizing Carsharing Profits

This section presents a MILP formulation to optimize the fleet size of one-way and round-trip modes in order to maximize the company's profit and uses this formulation to simulate the carsharing performance on the city of São Paulo[1]. This results were useful to estimate how profitable a carsharing business model can be.

Different scenarios are analyzed for the one-way and round-trip settings, varying service costs, rental prices, number of clients, rental duration and driven distance. The location of stations used in this simulation are real vehicle dealerships in São Paulo. Figure 3.1 presents the location of stations. Black lines in Figure 3.1 indicate city districts with at least one simulated carsharing station. A São Paulo district is the smallest official

[1]This section has been published as Monteiro et al. [2019a].

spatial unit adopted by the local government. The thinner gray lines are the city's road network. The neighborhoods with at least one carsharing station are "Jardim América", "Indianópolis", "Ipiranga", "Moema", "Perdizes", "Santo Amaro", "Vila Cordeiro" and "Vila Mariana".

Figure 3.1: Vehicle dealerships used as carsharing stations

3.1.1 MILP Formulation

The proposed formulation can be used for both round-trip and one-way carsharing modes. Following Nourinejad et al. [2015], it does not split the operating day in time intervals. By doing so, carsharing dynamics are simulated through a continuous time span, representing the flow of clients and vehicles more realistically. The time is represented as events of clients starting and finishing rentals.

Before starting the optimization, clients are generated to simulate the demand for carsharing. In this simulation, clients can be served or not. Each client is assigned to an origin station and to a destination station. When the simulated scenario is based on round-trip, the origin and the destination stations must be the same. On the one-way scenarios, the origin and destination stations may be different. The number of clients starting and ending a rental in each station is proportional to the number daily trips made in the São Paulo's district where the station is located. Thus, the demand is divided throughout the city, simulating more clients in regions with higher mobility occurrence and vice-versa.

Tables 3.1, 3.2 and 3.3 define, respectively, the indices, variables and model parameters of this MILP formulation. Stations are indicated by indices s, r and l. The times when trips have started and ended are represented by the pairs (i, j), and (p, q). Those times were generated within the range 8:00 AM to 8:00 PM, according with the operating hours of dealerships. This formulation considers that clients may not go straight from origin station to his/her destination station following the shortest path. It is possible that some clients prefer doing multiple smaller trips during the same rental, taking the most of the car by making longer rentals. This was considered by randomly generating an additional time spent and driven distance for each rental. Both additional time and distance follow a uniform distribution, and the total time spent and driven distance per rental were limited by parameters set in the evaluated scenarios.

Each trip demand is represented by a binary variable x indexed by four indices: two indicating the time when it has started and finished, and other two indicating the origin and destination stations. By doing so, variable $x_{s,r}^{i,j}$ represents a trip demand from station s at time i, going to station r at time j. This variable will have value 1 whether the client was served and 0 otherwise.

Table 3.1: Indices of the Maximizing Profits MILP

Index	Description
$s, r, k, l \in \mathbb{S}$	Station indicating the trip origin or destination
$(i, j) \in \mathbb{T}$	Time i indicating when trip $x_{s,r}^{i,j}$ started and time j about when it finished

Table 3.2: Variables of the Maximizing Profits MILP

Variable	Description
n_s	Number of vehicles to be allocated in station s
$x_{s,r}^{i,j}$	Trip demand from station s at time i, to station r at time j

Model Parameter $d_{s,r}^{i,j}$ represents the distance driven by trip demand $x_{s,r}^{i,j}$. In this simulation, the rental price is calculated according to two different formulas, each of them used by one carsharing company from São Paulo. Equation 3.1 presents the prices charged by the Turbi[2] company, and Equation 3.2 presents the prices applied by the Zazcar[3] company. The formula used by Turbi charged[4] R$8 per rented hour (represented

[2]https://turbi.com.br/
[3]This company does not exist anymore
[4]Brazilian currency: Reais (R$)

Table 3.3: Model Parameters of the Maximizing Profits MILP

Parameter	Description
$r^{i,j}_{1,s,r}$	Revenue obtained by Turbi for serving trip demand $x^{i,j}_{s,r}$
$r^{i,j}_{2,s,r}$	Revenue obtained by Zazcar for serving trip demand $x^{i,j}_{s,r}$
$d^{i,j}_{s,r}$	Distance driven by client $x^{i,j}_{s,r}$
$c^{i,j}_{1,s,r}$	Cost for serving $x^{i,j}_{s,r}$ with vehicles not exclusive for carsharing
$c^{i,j}_{2,s,r}$	Cost for serving $x^{i,j}_{s,r}$ with vehicles dedicated to carsharing
c^s	Cost for maintaining a vehicle at station s
p_s	Number of parking slots in station s

in Equation 3.1 from the subtraction between the ending time and the starting time), and R\$0.50 per driven kilometer. The formula used by Zazcar charged R\$10 per hour, R\$0.90 per kilometer and has a minimum price per rental of R\$20. These formulas are from March 10, 2019. As a matter of comparison, at that date 1 US Dollar was equivalent to about R\$3.87.

$$r^{i,j}_{1,s,r} = (j - i) + 0.50 d^{i,j}_{s,r} \tag{3.1}$$

$$r^{i,j}_{2,s,r} = \max\left(20, 10(j - i) + 0.90 d^{i,j}_{s,r}\right) \tag{3.2}$$

Trip costs are therefore calculated using the distance driven, rental duration and a fixed cost per vehicle. Along with the two pricing models presented in Equation 3.1 and Equation 3.2, two cost models are applied in the simulations. Those cost models are presented in Equation 3.3 and Equation 3.4. In both models, the distance cost is set as the same: R\$0.50 per km. That value was chosen because it is the lowest charged price per distance between the companies' price models, and considering that the gas price in São Paulo on March 10, 2019 was R\$4.144 on average[5]. With that gas price, a vehicle with a consumption of 11 km per liter would spend about R\$0.35 per km for the fuel costs only. In this simulation, the remaining R\$0.15 per km is associated to vehicle maintenance costs.

$$c^{s,r}_{1,i,j} = 1.1(j - i) + 0.50 d^{i,j}_{s,r} \tag{3.3}$$

$$c^{s,r}_{2,i,j} = 0.50 d^{i,j}_{s,r} \tag{3.4}$$

The rental duration cost and the fixed cost per vehicle are different on the two price models. The cost formula from Equation 3.3 assumes that vehicles are not exclusively used for carsharing. In this scenario, the vehicle cost only applies when the vehicle is rented.

[5] https://precodoscombustiveis.com.br/pt-br/city/brasil/sao-paulo/sao-paulo/3830

When the vehicle is idle, the company can use it for other purposes, not generating costs for the carsharing operation itself. Therefore, instead of defining a fixed value for the vehicle use, this model defines a cost per rented hour. This cost must compensate for vehicle depreciation on the first year of use, including taxes. Vehicle depreciation is about 12% per year da Silva and de Oliveira [2018], and the annual circulation tax is 4%, both in relation to its value[6]. Adding these rates, a cost of 16% per year is then applied to the vehicle's value. Considering a market price of R$30,000, the vehicle's cost per year is about R$4,800. From this value, the cost per day is about R$13 and the cost per hour (considering the 12 working hours per day on the stations) is about R$1.10, represented by the 1.1 in the Equation 3.3. When using this Cost Model, the fixed cost per vehicle c^s is not assessed by the objective function. Equation 3.4 presents Cost Model 2, which does not include a vehicle cost per hour. However, when calculating the objective function for the Cost Model 2, each vehicle has a daily fixed cost c^s = R$13.

The optimization's objective function is presented by Equation 3.5 when using Cost Model 1 and by Equation 3.6 when using Cost Model 2. The objective function aims to maximize the carsharing company's profits by taking the rental revenue, and subtracting from it the sum of costs for serving the trip demands and maintaining the vehicles. Equations from 3.7 to 3.10 define the MILP constraints.

$$\arg\max_{x_{s,r}^{i,j}} \sum_{s,r\in\mathbb{S}} \sum_{i,j\in\mathbb{T}} x_{s,r}^{i,j}(r_{1,s,r}^{i,j} - c_{1,s,r}^{i,j}) \tag{3.5}$$

or

$$\arg\max_{x_{s,r}^{i,j},\ n_s} \sum_{s,r\in\mathbb{S}} \sum_{i,j\in\mathbb{T}} x_{s,r}^{i,j}(r_{2,s,r}^{i,j} - c_{2,s,r}^{i,j}) - \sum_{s\in\mathbb{S}} c_s n_s \tag{3.6}$$

Subject to:

$$x_{s,r}^{i,j} \le n_s + \sum_{k\in\mathbb{S}} \sum_{\substack{p,q\in\mathbb{T}\\ q<i}} x_{k,s}^{p,q} - \sum_{l\in\mathbb{S}} \sum_{\substack{p,q\in\mathbb{T}\\ p<i}} x_{s,l}^{p,q} \qquad \forall i,j\in\mathbb{T},\ \forall s,r\in\mathbb{S} \tag{3.7}$$

$$n_s \le p_s \qquad \forall s\in\mathbb{S} \tag{3.8}$$

$$x_{s,r}^{i,j} \in \{0,1\} \qquad \forall i,j\in\mathbb{T},\ \forall s,r\in\mathbb{S} \tag{3.9}$$

$$n_s \in \mathbb{N}^0 \tag{3.10}$$

Equation 3.7 ensures that a trip demand $x_{s,r}^{i,j}$ will only be served if there is at least one vehicle available at station s. The number of available vehicles at the station s is the original number of vehicles allocated to that station (n_s), added to the sum of other trip demands $x_{k,s}^{p,q}$ that have already rented and delivered a vehicle to that station before trip $x_{s,r}^{i,j}$ has been served. Since it is possible that other trips $x_{s,l}^{p,q}$ used a vehicle from the same station s before the trip demand $x_{s,r}^{i,j}$ was served, vehicles rented by the trip

[6] https://portal.fazenda.sp.gov.br/servicos/ipva/Paginas/mi-aliquota.aspx

demands $x_{s,l}^{p,q}$ are subtracted from the available ones at station s. Equation 3.8 limits the number of allocated vehicles in each station to the number of parking slots in the station. Equation 3.9 defines the trip demand variables as binaries, and Equation 3.10 defines the number of allocated vehicles as an integer and positive number, including zero.

The maximum of vehicles to be simulated is 27, due to limits of the vehicle dealerships considered. Simulations were performed varying the number of clients, the rental duration and the distance traveled. The simulated number of clients were 100 and 300. As the users were generated randomly, their rental durations and driven distances varied inside minimum and maximum intervals. Two levels of duration and driven distance intervals were evaluated for one-way and for round-trip modes. For the one-way mode, the duration intervals were from a minimum of 22 minutes and 30 seconds to a maximum of 2 hours per rental, and twice that: going from 45 minutes to 4 hours per rental. One distance interval for one-way trips started from 2.5 km and ended at 25 km, and the other distance interval is twice that: starting from 5 km and up to 50 km. As the round-trip mode consists in the client having to drive back to the origin station, the expected duration and distance on round-trip should be greater than those expected for one-way. By doing so, the intervals set to the round-trip mode are twice the ones set for one-way mode. The following subsection presents the simulation results using the proposed MILP model and these chosen parameters.

3.1.2 Simulation Results

All simulation results are presented on Table 3.4. Since six different factors were assessed (carsharing mode, number of clients, duration interval, distance interval, cost model, and Price Model) each one with two possible values, Table 3.4 has $2^6 = 64$ lines. The results are sorted in ascending order by daily profit. Besides the profit, the table also shows the optimal number of vehicles to be allocated and the total cost (adding client and vehicle costs) for each scenario.

Figure 3.2 summarizes the scenarios presented in Table 3.4 by indicating them in a graph. It is possible to identify a leap made between scenario 55, with a profit of R$3,075, to scenario 57, with a profit of R$3,743. From scenario 57 onward, every scenario is in the round-trip mode, with 300 clients, long duration and long distance, using Price Model 2 and using all the 27 vehicles.

We observe that the number of clients served is a strong contributor to the profit. All the first 10 scenarios (with less profit) have only 100 clients, while the last 9 scenarios (with more profit) have 300 clients. The carsharing mode also has a big influence on the

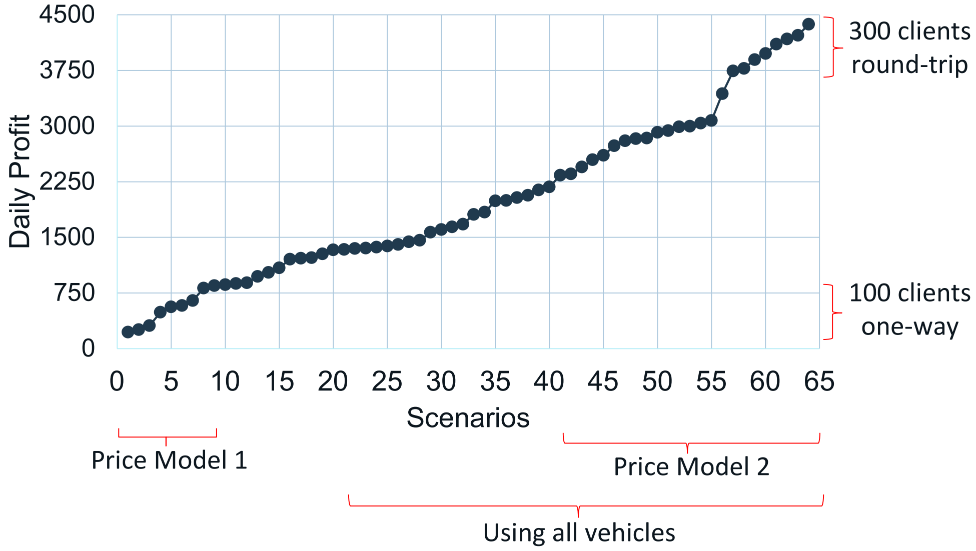


Figure 3.2: Summary of patterns found on the simulated scenarios

profits. However, the highest profits with round-trip correspond to large values for the duration and distance interval parameters. Therefore, a naive analysis comparing only the carsharing modes should be avoided. Nevertheless, all scenarios with one-way and 100 clients are in the first half of the Table 3.4, and all the scenarios with round-trip and 300 clients are in the second half of the Table 3.4. The first 8 scenarios and the last 8 scenarios on the Table 3.4 are uniform as to the carsharing mode and the number of clients. Those scenarios only vary on the cost model, indicating that it does not influence as much as the mode and the number of clients. Even though there is overlapping among the modes and parameters, the extreme profit cases tend to be well separated.

In general, the round-trip mode is more profitable than one-way because the price models assign more value to the rental duration than to the distance driven. Thus, even when a one-way scenario serves more clients than the round-trip, if the one-way vehicles stay idler than the round-trip ones, probably the one-way profits will be lower than the round-trip ones. However, this situation can reverse as the one-way mode gets more clients, causing the vehicles to be more frequently used.

The extreme lower and higher profits are strongly different. The scenario with the smallest profits makes R$223 a day, and the scenario with the highest profits makes R$4,373 a day, about 19.61 times greater than the smallest profit. The cost and price models cause a big difference in profits. The first 8 scenarios used Price Model 1, and the last 24 scenarios used Price Model 2. Although the Cost Models 1 and 2 appeared throughout the table, all the scenarios with less than 27 vehicles appeared only with Cost Model 2. That happened because, in Cost Model 2, the vehicles are considered to be exclusively used for carsharing. Therefore, if some of the 27 vehicles will not be used at least enough to cover their fixed cost, it is more profitable to remove those idle vehicles from the carsharing operation than to keep them, else costs outweighs profits.

All the scenarios that did not use all the 27 vehicles appear among the first 20 scenarios in Table 3.4. Most of them use the one-way mode, have only 100 clients, trips have short durations and involve short distances, and use Price Model 1, as expected, since there would be idle vehicles.

Table 3.4 is presented in the following pages. Columns about number of clients, number of needed vehicles, cost and profit are colored to facilitate the comparison of values. The more clients are simulated, the greener the cell color is. The same pattern is applied on the profit column. However, columns regarding needed vehicles and costs involved have an inverse pattern of colors: the lesser the number, the greener the cell is. This difference of color pattern was applied to illustrate that costs (including the investment of acquiring vehicles) are usually avoided by a for-profit company.

Although these evaluated scenarios provide insightful analysis about the managerial dynamics of a carsharing service, these results neither assure that this MILP will be solved in tractable time for large-scale instances, nor define the location for carsharing stations. Next section presents the computation performance comparison of two round-trip fleet-sizing MILP formulations.

Table 3.4: Simulation of carsharing profits on São Paulo, sorted by daily profit

#	Mode	Clients	Duration	Distance (km)	Cost Model	Price Model	Needed Vehicles	Cost (R$)	Profit (R$)
1	one-way	100	22m:30s - 2h	5 - 50	2	1	15	601	223
2	one-way	100	22m:30s - 2h	2.5 - 25	2	1	14	510	256
3	one-way	100	45m - 4h	2.5 - 25	2	1	16	552	309
4	one-way	100	45m - 4h	5 - 50	2	1	21	818	490
5	one-way	100	22m:30s - 2h	5 - 50	1	1	27	642	563
6	one-way	100	22m:30s - 2h	2.5 - 25	1	1	27	537	580
7	one-way	100	45m - 4h	2.5 - 25	1	1	27	566	647
8	one-way	100	45m - 4h	5 - 50	1	1	27	853	815
9	one-way	100	22m:30s - 2h	2.5 - 25	2	2	20	597	849
10	one-way	100	22m:30s - 2h	5 - 50	2	2	23	686	863
11	one-way	300	22m:30s - 2h	2.5 - 25	2	1	27	1,416	878
12	one-way	300	22m:30s - 2h	5 - 50	2	1	27	1,671	888
13	one-way	100	45m - 4h	2.5 - 25	2	2	22	611	974
14	one-way	300	45m - 4h	2.5 - 25	2	1	27	1,359	1,026
15	round-trip	100	45m - 4h	5 - 50	2	1	27	1,257	1,090
16	round-trip	100	45m - 4h	10 - 100	2	1	26	1,608	1,204
17	one-way	300	45m - 4h	5 - 50	2	1	27	1,821	1,217
18	one-way	100	22m:30s - 2h	2.5 - 25	1	2	27	539	1,226
19	one-way	100	22m:30s - 2h	5 - 50	1	2	27	634	1,278
20	one-way	100	45m - 4h	5 - 50	2	2	25	874	1,331
21	round-trip	100	45m - 4h	5 - 50	1	1	27	1,350	1,337
22	one-way	300	22m:30s - 2h	2.5 - 25	1	1	27	1,300	1,350
23	one-way	100	45m - 4h	2.5 - 25	1	2	27	555	1,354

#	Mode	Clients	Duration	Distance (km)	Cost Model	Price Model	Needed Vehicles	Cost (R$)	Profit (R$)
24	one-way	300	22m:30s - 2h	5 - 50	1	1	27	1,549	1,367
25	round-trip	100	1h:30m - 8h	10 - 100	2	1	27	1,363	1,382
26	round-trip	100	1h:30m - 8h	5 - 50	2	1	27	1,257	1,404
27	round-trip	100	45m - 4h	10 - 100	1	1	27	1,739	1,439
28	one-way	300	45m - 4h	2.5 - 25	1	1	27	1,282	1,459
29	round-trip	100	1h:30m - 8h	10 - 100	1	1	27	1,527	1,568
30	round-trip	100	1h:30m - 8h	5 - 50	1	1	27	1,408	1,604
31	one-way	300	45m - 4h	5 - 50	1	1	27	1,756	1,640
32	one-way	100	45m - 4h	5 - 50	1	2	27	851	1,677
33	round-trip	300	45m - 4h	10 - 100	2	1	27	2,292	1,809
34	round-trip	300	45m - 4h	5 - 50	2	1	27	1,908	1,840
35	round-trip	300	1h:30m - 8h	5 - 50	2	1	27	1,629	1,992
36	round-trip	300	45m - 4h	10 - 100	1	1	27	2,469	1,998
37	round-trip	300	45m - 4h	5 - 50	1	1	27	2,045	2,035
38	round-trip	300	1h:30m - 8h	10 - 100	2	1	27	1,891	2,067
39	round-trip	300	1h:30m - 8h	5 - 50	1	1	27	1,835	2,138
40	round-trip	300	1h:30m - 8h	10 - 100	1	1	27	2,123	2,179
41	one-way	300	22m:30s - 2h	2.5 - 25	2	2	27	1,442	2,337
42	round-trip	100	45m - 4h	5 - 50	2	2	27	1,286	2,355
43	one-way	300	45m - 4h	2.5 - 25	2	2	27	1,372	2,447
44	one-way	300	22m:30s - 2h	5 - 50	2	2	27	1,739	2,546
45	round-trip	100	45m - 4h	5 - 50	1	2	27	1,392	2,605
46	round-trip	100	1h:30m - 8h	5 - 50	2	2	27	1,286	2,734
47	round-trip	100	1h:30m - 8h	10 - 100	2	2	27	1,388	2,801

#	Mode	Clients	Duration	Distance (km)	Cost Model	Price Model	Needed Vehicles	Cost (R$)	Profit (R$)
48	round-trip	100	45m - 4h	10 - 100	2	2	27	1,743	2,831
49	one-way	300	22m:30s - 2h	2.5 - 25	1	2	27	1,282	2,838
50	one-way	300	45m - 4h	2.5 - 25	1	2	27	1,256	2,916
51	round-trip	100	1h:30m - 8h	5 - 50	1	2	27	1,446	2,938
52	round-trip	100	1h:30m - 8h	10 - 100	1	2	27	1,551	2,989
53	one-way	300	45m - 4h	5 - 50	2	2	27	1,824	2,997
54	one-way	300	22m:30s - 2h	5 - 50	1	2	27	1,588	3,040
55	round-trip	100	45m - 4h	10 - 100	1	2	27	1,851	3,075
56	one-way	300	45m - 4h	5 - 50	1	2	27	1,727	3,437
57	round-trip	300	1h:30m - 8h	5 - 50	2	2	27	1,708	3,743
58	round-trip	300	45m - 4h	5 - 50	2	2	27	2,016	3,776
59	round-trip	300	1h:30m - 8h	5 - 50	1	2	27	1,919	3,896
60	round-trip	300	45m - 4h	5 - 50	1	2	27	2,173	3,977
61	round-trip	300	1h:30m - 8h	10 - 100	2	2	27	2,030	4,104
62	round-trip	300	45m - 4h	10 - 100	2	2	27	2,570	4,174
63	round-trip	300	1h:30m - 8h	10 - 100	1	2	27	2,290	4,224
64	round-trip	300	45m - 4h	10 - 100	1	2	27	2,725	4,373

3.2 Computational Performance

This section presents two MILP formulations for simulating the round-trip carsharing profits, compares their computational performance and describes the optimal solution found[7]. Both formulations aim to solve the same problem and were evaluated under the same data, and therefore, generate the same global optimal results. The analysis made in this section were useful to understand how large are the instances that can be solved using the computational resources available for this PhD thesis.

Figure 3.3 presents the location of 100 stations used in the experiments. Differently from the last section, these stations were randomly generated onto positions of street lengths in São Paulo. Therefore, regions with larger total street length are more likely to receive a carsharing station. That procedure also avoids locating stations on regions with only water, woods or no driving access.

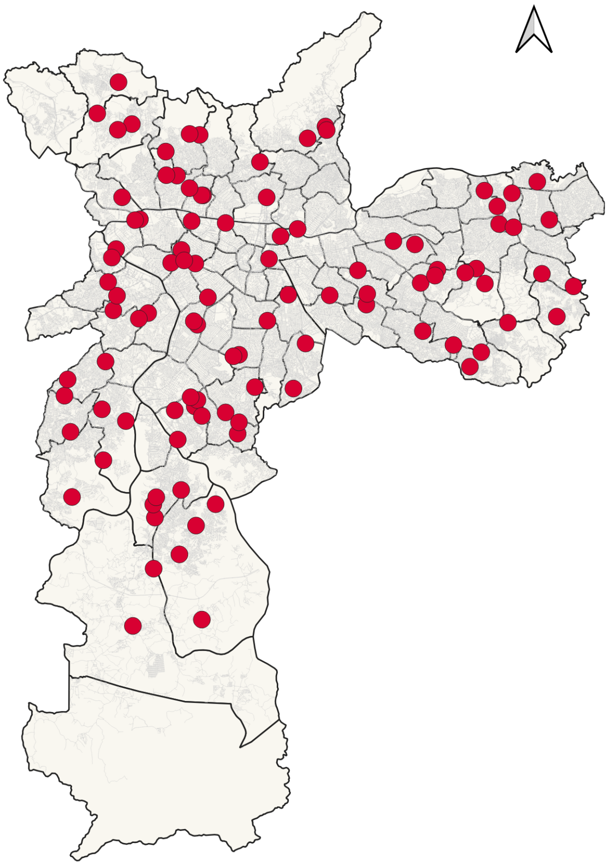


Figure 3.3: Generated locations for the carsharing stations

As in the former section, the number of generated clients varies according to the number of trips made in the district where the station is located. Thus, there are more simulated trips in regions with more demand for trips. This simulation also considers that

[7]This section has been published as Monteiro et al. [2019b]

clients may not go straight from origin station to his/her destination station following the shortest path. The possibility of clients doing multiple smaller trips during the same rental was also included by randomly generating an additional time spent and driven distance for each rental. Both additional time and distance also follow a uniform distribution. The total rental duration varied from 45 minutes to four hours, and the total driven distance varied from five kilometers to 50 kilometers.

Subsection 3.2.1 describes the formulation based on the Big-M method. The Big-M method consists in defining big enough constants and multiply them by specific variables on the objective function or constraints in order to ensure the feasibility of some solutions [Camm et al., 1990; Bazaraa et al., 2010; Klotz and Newman, 2013]. The formulation presented in the following subsection uses the Big-M method to guarantee that earlier clients arriving at the stations will have priority on being served.

3.2.1 MILP Formulation with Big-M

The indices, variables and model parameters used in this formulation are presented, respectively, by Tables 3.5, 3.6 and 3.7.

Table 3.5: Indices of the Formulation with Big-M

Index	Description
$s \in \mathbb{S}$	Station s indicating the trip origin of $x_s^{i,j}$
$(i,j) \in \mathbb{T}$	Time i indicating when trip $x_s^{i,j}$ started and time j about when it finished

Table 3.6: Variables of the Formulation with Big-M

Variable	Description
n_s	Number of vehicles to be allocated in station s
$x_s^{i,j}$	Trip demand from station s at time i to the same station s at time j

Revenue and costs are calculated using the Price Model 2 and the Cost Model 2 presented in the last section. By doing so, vehicles are assumed to be exclusively used for carsharing and the prices are the same as those the Zazcar company used to charge. The revenue from each rental is determined at[8] R\$10 per hour plus R\$0.90 per driven kilometer, with a minimum fare of R\$20.

[8] Brazilian currency: Reais (R\$). In comparison, the exchange rate on August 13, 2019, was of R\$3.96 per US dollar

Table 3.7: Model Parameters of the Formulation with Big-M

Parameter	Description
$r_s^{i,j}$	Revenue obtained for serving trip demand $x_s^{i,j}$
$d_s^{i,j}$	Distance driven on trip demand $x_s^{i,j}$
$c_s^{i,j}$	Cost yielded by trip demand $x_s^{i,j}$
c_s	Cost for maintaining a vehicle at station s
p_s	Number of parking slots in station s
$M_s^{i,j}$	"Big-M" used as maximum number of trips that station s can serve

Equation 3.11 defines the revenue $r_s^{i,j}$. Since this section only considers round-trip carsharing, the trip demand $x_s^{i,j}$ is indicated by only one station index s. The cost $c_s^{i,j}$ is calculated as R\$0.90 per driven kilometer, and cost c_s is defined as R\$13 per day and per vehicle, indicating the vehicle's depreciation along the time of use.

$$r_s^{i,j} = \max\Big(20, 10(j-i) + 0.90 d_{x_s^{i,j}}\Big) \tag{3.11}$$

Equation 3.12 presents the objective function, with the goal to maximize the difference between total revenue and total cost, generating the profits. The constraints are expressed from Inequality 3.13 to Equation 3.17.

$$\underset{x_s^{i,j},\ n_s}{\arg\max} \sum_{s \in \mathbb{S}} \sum_{i,j \in \mathbb{T}} x_s^{i,j}(r_s^{i,j} - c_s^{i,j}) - \sum_{s \in \mathbb{S}} c_s n_s \tag{3.12}$$

Subject to:

$$x_s^{i,j} \leq n_s + \sum_{\substack{p,q \in \mathbb{T} \\ q<i}} x_s^{p,q} - \sum_{\substack{m,n \in \mathbb{T} \\ m<i}} x_s^{m,n} \qquad \forall i,j \in \mathbb{T},\ \forall s \in \mathbb{S} \tag{3.13}$$

$$M_s^{i,j} x_s^{i,j} \geq n_s + \sum_{\substack{p,q \in \mathbb{T} \\ q<i}} x_s^{p,q} - \sum_{\substack{m,n \in \mathbb{T} \\ m<i}} x_s^{m,n} \qquad \forall i,j \in \mathbb{T},\ \forall s \in \mathbb{S} \tag{3.14}$$

$$n_s \leq p_s \qquad \forall s \in \mathbb{S} \tag{3.15}$$

$$x_s^{i,j} \in \{0,1\} \qquad \forall i,j \in \mathbb{T},\ \forall s \in \mathbb{S} \tag{3.16}$$

$$n_s \in \mathbb{N}^0 \tag{3.17}$$

Inequality 3.13 limits trip demand $x_s^{i,j}$ to only be served if there is at least one available vehicle. Inequality 3.14 ensures that trip demand $x_s^{i,j}$ will be served if there is at least one available vehicle. This inequality guarantees that the carsharing company will not refuse to serve a client in order to "reserve" a vehicle for another client that would be more profitable for the company. By doing so, this formulation maintains a property of first-come-first-serve. Inequality 3.15 limits the number of vehicles to be allocated in station s to the number of existent parking slots p_s. Inequality 3.16 defines the trip's variables as binaries. Finally, constraint 3.17 defines variables n_s as positive integers including zero.

The Big-M method applied on Inequality 3.14 is important for balancing both sides of that inequality. If Big-M was not used, the binary variable $x_s^{i,j}$ would also limit the inequality's right hand side to one. The Big-M multiplying the $x_s^{i,j}$ makes the left hand side have a value greater than one when variable $x_s^{i,j}$ is equal to one, and makes the left hand side equal to zero when variable $x_s^{i,j}$ is zero.

Although this formulation is relatively short and simple, constraint 3.14 can reduce the computational performance of the formulation. That happens because the Big-M method may cause numerical errors or increase the gap between the linear relaxation and an optimal integral solution [Camm et al., 1990; Klotz and Newman, 2013]. Next section presents an alternative version of this formulation avoiding the use of the Big-M method.

3.2.2 MILP Formulation without Big-M

Avoiding to use the Big-M implies the need of creating additional variables and constraints. Table 3.8 defines the indices, Table 3.9 presents the variables and Table 3.10 defines the model parameters used in this formulation.

Table 3.8: Indices of the Formulation without Big-M

Index	Description
$s \in \mathbb{S}$	Station s indicating the trip origin of $x_s^{i,j}$
$(i,j) \in \mathbb{T}$	Time i indicating when trip $x_s^{i,j}$ started and time j about when it finished
$(i,s) \in \mathbb{V}$	Vehicle of station s indexed by the time i of its first trip
$i-1$	Starting time of the trip demand just before i
$j-1$	Ending time of the trip demand just before j
$i+1$	Starting time of the trip demand just after i

Table 3.9: Variables of the Formulation without Big-M

Variable	Description
$x_s^{i,j}$	Trip demand from station s at time i to the same station at time j
v_s^i	Vehicle of station s. It is indexed by the start time i of its first trip

The change in variables consists in splitting the number of allocated vehicles in each station n_s into several binary variables v_s^i, one for each possible vehicle. Therefore,

Table 3.10: Model Parameters of the Formulation without Big-M

Parameter	Description
$r_s^{i,j}$	Revenue obtained for serving trip $x_s^{i,j}$
$d_s^{i,j}$	Distance driven by trip demand $x_s^{i,j}$
c_s	Cost for maintaining a vehicle
$c_s^{i,j}$	Cost yielded by trip $x_s^{i,j}$

one v_s^i is defined for each parking slot at station $s \in \mathbb{S}$. This change allows constraints relating the vehicle variables directly to the trip demand variables, since now both are binaries. The variables v_s^i are indexed by the same time i used by the first trip that vehicle serves. By doing so, it possible to assure the property of first-come-first-serve of the first trips by constraining their variables directly to the station vehicles. After all first trips are served, the other trips are decided to be served or not according with the vehicles being delivered back to the station.

Equation 3.18 presents the objective function, whose rationale was kept the same as in Equation 3.12. Constraint 3.19 ensures that the first trips on station s will be served by the allocated vehicles in that station. Constraint 3.20 forces the vehicles' variables to maintain an order of priority. Since each vehicle variable is associated to a trip demand, it will assure that the vehicle for an initial trip will not be ignored for saving resources to a later trip. Constraint 3.21 limits trip $x_s^{i,j}$ to be served only if there is at least one vehicle available after considering the trips already made. Constraint 3.22 ensures that a trip with no "reserved" vehicle $v_s^{i,j}$ will only be served if its earlier trip was served or another trip was just finished, yielding an available vehicle to the same station. Constraint 3.23 defines the trip variables as binaries. Finally, constraint 3.24 defines the vehicle variables as binaries.

$$\underset{x_s^{i,j},\ v_s^i}{\arg\max} \sum_{s \in \mathbb{S}} \sum_{i,j \in \mathbb{T}} x_s^{i,j}(r_s^{i,j} - c_s^{i,j}) - c_s \sum_{i,s \in \mathbb{V}} v_s^i \tag{3.18}$$

Subject to:

$$v_s^i \leq x_s^{i,j} \qquad \forall (i,s) \in \mathbb{V} \tag{3.19}$$

$$v_s^{i+1} \leq v_s^i \qquad \forall (i,s) \in \mathbb{V} \mid \exists i+1 \tag{3.20}$$

$$x_s^{i,j} \leq \sum_{i,s \in \mathbb{V}} v_s^i + \sum_{\substack{p,q \in \mathbb{T} \\ q<i}} x_s^{p,q} - \sum_{\substack{m,n \in \mathbb{T} \\ m<i}} x_s^{m,n} \qquad \forall i,j \in \mathbb{T},\ \forall s \in \mathbb{S} \mid (i,s) \notin \mathbb{V} \tag{3.21}$$

$$x_s^{i,j} \leq x_s^{i-1,j-1} + \sum_{\substack{p,q \in \mathbb{T} \\ q<i \\ q>j-1}} x_s^{p,q} \qquad \forall i,j \in \mathbb{T},\ \forall s \in \mathbb{S} \mid (i,s) \notin \mathbb{V} \tag{3.22}$$

$$x_s^{i,j} \in \{0,1\} \qquad \forall i,j \in \mathbb{T}, \forall s \in \mathbb{S} \tag{3.23}$$

$$v_s^i \in \{0, 1\} \qquad \forall (i, s) \in \mathbb{V} \tag{3.24}$$

Both formulations were experimentally run using the previously described data for the city of São Paulo. Next section presents the experimental results.

3.2.3 Simulation Results

This section presents the experimental results of run time and number of served trips, number of vehicles needed and profits that a carsharing company would earn. The evaluated scenarios have 1,000, 2,000, 4,000, 8,000, 16,000, 32,000 and 64,000 trips. The maximum number of vehicles and parking slots simulated were 1,000 and 5,000.

The simulations were performed on a Mac Mini Server (Late 2012) with S.O. MacOS Mojave 10.14.6, processor Intel Core i7 2.3 GHz, and RAM of 16 GB. The models were implemented using Python 3.7, with the wrapper PuLP[9] version 1.6.0 and the solver CBC[10] version 2.10.0. No swap operations between main and secondary memories were needed.

Figure 3.4 presents boxplots of the optimization run times for all scenarios. A maximum time limit of 30 minutes per run was set, however, even with this time limit the solver spent more than 30 minutes in some runs. Each boxplot represents 40 runs for each evaluated scenario. The axis "Time (seconds)" is shown in logarithmic scale to make the visual comparison easier. Boxplots in red were simulated using the proposed formulation with Big-M and with at maximum 1,000 vehicles. Boxplots in blue and green use the proposed formulation without the Big-M method; blue shows results for a fleet of 1,000 vehicles, and green corresponds to 5,000 vehicles available.

Even in logarithmic scale, the boxes representing 50% of the data (between the first quartile, Q_1, and the third quartile, Q_3) cannot be seen in Figure 3.4 for the blue and green boxplots. However, the red boxplots (regarding the formulations with the Big-M method) usually showed that variation more clearly. That pattern indicates that run times vary more widely in the formulation with Big-M. That higher variation can be verified in Tables 3.11 and 3.12. Besides, the simulations with Big-M and 8,000 trips exceeded the time limit of 30 minutes in some runs. All scenarios with Big-M and more than 8,000 trips also exceeded that time limit. In those cases, the solution obtained is not guaranteed to be the optimal.

All the scenarios using the formulation without Big-M (blue and green boxplots) were solved with optimality guarantee. Among scenarios with the same number of trips

[9] https://pythonhosted.org/PuLP/

[10] https://projects.coin-or.org/Cbc

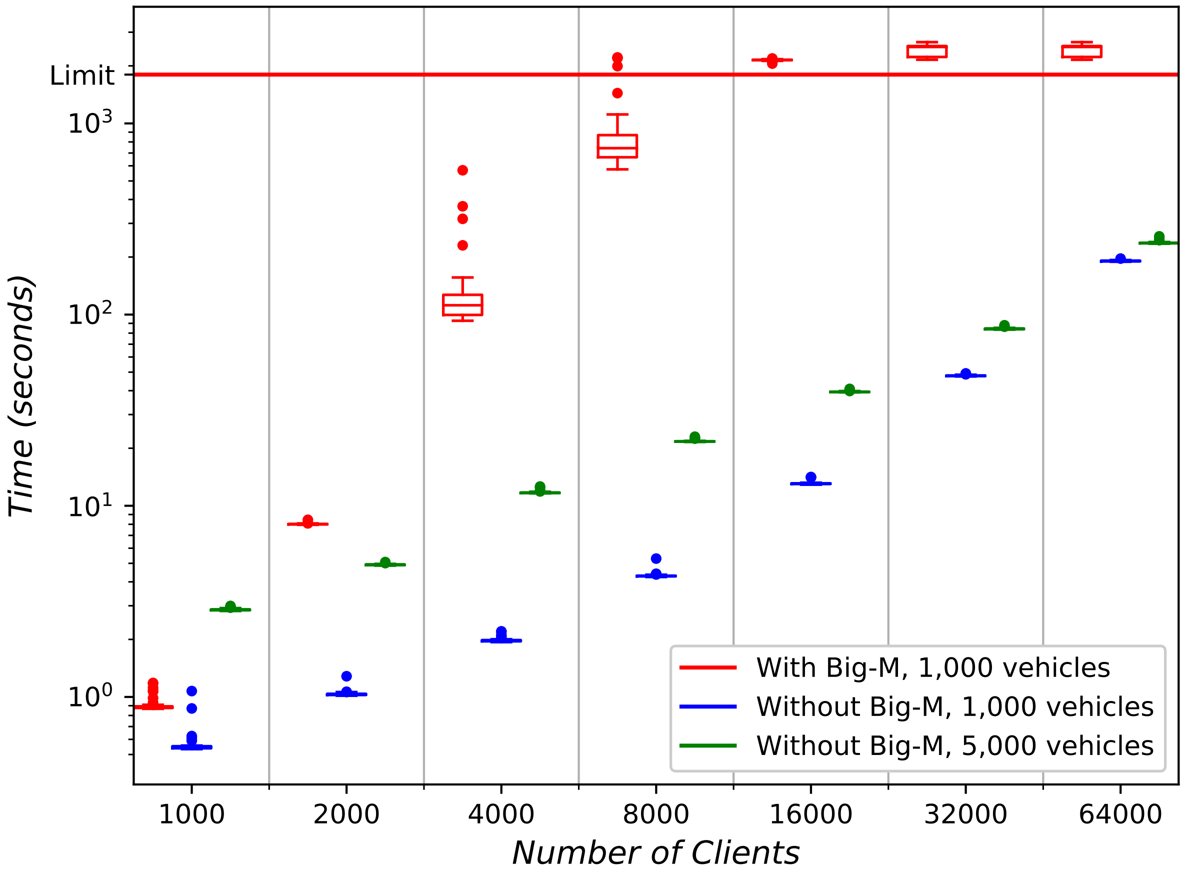


Figure 3.4: Time spent by the evaluated formulations

and except for the outliers (dots above the boxplots), no boxplot shared the same range of values. Therefore, there is significant difference between the run time of all the scenarios evaluated [Krzywinski and Altman, 2014]. Thus, it can be asserted that the formulation without Big-M achieves faster run times than the formulation with Big-M. Besides, starting from 2,000 trips, the formulation without Big-M but with 5,000 vehicles is even faster than the formulation with Big-M but only with 1,000 vehicles.

Tables 3.11 and 3.12 present the basic statistics for the simulations. In both tables, the symbol $M_s^{i,j}$ indicates results regarding the formulation with Big-M, and the symbol $\mathbb{V}$ indicates results from the formulation without Big-M. As shown by Table 3.11, the standard deviation for the scenarios with 4,000 and 8,000 trips raised quickly, when compared to the standard deviation from other scenarios. This higher variation is expected because formulations with Big-M usually yield a larger gap between the linear relaxation and the optimal integral solution; and because the used MILP solver applies a non-deterministic algorithm. Therefore, the time spent by reducing the gap between the linear relaxation and the optimal integral solution is expected to vary more. That difference was strongly reduced in Table 3.12, probably due to the time limit imposed.

Table 3.13 compares the run times of the scenario with 5,000 vehicles (green boxplots), to the run times from the scenario with 1,000 vehicles and also without the use

Table 3.11: Time Spent by Running the Optimization for Low-Demand (seconds)

	Number of Trips							
	1,000		2,000		4,000		8,000	
Statistics	$M_s^{i,j}$	$\mathbb{V}$	$M_s^{i,j}$	$\mathbb{V}$	$M_s^{i,j}$	$\mathbb{V}$	$M_s^{i,j}$	$\mathbb{V}$
Minimum	0.87	0.53	7.95	1.02	92.84	1.94	574.96	4.24
Q_1	0.88	0.54	7.99	1.02	99.60	1.96	665.10	4.27
Median	0.88	0.54	8.00	1.03	111.90	1.97	742.23	4.29
Mean	0.91	0.57	8.02	1.04	138.45	1.99	905.03	4.32
Q_3	0.89	0.55	8.03	1.04	126.90	1.98	868.45	4.31
Maximum	1.18	1.07	8.45	1.28	568.70	2.20	2212.52	5.29
Standard Deviation	0.08	0.10	0.08	0.04	88.40	0.06	446.04	0.16

Table 3.12: Time Spent by Running the Optimization for High-Demand (seconds)

	Number of Trips					
	16,000		32,000		64,000	
Statistics	$M_s^{i,j}$	$\mathbb{V}$	$M_s^{i,j}$	$\mathbb{V}$	$M_s^{i,j}$	$\mathbb{V}$
Minimum	2055.03	12.90	2153.38	47.48	2124.38	188.99
Q_1	2141.20	13.00	2226.37	47.71	2207.79	189.74
Median	2146.41	13.05	2511.32	47.79	2519.39	190.40
Mean	2144.21	13.12	2433.02	47.93	2489.02	190.76
Q_3	2152.84	13.10	2538.97	48.03	2599.07	191.55
Maximum	2186.84	14.14	2661.79	49.22	2735.89	196.31
Standard Deviation	22.71	0.29	150.40	0.43	189.35	1.40

of Big-M (blue boxplots) for low-demand scenarios. Since 5,000 vehicles is 5 times 1,000 vehicles, it was expected that the rate of run time would be about 5 times longer. That proportional response can be observed up to the scenario with 8,000 trips. After that, the optimization with up to 5,000 vehicles started to be not so much slower than the optimization with up to 1,000 vehicles. These results are presented on Table 3.14.

Tables 3.15 and 3.16 compare the optimal solutions found. The numbers of served trips, earned profits and used vehicles used tended to increase together in similar rates through the scenarios. However, that increase seemed to saturate in the scenarios with a high-demand of trips. Up to the scenario with 8,000 trips, as the demand doubled, the percentage of increase more than doubled. But starting from the demand of 16,000 trips, as the demand doubles, the percentage of increase did not change significantly. That saturation indicates that more than 5,000 vehicles and parking slots are needed to significantly raise profits and increase the number of trips served when the demand is of at least 16,000 trips. However, it is possible that only offering the round-trip modality

Table 3.13: Time Comparison Varying to 5,000 Vehicles for Low-Demand (seconds and proportion)

	Number of Trips							
	1,000		2,000		4,000		8,000	
Statistics	Time	Rate	Time	Rate	Time	Rate	Time	Rate
Minimum	2.82	5.28	4.87	4.79	11.61	6.00	21.57	5.09
Q_1	2.84	5.27	4.89	4.78	11.66	5.96	21.66	5.07
Median	2.85	5.24	4.91	4.77	11.69	5.95	21.70	5.06
Mean	2.87	5.01	4.92	4.74	11.74	5.92	21.79	5.04
Q_3	2.87	5.20	4.94	4.75	11.73	5.92	21.75	5.05
Maximum	2.99	2.79	5.08	3.96	12.61	5.72	22.98	4.34
Standard Deviation	0.04	0.04	0.04	0.03	0.20	0.09	0.30	0.06

Table 3.14: Time Comparison Varying to 5,000 Vehicles for High-Demand (seconds and proportion)

	Number of Trips					
	16,000		32,000		64,000	
Statistics	Time	Rate	Time	Rate	Time	Rate
Minimum	39.19	3.04	83.46	1.76	234.89	1.24
Q_1	39.30	3.02	83.79	1.76	235.95	1.24
Median	39.40	3.02	84.15	1.76	236.56	1.24
Mean	39.53	3.01	84.31	1.76	237.70	1.25
Q_3	39.51	3.02	84.62	1.76	237.46	1.24
Maximum	40.87	2.89	87.94	1.79	256.58	1.31
Standard Deviation	0.37	0.03	0.85	0.02	4.24	0.02

would not attract a high demand of trips in all days, maybe making it worthy for the carsharing company to also offer less restrictive modalities.

Next section presents a proposed MILP formulation and results for optimizing the fleet size of a carsharing company that also offers one-way modality and considers that clients may be willing to walk to nearby stations to get an available vehicle or parking slot.

Table 3.15: Optimal Solutions for Low-Demand

	Number of Trips							
	1,000		2,000		4,000		8,000	
Number of Vehicles	1,000	5,000	1,000	5,000	1,000	5,000	1,000	5,000
Served Trips	931	1,000	1,518	2,000	2,053	3,912	2,388	6,883
Increase in Served	7.41%		31.75%		90.55%		188.23%	
Profit (R$)	79,308	84,661	133,877	172,360	182,993	338,187	217,231	605,970
Increase in Profits	6.75%		28.75%		84.81%		178.95%	
Needed Vehicles	543	600	779	1,093	907	1,971	972	3,251
Increase in Vehicles	10.50%		40.31%		117.31%		234.47%	

Table 3.16: Optimal Solutions for High-Demand

	Number of Trips					
	16,000		32,000		64,000	
Number of Vehicles	1,000	5,000	1,000	5,000	1,000	5,000
Served Trips	2,596	9,972	2,664	11,812	2,729	12,897
Increase in Served	284.13%		343.39%		372.59%	
Profit (R$)	237,779	886,743	248,519	1,067,651	254,744	1,189,949
Increase in Profits	272.93%		329.61%		367.12%	
Needed Vehicles	987	4,205	989	4,685	999	4,915
Increase in Vehicles	326.04%		373.71%		391.99%	

3.3 Maximizing Clients Served

This section presents the carsharing simulations considering the client's flexibility to walk to nearby stations and imposing operational constraints that emerge when offering a one-way carshaing service[11]. Simulated scenarios and parameters were drawn from real carsharing company and vehicle dealerships from the city of São Paulo, Brazil. Since those companies neither work with electric vehicles nor with free-floating carsharing, constraints regarding charging vehicle batteries and street parking availability are not included in this formulation.

Besides, it was assumed that the vehicle dealership stores will work as stations for this evaluated carsharing service, which will operate within the dealerships working hours, i.e. from 8 a.m. to 6 p.m.. Vehicle maintenance tasks are expected to happen after the dealerships opening hours, when the carsharing service is not working anymore at that day. Therefore, maintenance tasks were not included in the formulation. Further

[11]This section has been published as Monteiro et al. [2021a]

explanations about the data and evaluated scenarios are presented in the Section 3.3.2. The next subsection presents the MILP formulation.

3.3.1 MILP Formulation

The indices, variables and model parameters used in this formulation are presented, respectively, by Tables 3.17, 3.18 and 3.19.

Table 3.17: Indices of the Maximizing Served Clients MILP

Index	Description
$s, r, k, l \in \mathbb{S}$	Station indicating the trip origin or destination
$(i, j) \in \mathbb{T}$	Time i indicating when trip $x_{s,r}^{i,j}$ started and time j about when it finished

Table 3.18: Variables of the Maximizing Served Clients MILP

Variable	Description
n_s	Number of vehicles allocated to station $s \in \mathbb{S}$
γ	Greatest n_s value among all station $s \in \mathbb{S}$
$x_{s,r}^{i,j}$	Client demand from station s at time i to station r at time j
$x_{k,l}^{i,j}$	Replica of $x_{s,r}^{i,j}$ from nearby origin station k to nearby destination station l

Table 3.19: Model Parameters of the Maximizing Served Clients MILP

Parameter	Description
V	Maximum total number of vehicles
W	Weight for γ not to yield a value greater than one
ϵ	Limits the number of vehicles through the Pareto Front
P_s	Number of parking slots in station s
$M_s = P_s$	Big-M with the max. clients the station s can serve at same time
D	Maximum distance clients are flexible to walk
$\delta(s, k)$	Shortest path distance between stations s and k

The MILP model is formulated using the epsilon-constraint programming as in Bruglieri et al. [2018]. The epsilon-constraint method was also applied. It consists in solving the model multiple times but varying an ϵ threshold of one constraint in each time. By doing so, the set of optimal solutions found after varying the constraint can be put together, forming the Pareto Front [Mavrotas, 2009]. In this work, the ϵ threshold was defined for limiting the total number of allocated vehicles in each simulation run. By doing so, every scenario was simulated up to V times (maximum total number of vehicles), varying the threshold ϵ from 1 to V.

This formulation also focuses on minimizing the highest number of vehicles to be allocated to a station in order to avoid fleet size discrepancies among stations. Therefore, the variable γ is used to penalize solutions with high fleet size in the same station. In order to prioritize the optimal solution for maximum number of clients served, the subtracted penalty γ must be smaller than one. Otherwise, the optimized function can avoid serving some clients not to being penalized. To solve that, the subtracted penalty γ is adjusted to always be smaller than one due to the weight W multiplying it. Equation 3.26 defines the W value based on the optimization parameters. The rationale of Equation 3.26 is based on the upper bound for γ, defined by the Inequality 3.25.

Since W multiplied by γ must yield a value lower than one, the greater the γ, the lower the value of W. A naive approach would set W as the lowest fraction allowed by the computer. However, that can lead to numerical errors during the optimization computations. Therefore, a safer way to set a small enough W consists in first defining an upper bound to γ. As γ stands for the greatest number of allocated vehicles in the same station, its value is limited to the highest number of parking slots (P_s) among all stations, as shown in Inequality 3.25.

$$\begin{gathered} \underset{\forall s \,\in\, \mathbb{S}}{max(P_s)} \geq \gamma \\ \\ W\underset{\forall s \,\in\, \mathbb{S}}{max(P_s)} \geq W\gamma \end{gathered} \tag{3.25}$$

Therefore, if W times the highest number of parking slots is equalled to a value smaller than one, for example 0.1, it is possible to define a constant value for W that works as a good multiplier to ensure that the multiplication $W\gamma$ will always be lower than one. Equation 3.26 shows that rationale.

$$\begin{gathered} W\underset{\forall s \,\in\, \mathbb{S}}{max(P_s)} = 0.1 \\ \\ W = \frac{0.1}{\underset{\forall s \,\in\, \mathbb{S}}{max(P_s)}} \end{gathered} \tag{3.26}$$

The whole MILP model is defined as follows. Equation 3.27 presents the objective function, and Inequalities from 3.28 to 3.36 compose the MILP's constraints.

$$\underset{x_{s,r}^{i,j},\ \gamma}{\arg\max} \sum_{s\,\in\,\mathbb{S}} \sum_{r\,\in\,\mathbb{S}} \sum_{i,j\,\in\,\mathbb{T}} x_{s,r}^{i,j} - W\gamma \tag{3.27}$$

Subject to:

$$x_{s,r}^{i,j} + \sum_{\substack{k \in \mathbb{S} \\ k \neq s \\ \delta(k,s) \leq D}} \sum_{\substack{l \in \mathbb{S} \\ l \neq r \\ \delta(l,r) \leq D}} x_{k,l}^{i,j} \leq 1 \qquad \forall i,j \in \mathbb{T},\ \forall s,r \in \mathbb{S} \tag{3.28}$$

$$x_{s,r}^{i,j} \leq n_s + \sum_{k \in \mathbb{S}} \sum_{\substack{p,q \in \mathbb{T} \\ q<i}} x_{k,s}^{p,q} - \sum_{l \in \mathbb{S}} \sum_{\substack{p,q \in \mathbb{T} \\ p<i}} x_{s,l}^{p,q} \qquad \forall i,j \in \mathbb{T},\ \forall s,r \in \mathbb{S} \tag{3.29}$$

$$M_s x_{s,r}^{i,j} \geq n_s + \sum_{k \in \mathbb{S}} \sum_{\substack{p,q \in \mathbb{T} \\ q<i}} x_{k,s}^{p,q} - \sum_{l \in \mathbb{S}} \sum_{\substack{p,q \in \mathbb{T} \\ p<i}} x_{s,l}^{p,q} \qquad \forall i,j \in \mathbb{T},\ \forall s,r \in \mathbb{S} \tag{3.30}$$

$$P_s \geq n_s + \sum_{k \in \mathbb{S}} \sum_{\substack{p,q \in \mathbb{T} \\ q<i}} x_{k,s}^{p,q} - \sum_{l \in \mathbb{S}} \sum_{\substack{p,q \in \mathbb{T} \\ p<i}} x_{s,l}^{p,q} \qquad \forall i,j \in \mathbb{T},\ \forall s,r \in \mathbb{S} \tag{3.31}$$

$$n_s \leq \gamma \qquad \forall s \in \mathbb{S} \tag{3.32}$$

$$\sum_{s \in \mathbb{S}} n_s \leq \epsilon \tag{3.33}$$

$$n_s \leq P_s \qquad \forall s \in \mathbb{S} \tag{3.34}$$

$$x_{s,r}^{i,j} \in \{0,1\} \qquad \forall i,j \in \mathbb{T},\ \forall s,r \in \mathbb{S} \tag{3.35}$$

$$n_s \in \mathbb{N}^0 \tag{3.36}$$

Replicas of client variables were created to simulate scenarios where clients are willing to walk to get an available vehicle or to drive further to find an available parking slot. Each replica represents the original client starting from a different station, or going to a different station. Replicas are also binary variables and work normally as they were trip demands from other client. The only difference is only one trip demand may be served among the original demand and all replicas made from it. That constraint is presented by Inequality 3.28.

Inequalities 3.29 to 3.31 organize the flow of vehicles and clients through the stations. Inequality 3.29 limits client servicing if no vehicles are available in his/her starting station at the time when he/she looked for the service. There is an available vehicle in a station if the difference between the number of clients who earlier delivered a vehicle to that station, and the number of clients who earlier rented a vehicle from that station yields a result greater than zero. Since a station s can start the day with some vehicles already allocated, the variable n_s is also added to Inequality 3.29 to represent the first served clients of the day, before any other rental delivers a vehicle to that station. Therefore, if even after adding the variable n_s to that calculation the result is zero, client x_r^s, in his/her original variable, cannot be served.

However, Inequality 3.29 does not guarantee the cases in which vehicles are available. Even if that calculation results in a value greater than zero, the optimization could prefer not to serve client $x_{s,r}^{i,j}$ in order to serve a next client, who would return the rented vehicle in less time or deliver it to a more demanding station, for example. Aiming to make the optimization more realistic by giving priority to clients who have arrived first and found an available vehicle, the Inequality 3.30 uses the Big-M method to ensure servicing $x_{s,r}^{i,j}$.

By applying the Big-M in Inequality 3.30, if there is at least one available vehicle in starting station s, the variable $x_{s,r}^{i,j}$ will necessarily go up to one due to the $\geq$ inequality. And if there is more than one available vehicle, the binary variable x_r^s will not limit the constraint's right side, since x_r^s multiplied by the Big-M constant M_s will yield a result always greater the constraint's right side. Therefore, the Inequality 3.30 assures the priority to clients who arrived first, without preventing other clients to be served since their station has available vehicles and parking slots. This rationale is similar to the used on MILP formulation with Big-M from the last section.

Inequality 3.31 imposes the issue (2) mentioned in Chapter 2. In this constraint, a client $x_{s,r}^{i,j}$ cannot be served if the destination station r has no available parking slot. Inequality 3.31 has no effect on the round-trip mode, since every vehicle on the round-trip mode starts the day in its parking slot at the starting station and, after each rental, the vehicle must be returned to the same station.

Inequality 3.32 ensures that γ will hold the greatest number of vehicles allocated in the same station. Inequality 3.33 limits the total number of vehicles to the current ϵ value, which varies to generate the Pareto Front solutions. Inequality 3.34 limits the number of vehicles in a station to its number of parking slots. Equation 3.35 defines the trip demands $x_{s,r}^{i,j}$ as binary, and Equation 3.36 defines the variable n_s as integer positive value, including zero.

Next subsection presents how the clients and replicas were generated for this optimization model.

3.3.2 Data and Parameters

This section presents the data and parameters used in the optimization. When the optimization starts, all the client and replica variables are already defined for the solver to optimize using the formulation presented in Section 3.3.1. Table 3.20 shows the simulation's parameters.

The simulations represent one day of carsharing and were performed varying the demand of clients, stations, parking slots and maximum distance walked by the clients to be served. As shown in Table 3.20, the simulated number of clients were 100, 300 and 500. These parameters are consistent with the ones used in Boyacı and Zografos [2019], being

Table 3.20: Simulation's Parameters

Parameter	Value
Clients	100, 300 and 500 per day
Max. walking distance	0m, 250m, 500m, 750m and 1 km
Min. and Max. one-way trip duration	45 min to 4 hours
Min. and Max. round-trip duration	1.5 hours to 8 hours
Number of original parking slots	27
Number of additional parking slots	4
Number of partner parking slots	22 or 66

reasonable for the analysis of a local carsharing service. After client generation, properties such as starting and ending stations were maintained for every simulation. Using the same client setup allows for fairer comparisons as the other parameters vary.

The range between minimum and maximum round-trip duration (from 1.5 hours to 8 hours) was provided by a carsharing company which operates the round-trip mode in São Paulo. Although the company does not set those minimum and maximum durations for rentals, clients commonly follow that duration range since shorter trips will probably be cheaper or easier if made by on-demand ride services (such as from TNC), and longer trips will probably be cheaper if made using a common rental car for a full day.

As the carsharing company does not offer one-way rentals, the expected duration range for one-way in the simulations was based on the round-trip duration range. Since the round-trip mode consists in the client having to drive back to the origin station, the duration on round-trip is expected to be longer than in the one-way mode. In this simulation, one-way rental duration was simulated as half of the round-trip duration: varying from 45 minutes to 4 hours per rental.

The locations of stations are based on real vehicle dealerships in São Paulo, Brazil. The number of clients generated who start or end a rental in each station is proportional to the total number of trips made using an individual mean of transport (such as taxi or private vehicle) in the São Paulo district where the station is located. A São Paulo district is the smallest official spatial unit adopted by the local government, and was the spatial unit used in the dataset from which the demand was drawn. This dataset used is part of the Metrô (Companhia do Metropolitano de São Paulo) origin-destination survey, the same one used by Tomasiello et al. [2020] to generate agents to their simulation. Therefore, the simulated demand was divided throughout the city's districts, simulating more clients in regions with more individual trips and vice versa.

The carsharing company which provided the round-trip duration range reported that the usual carsharing client in São Paulo used the vehicle to go to more than one destination before finishing the rental. Probably this fact is related to the client preferring to use another transportation service instead of carsharing when he/she has only one destination. Therefore, the carsharing demand in São Paulo does not follow conventional patterns of commuting, being expected to neither cause a demand peak exactly when people are going to work nor cause only other demand peak exactly when people are coming back to home. Thus, rental start and end times were generated randomly between the vehicle dealerships opening hours, and the start and end times follow a uniform distribution. An additional time spent by the client going to multiple destinations before finishing the rental was also included in the simulation. The additional times also followed and uniform distribution and the total rental time do not surpassed the duration range set as parameter.

The original parking slots mentioned in Table 3.20 are the same used in Section 3.1, regarding the stations based on real vehicle dealerships from São Paulo. The 27 original parking slots are distributed throughout 11 stations, in which the number of parking slots varies from one to four. Additional parking slots are available at two vehicle dealerships that are not used for carsharing. Each of those parking locations would have 2 additional parking slots. Since the total number of parking slots ($27 + 4 = 31$) is still low for a metropolis such as São Paulo, partner parking slots were also included.

Partner parking slots are alternative stations for the carsharing service, available at commercial establishments such as parking garages, hotels, colleges and shopping malls, selected using a "Location-Allocation" model as proposed in Lage et al. [2019]. The addition of partner parking slots yielded a set of 22 stations. Simulations were performed varying from 1 to 3 parking slots in each partner station (thus, from 22 to 66 parking slots).

Combining the original number of parking slots to the others mentioned in Table 3.20, we used six settings of parking slots:

- Only the original parking slots: total of 27
- Original + additional parking slots: $27 + 4 = 31$
- Original + partner stations with one parking slot available: $27 + 22 * 1 = 49$
- Original + additional parking slots + partner stations with one parking slot: $27 + 4 + 22 * 1 = 53$
- Original + partner stations with three parking slots available: $27 + 22 * 3 = 93$
- Original + additional parking slots + partner stations with three parking slots: $27 + 4 + 22 * 3 = 97$

The Figure 3.5 illustrates the distribution of simulated stations on the involved districts of São Paulo. Original, additional and partner stations are represented, respectively, by red, orange and yellow points. The ten regions indicate the São Paulo districts which have at least one simulated station.

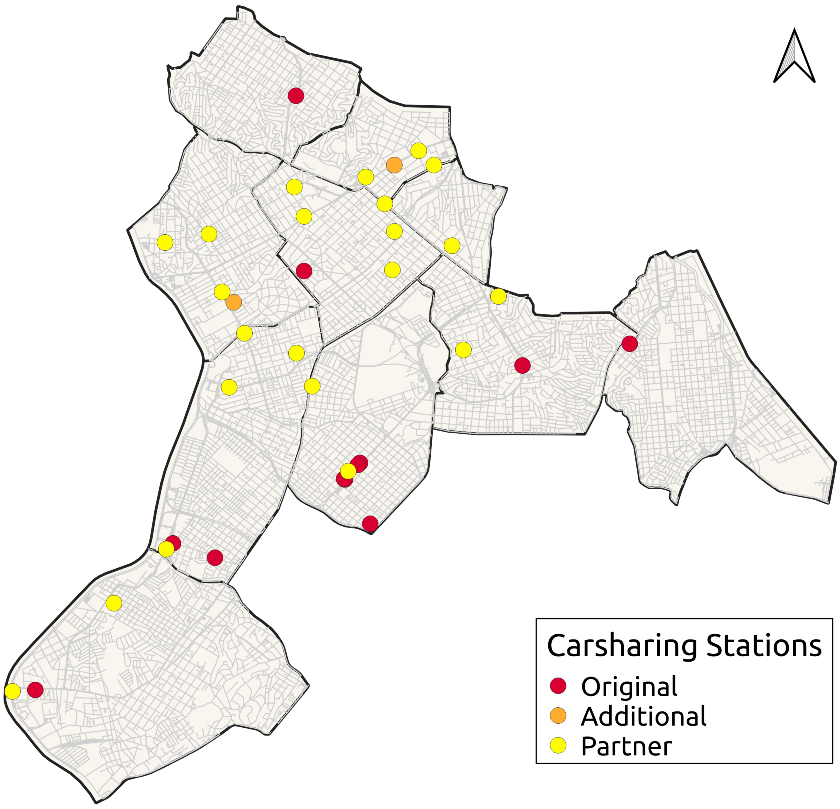


Figure 3.5: Locations of all carsharing stations simulated

Simulations also considered the possibility that clients would be flexible to walk for some distance to reach a viable station. The maximum walking distances used in simulations were 0 meters (where the clients are not flexible to walk), 250 meters, 500 meters, 750 meters and 1 km. As the number of clients, the parameters for walking distance are consistent with the ones used in Boyacı and Zografos [2019]. The walking distance is measured as the shortest distance on the São Paulo road network from the desired station to another station with an available vehicle or parking slot to deliver the shared vehicle. The shortest distance did not consider driving right-of-way constraints, since clients would walk between the stations instead of driving. Next subsection presents experimental results.

3.3.3 Simulation Results

The simulations were performed using the same hardware and software than used and described in subsection 3.2.3. No swap operations between main and secondary memories were needed.

The time limit set for each CBC solver run was 20 minutes. All other parameters follow CBC's defaults. Therefore, the linear objective and constraints were handled using the Simplex algorithm and the integrality of integer variables was dealt using the Branch-and-Cut algorithm. The constraint represented by Equation 3.33 was updated before every CBC run to increase the constant ϵ and to build the Pareto Front. That update was made directly in the PuLP model from the previous solution of the Pareto Front. By doing so, it was possible to avoid reloading the simulated data, and also to avoid rebuilding the constraints for each ϵ value.

Figures 3.6 to 3.13 present results. Each graph shows the variation of the number of served clients in relation to the number of needed vehicles. The number of parking slots is indicated by the color bar. Figure 3.6 shows round-trip scenarios considering that clients would not walk to another station to rent a vehicle. It shows the number of served clients considering a demand of 100, 300 or 500 potential clients, i.e., clients who seek the service, but will not necessarily be served. In this scenario, all results reached guaranteed optimality. In other words, it is assured that the result shown is the best possible combination of vehicles and parking slots to serve as many customers as possible. In this work, the solutions with optimality guarantee are indicated on the legend by "O. G.".

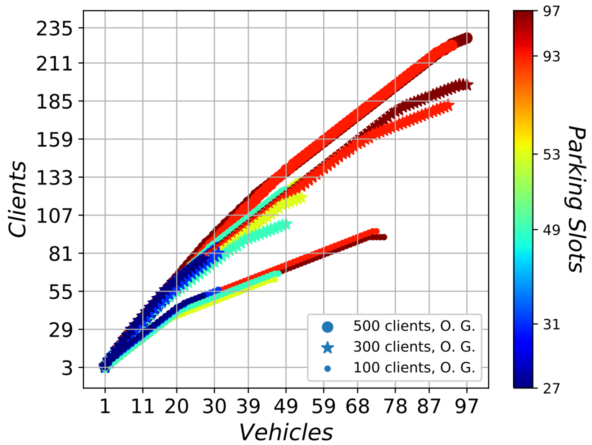


Figure 3.6: Round-trip scenarios with no walking

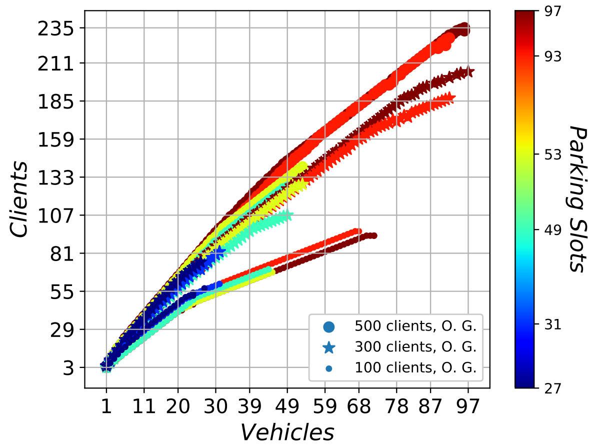


Figure 3.7: Round-trip scenarios with 500m walking

The scenarios with higher client demand present lower saturation while increasing the number of served clients. This indicates that increasing the number of vehicles and the number of available parking slots to more than 100 each can be beneficial. Consider the scenarios with 100 and 500 potential clients, and with 97 available parking slots. The scenario with 500 potential clients can serve 138% more clients than the scenario with 100 clients, using only 27% more vehicles.

Figure 3.7 shows round-trip scenarios where the clients are flexible to walk up to 500 meters between stations while looking for an available vehicle to rent. Results follow the same patterns as in Figure 3.6, and the number of served clients does not increase significantly. The maximum number of clients served increased from 228 to 235, a variation of only seven clients, equivalent to 3.1%. Such small increment probably happened because the round-trip modality does not cause imbalances in the number of vehicles among stations along the day as the one-way modality does. Therefore, there is no big room for improvement by clients walking. The summary of the variation for all walking distances simulated is presented in Table 3.21.

As round-trip clients must deliver the vehicle at the same station from which it was rented, every vehicle on round-trip can have a dedicated parking slot. Thus, the one-way issue (2) mentioned earlier does not affect the round-trip performance. That pattern appears on Figures 3.6 and 3.7, where, in all simulated scenarios, there is no need to use more parking slots than offered vehicles.

Figure 3.8 presents the results for one-way rentals with no walking between stations and without constraints regarding issue (2). This scenario is equivalent to the "partial-floating" case defined by Repoux et al. [2014], where clients can park the vehicle outside the station, but nearby it, when there are no available parking slots in the station. The number of clients served is greater than in the round-trip mode, because one-way rentals tend to be shorter. However, having more clients served does not necessarily result in higher revenue or profit, nor even in more idle vehicles. The increase in the number of served clients shows signs of saturation even for the demand of 500 clients.

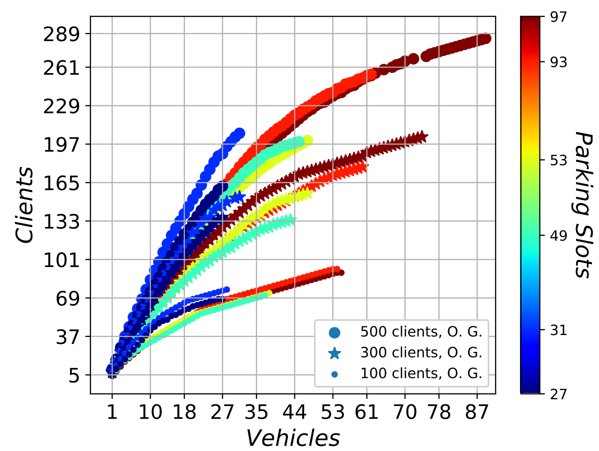


Figure 3.8: One-way scenarios with no issue 2) constraints and no walking

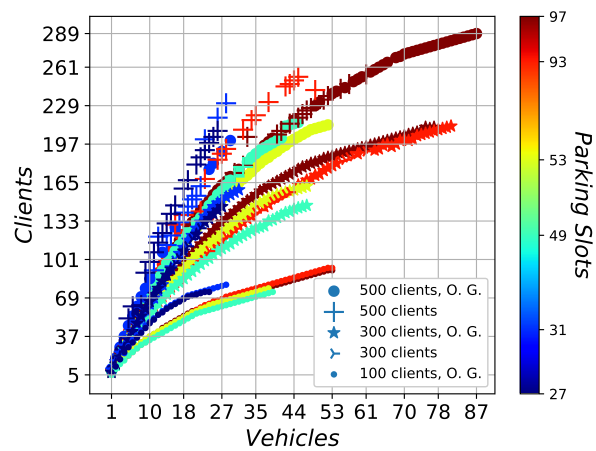


Figure 3.9: One-way scenarios with no issue 2) constraints and 500m walking

Figure 3.9 shows the one-way results also without issue (2) constraints, but considering that the clients are flexible to walk up to 500 meters to find an available vehicle. Some simulations using demand of 500 clients did not obtain optimality guarantee due to the higher complexity of the one-way optimization. Therefore, there is no guarantee that the results with the "+" sign indicated in the legend are the solution with best possible combination for that number of vehicles and parking slots. Each point shown on the graph without optimality guarantee used up to three attempts of 20 minute processing to find an optimal solution. New simulations using a longer run time can achieve the optimal solutions. However, since the non-optimal solutions maintained the same pattern as the optimal solutions, we expect that the optimal solutions are not significantly different from the solutions shown in Figure 3.9.

Even though the number of served clients increased only from 285 to 289 (four more clients, 1.4%), the number of vehicles required to serve those 289 clients was reduced from 89 to 87 (two fewer vehicles, 2.2%). Thus, the walking distance thresholds caused not only an increase in the number of served clients, but also a reduction in the number of vehicles needed to serve that many clients.

Figures 3.10 and 3.11 present the effect of clients walking to be served, but in scenarios with the one-way issue (2) applied only on the partner stations. In practical terms, it is as if the original and additional stations (owned by the carsharing company) can always find enough space to park the one-way delivered vehicles (i.e., parking on the street, or elsewhere), but partner stations are limited to the preset parking slots.

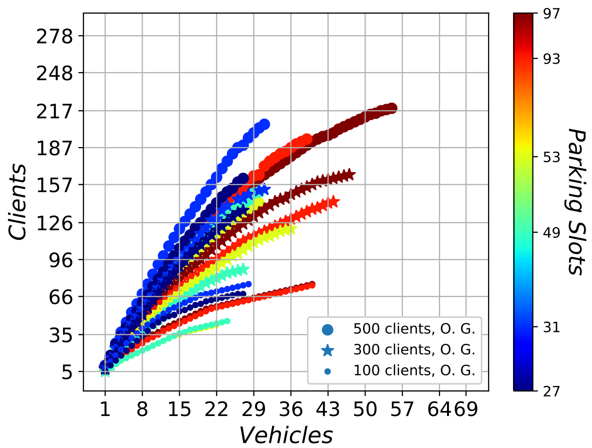


Figure 3.10: One-way with partners subject to issue (2) and no walking

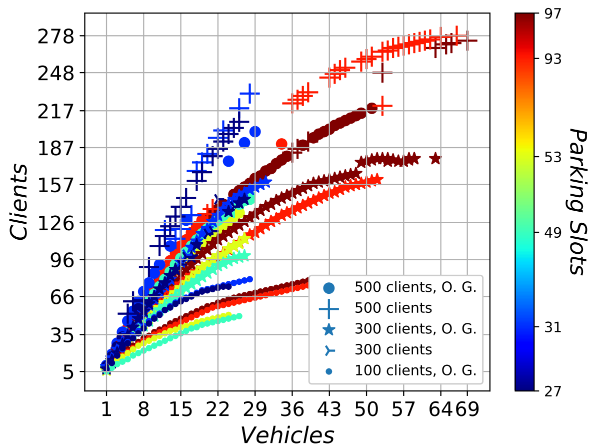


Figure 3.11: One-way with partners subject to issue (2) and 500m walking

Differently from Figure 3.9, in which the clients are flexible to walk and the issue (2) constraints are not applied, all solutions shown on the Figure 3.10 have optimality guarantee. That pattern also occurs on the next figures, indicating that, for the proposed MILP, adding the client walking distance thresholds makes the optimization problem harder to solve than when adding issue (2) constraints.

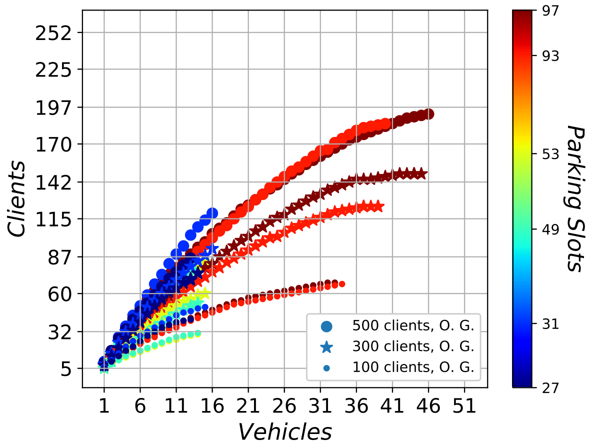


Figure 3.12: One-way with all parking slots subject to issue (2) and no walking

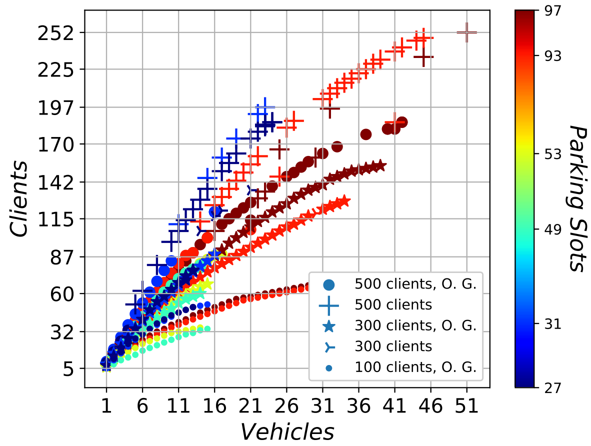


Figure 3.13: One-way with all parking slots under issue (2) and 500m walking

Regarding the carsharing performance, adding issue (2) constraints reduces the maximum number of served clients. This number was reduced from 285 to 219 (66 fewer clients, 23.2%), and the number of needed vehicles drops from 89 to 55 (34 fewer vehicles, 38.2%). Therefore, although issue (2) reduces the number of clients and consequently the revenue obtained, the investment on vehicles can also be reduced.

Besides, Figure 3.10 shows the scenario with 31 parking slots overcoming even the one with 93 parking slots. That happens because solutions using 27 and 31 parking slots are not affected by the one-way issue (2), since those parking slots are not from partners. After applying issue (2) constraints on the partners' parking slots, the difference between the maximum number of served clients using 31 and 97 parking slots dropped from 80 to only 14 clients. This relative difference dropped from 28.1% to only 6.4%.

Figure 3.11 shows how clients walking up to 500 meters can overcome issue (2) on the partner stations. The maximum number of served clients increased from 219 to 278 (up 59 clients, 26.9%) and the number of required vehicles increased from 55 to 69 (up 14 vehicles, 25.5%). Differently from the scenarios shown in Figure 3.9, where the clients walked and the parking slots were not subject to issue (2) constraints, the number of served clients increased together with the number of needed vehicles. This indicates that, as the clients are flexible to walk to other stations seeking for available vehicles and parking slots, the drawbacks from issue (2) are reduced.

Figures 3.12 and 3.13 present the results of applying issue (2) constraints on all parking slots. The performance of these scenarios with only original and additional parking slots (27 and 31 slots respectively) were also impaired. However, Figures 3.12 and 3.13 show the greatest increase in the number of clients served due to the flexibility to walk. The maximum number of served clients increased from 192 to 252 (an increase of 31.3%), using only five more vehicles (a 12.12% increase). Nevertheless, those performance benefits are only possible if a higher number of clients are flexible to walk to another station.

Table 3.21 presents simulation results regarding walking distances, inclusion of issue (2) constraints, maximum number of served clients, vehicles required, and clients who walked to be served. Each line shows the percent increase in the maximum number of served clients, and in the maximum number of needed vehicles if compared to the results from the line above it. The percentage of served clients who had to walk is also shown. The better the value, the greener the cell's background color is.

In general, the greatest increase in percentages either of served clients or needed vehicles happened on the scenarios with 500 or 750 meters of walking. The increasing percentages of one-way served clients were higher than those obtained by Boyacı and Zografos [2019], in which the cumulative increasing along the scenarios barely surpassed 25%. In this simulation, the increase of served clients by walking up to 500 meters instead of 250 meters was up to 30.6% when considering all constraints. A possible reason for the increase of percentages in this simulation being higher than the observed in Boyacı and Zografos [2019] is due to different distances between the simulated stations, since the stations in both works are not equally distanced between themselves.

The number of clients who had to walk to be served always increased as the walking distance increases. Therefore, the walking distance is certainly useful to replace or strengthen the one-way relocation results for maximizing the number of served clients. However, in real situations it is likely that requiring clients to walk lengthy distances would decrease the demand. Furthermore, the benefits of client walking starts to saturate for lengthier walking distances, depending on the distances among stations.

Table 3.22 presents the rate of the number of parking slots divided by the number of needed vehicles, considering the demand of 500 clients per day. The greater the result, the more parking slots are needed per vehicle and the cell's background is illustrated redder. Round-trip mode is not shown in Table 3.22 since their rates would always be 1. That happens for round-trip because all vehicles must be returned to their starting station after the rentals, and the starting station has available parking slots for their allocated vehicles.

The highest rates occur on the scenarios where issue (2) constraints apply to all stations, with the shortest (or nonexistent) walking distance, and using 49 and 53 parking slots. On those scenarios, there are 22 partner stations with only one parking slot each, making issue (2) even more restrictive. Table 3.22 also presents reddish cells near such high rate scenarios, but their value diminish as they indicate scenarios with more parking slots and vehicles in the same stations, and with clients flexible to walk further. Thus, besides the walking distance, having more parking slots in the same station helps to reduce the effects of issue (2). If not being able to include more parking slots in the same station, the additional parking slots should be nearby other stations. By doing so, if a client could not park the rented vehicle in the desired location, the client could easily find and park it in another nearby parking slot, intensifying the sharing dynamics.

Table 3.21: Maximum Clients Served, Vehicles and Walking Percentage

Scenario	Max. Clients	Raise (%)	Max. Vehicles	Raise (%)	Walking Clients (%)
round-trip no walking	228	-	97	-	-
round-trip 250 meters	232	1.8	97	0	1
round-trip 500 meters	235	1.3	97	0	5
round-trip 750 meters	238	1.3	97	0	16
round-trip 1 km	248	4.2	97	0	32
one-way no walking no constraints	285	-	89	-	-
one-way 250 meters no constraints	287	0.7	89	0	1
one-way 500 meters no constraints	289	0.7	87	-2.2	5
one-way 750 meters no constraints	326	12.8	87	0	24
one-way 1 km no constraints	378	16	88	1.1	44
one-way no walking cstr. on partners	219	-	55	-	-
one-way 250 meters cstr. on partners	230	5	62	12.7	3
one-way 500 meters cstr. on partners	278	20.9	79	27.4	8
one-way 750 meters cstr. on partners	309	11.2	74	-6.3	20
one-way 1 km cstr. on partners	357	15.5	69	-6.8	35
one-way no walking all constraints	192	-	46	-	-
one-way 250 meters all constraints	193	0.5	43	-6.5	3
one-way 500 meters all constraints	252	30.6	51	18.6	4
one-way 750 meters all constraints	292	15.9	68	33.3	20
one-way 1 km all constraints	329	12.7	69	1.5	35

In a real environment, generic unexpected events as car crashes or unscheduled vehicle maintenance can interfere on the vehicles availability. Besides, external events such as traffic and rainfall can slow down the sharing dynamics, hampering the flow of clients between stations. In these cases, increasing the number of vehicles and parking

Table 3.22: Rates of One-Way Vehicles per Parking Slot

Scenario	Parking Slots 27	31	49	53	93	97
one-way no walking no constraints	1	1	1.09	1.13	1.48	1.09
one-way 250 meters no constraints	1	1	1.04	1.02	1.16	1.09
one-way 500 meters no constraints	1.04	1.07	1.09	1.02	1.16	1.15
one-way 750 meters no constraints	1	1	1	1.02	1.12	1.17
one-way 1 km no constraints	1	1	1.02	1	1.13	1.10
one-way no walking cstr. on partners	1	1	1.63	1.77	2.38	1.76
one-way 250 meters cstr. on partners	1	1.03	1.75	1.67	1.79	1.56
one-way 500 meters cstr. on partners	1.04	1.11	1.75	2.04	1.39	1.41
one-way 750 meters cstr. on partners	1	1	1.20	1.13	1.39	1.31
one-way 1 km cstr. on partners	1	1	1.09	1.08	1.37	1.23
one-way no walking all constraints	2.08	1.94	3.77	3.53	2.33	2.11
one-way 250 meters all constraints	1.69	2.07	3.50	3.31	2.33	2.26
one-way 500 meters all constraints	1.17	1.30	3.50	3.12	1.98	1.90
one-way 750 meters all constraints	1.17	1.30	1.48	1.47	1.50	1.42
one-way 1 km all constraints	1.25	1.30	1.29	1.47	1.45	1.41

slots would help the carsharing service to be more resilient and reliable for its clients.

In summary, the insights obtained from section 3.1 to this section are:

- computational performance is essential to offer a scalable and optimized service;
- it is unlikely that all trips demand will be served due to the constrained number of vehicles and parking slots;
- it is better to have multiple parking slots in the same station, than having multiple stations but with fewer parking slots each.

The next section presents the proposed general business model for a low-cost carsharing service in São Paulo after considering those insights.

3.4 Proposed Carsharing Business Models

Taking into consideration the insights obtained from analyzing carsharing company profits for round-trip and one-way modalities, differences of computational performance by changing the mathematical formulation to be solved, and benefits for the carsharing dynamics when clients are flexible to be served in nearby stations, this section proposes a general business model that aims to provide low-cost carsharing rentals and can be simulated in an efficient computational time.

It is mentioned as a general business model because different service structures and carsharing modes can be applied to it. This general business model is comprised by the following rules:

- the carsharing company monthly rents, instead of buying, the required vehicles and parking slots;
- clients are flexible to walk to get to available vehicles or parking spaces nearby;
- there is no staff performing vehicle relocations. The supply of vehicles is balanced by the own demand of trips;
- scheduled trips selected to be served have lower rental prices.

The following subsection elaborates how the results from former sections produced insights for this general carsharing business model.

3.4.1 Reasons for Proposing this General Business Model

As mentioned in section 1.3, the research questions of this work consist in computationally plan and simulate an one-way carsharing service so that:

1. a carsharing company is able to profit by offering the service;
2. there are algorithms able to solve big instances of planning and simulation for such carsharing service;

3. the service is low-cost.

The results obtained from section 3.1 indicate that it is possible to make profits with a carsharing service in São Paulo. Also, the MILP models were solved quickly for the evaluated instances. However, the experiments from section 3.1 are sustained by the assumption that no vehicle relocation is needed between carsharing rentals. Indeed, if vehicle relocations are included, operation costs will increase [Jorge et al., 2015; Bruglieri et al., 2018] and it may remove the feature of being a low-cost service. Besides, vehicle relocation operations turn the carsharing fleet-sizing problem into an NP-Hard problem [Albinski, 2015].

Section 3.2 shows that a profitable carsharing service in São Paulo may be optimized in reasonable time if the MILP formulation generates a linear relaxation close to the optimal integral solution. In this case, formulations with no Big-M stand out, although they may require a higher number of constraints or variables. Thus, optimization problems that can be solved in polynomial-time and do not require an exponential number of variables represent an ideal way to plan and simulate such carsharing service. However, since vehicle relocations will not be included, the business model must rely on characteristics that enable the clients, by themselves, to take the most from the fleet, and consequently avoiding idle vehicles.

Results from section 3.3 showed that it is possible to increase the number of served clients by placing more vehicles in the same station instead of creating more stations with fewer vehicles each. Together with clients being flexible to walk to get an available vehicle, the carsharing service can mitigate the issues caused by the unbalanced supply and demand for vehicles along the day and stations. Such business model is possible because it does not have employees dedicated to relocating vehicles between stations. By doing so, it is possible to avoid personnel costs and reduce the time complexity of the problem being solved during the simulations.

Nevertheless, the business model depends either on a previous scheduling of its clients or in a trustful demand of clients. In both of these scenarios, some interested clients may not be served. This general business model is mainly based on scheduled trips. By doing so, the company can organize its fleet before the service starts, making the most from the available resources and trips demand. The following subsections explain how this carsharing service was planned.

3.4.2 Planning the location of stations

Carsharing rentals may have patterns that differ from the common trips made using other transportation services. As analyzed in Ampudia-Renuncio et al. [2020], free-floating rentals have patterns different than expected probably due to the necessity of parking. Since the client is also his/her driver, and thus is responsible to park the vehicle properly, areas with parking limitations may demotivate free-floating clients to use the service. In fact, parking limitations can be a part of policies to hinder individual transport, motivating people to use public transit [Lamour et al., 2019].

Although station-based clients do not suffer from this issue since stations provide parking slots for their clients, it is not possible to ensure that there will be a station with an available parking slot exactly at every client's desired origin and destination. However, it is possible to have stations positioned near the clients origin and destination points so that clients can reach them by walking. This trade-off between location of stations and the origin and destination of trips must be evaluated since carsharing clients may prefer not to use the service if they have to walk for 10 minutes to be served [Hahn et al., 2020].

Two methods of positioning shared mobility stations (also applicable to the carsharing ones) are presented in Section 4.1 and Section 4.2. The objective of both methods is to maximize the sum of utility values by placing stations along the city. In this proposed business model, the utility value varies based on where each station is placed. Along all the studied area, utility values are discretized by street segment, each one valuated as the number of possible trips starting or finishing nearby that street segment. By doing so, places that concentrate more possible clients are provided with stations. After defining the station locations according with this expected demand of clients, the number of parking slots to be placed in each station is defined by another optimization method.

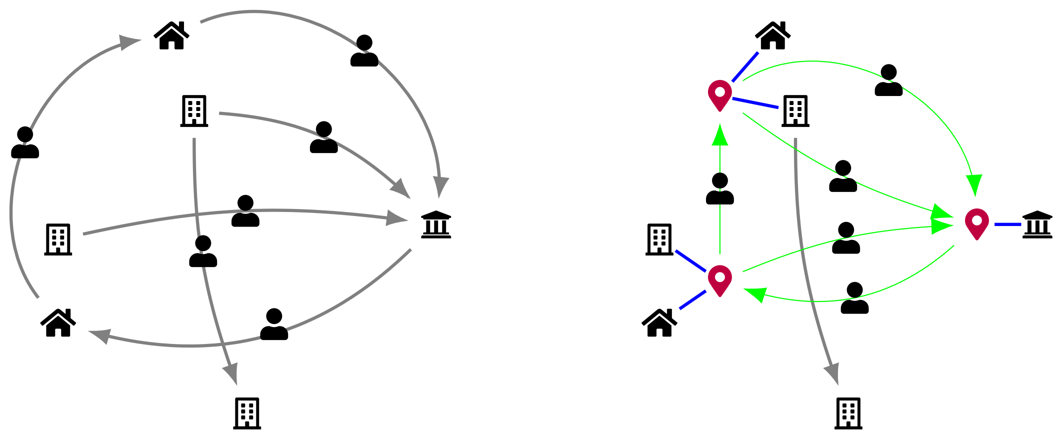
This whole procedure is split into two stages because if the selection of clients and prices are made together with the definition of location for stations, it is possible that some stations keep having their location changed even if they already have a stable and profitable set of clients. In this case, these unnecessary changes may either bother the clients or delay the flow of vehicles between clients if they are not used to the driving right-of-way of the new location where the vehicle should be delivered. Indeed, the convenience of use is pointed by Violin [2021] as a clients' motivator for using vehicle sharing services on demand in Brazil and abroad.

The stage regarding the clients and prices selection is explained in the following subsection.

3.4.3 Selecting clients and prices

This proposed general business model aims to offer carsharing services at the lowest possible price. Therefore, it must reduce operational costs as much as possible. Provided that carsharing prices are expected to be cheaper than ridesourcing services [Schwieterman and Bieszczat, 2017; Benetti, 2019], rental prices should also be limited to the prices charged by TNC companies. A way to achieve that is not to perform vehicle relocation operations, avoiding the associated costs. This is possible if the carsharing company selects in advance which demands will be served given the rental prices defined by the carsharing company itself. Therefore, the served clients will only be the ones whose trips' origin and destination match among themselves, ensuring a low-cost service to these drivers.

To do so, interested drivers must inform their commuting demands to the carsharing company and, according with the matching demands from other drivers, the company decides which demands will be served. Figure 3.14 illustrates this demand selection. Figure 3.14 (a) shows drivers who applied for using the carsharing service. Every arc represents a trip demand, going from its origin place to its destination. Figure 3.14 (b) shows in green the trip demands that were selected by the carsharing service. The selection focused in maximizing the profits of the carsharing company, considering the associated costs and reasonable prices that clients would allow to pay. Therefore, trips to or from low-demand places probably will not be selected, as represented by the trip demand (with a gray arc) going to the building at the bottom of Figure 3.14 (b).



(a) Drivers interested in using the service (b) Demands selected to be served

Figure 3.14: Illustration of the trips selection

The selected trips will start and finish exactly at carsharing stations, represented by location markers, or near these stations. These stations are placed near the origin and destination of trip demands and assume that drivers will be flexible to walk to it to get a car, or walk from it to reach their destination. The walking routes are expressed by the blue edges in Figure 3.14 (b). By grouping trips origin or destination to a nearby station, cars are able to be more reused than if they were spread along the area and sometimes too far from some interested clients. By doing so, the number of idle cars is diminished and costs are reduced.

After the selection, drivers with selected trips will be able to define a contract with the carsharing company assuring that their predefined and selected trip demands will be served. In this general business model, it is assumed that all interested clients are aiming to use carsharing in a weekly routine. Although this assumption does not fit the carsharing clients' profile observed by Hahn et al. [2020], such clients may emerge if stations with available vehicles are near their usual trips origin and destination, and carsharing prices are attractive and fixed. These assumptions corroborate with the Brazilian people's motivators for using vehicle sharing services on demand [Violin, 2021].

Since the demand is well known, the carsharing company does not need to have spare vehicles because no sudden increases in demand are expected. Changes in demand will usually happen when clients change their weekly routines, such as by start working in a different place, or by finishing an academic period. The seasonal patterns, such as academic periods, may justify the carsharing company in not owning the vehicles, since the demand probably will not keep at the same level throughout the whole year. Therefore, in this proposed general business model, the carsharing company may rent its vehicles from a long-term car rental agency instead of owning them and hiring staff to do their maintenance. By doing so, the carsharing company can focus on serving as many drivers as possible, delegating these operational tasks to third-party companies.

Provided that vehicle relocation operations will not be performed, the served trip demands must also ensure that the number of vehicles in each station at the end of the week will be equal to the number of vehicles which started the week at that station. However, imbalances in the fleet along the day and inside the same week may happen since rentals must not finish at the same station where they have started. Also, even with trips being scheduled, eventual fleet imbalances may happen and disrupt the expected flow of vehicles. This may be caused by unforeseen situations such as flat tire, car crashes, selected clients not been able to drive at that day, or simply vehicles arriving late in the destination station due to traffic jams. In these cases, the company may offer on-demand carsharing with allowed destinations to relocate vehicles and rebalance the fleet again along the stations.

This proposed general carsharing business model may work in different versions. The following subsection defines the three versions of this general business model that are compared in this work.

3.4.4 Evaluated Business Models

This work simulates the performance of three business models:

1. Scheduled Free-Floating;
2. Mixed Free-Floating;
3. Mixed Partial-Floating.

On the Scheduled Free-Floating business model, clients previously inform all their routine commuting demands to the carsharing company, and the carsharing company selects which demands will be served. In this work, the carsharing company monthly rents one parking slot per vehicle. Although these free-floating vehicles may be parked in public spaces within the working area, this decision of renting some parking slots was made for reducing the dependence on public parking spaces; and reducing the risks of thefts, vandalism, hail and flood that may damage vehicles parked outdoors.

The Mixed Free-Floating business model differs from the Scheduled Free-Floating by also serving trips not previously scheduled. However, the carsharing company also selects which of the on-demand trips will be served, based on the idle resources from the selected scheduled clients. By doing so, it is possible to avoid new supply and demand imbalances along the fleet and even fix fleet imbalances caused by unexpected issues such as flat tire and traffic jams delaying the flow of vehicles.

And finally, the Mixed Partial-Floating business model is similar to the Mixed Free-Floating but it will work in the modality named as "partial-floating" by Repoux et al. [2014]. In practical terms, it is the same modality as the scenario of one-way without issue (2) constraints discussed in Section 3.3. Thus, vehicles can be delivered in a station different than the one at the start of the rental, and vehicles can be parked outside the station but close to it if at that time the station has no more available parking slots.

Section 4.3 presents an LP formulation to select the best subset of clients that maximizes the company's profits. This formulation considers the stations positioned as in subsection 3.4.2, contemplating all rules of the business models suggested in this subsection, and also assessing the flexibility of clients to walk to nearby stations to get to a vehicle or to find an available parking space.

The next chapter formalizes the evaluated problems, discuss their computational time complexities, and presents their proposed mathematical formulations.

Chapter 4

Computational Complexity and MILP Formulations

This chapter discusses the computational complexity of the proposed problems and presents MILP formulations to solve them. The following section defines the Spacing Shared Mobility Stations problem, proves its NP-Completeness and presents a MILP formulation to solve it. Section 4.2 defines the Stations Allocation in Street Segments problem and presents a formulation to solve it. Section 4.3 presents a polynomial-time formulation to simulate the carsharing system performance. Section 4.4 describes the shortest path algorithm used to calculate the shortest distances in the network of pedestrian ways.

4.1 Spacing Shared Mobility Stations

The Spacing Shared Mobility Stations (SSMS) problem aims to maximize the total utility obtained by placing shared mobility stations along a network of pedestrian ways, considering that all stations must be spaced by a minimum distance threshold between themselves. These stations can be any facility of a shared mobility service, such as carsharing stations, chargers for electric vehicles or racks for bike sharing, which are important for the service but it is impracticable to spread them throughout all the streets. Solving the SSMS helps companies to offer their shared mobility services along all the studied area without needing to invest in more facilities than necessary. The SSMS is defined as follows.

Problem 4.1.1. *Given a network of pedestrian ways $\mathbb{G}$ (non-directed graph) composed by a set $\mathbb{V}$ (vertices) of street intersections, a set $\mathbb{E}$ (edges) of street segments which can have different lengths, and different utility value yielded per street segments and intersections by placing stations on them, what is the maximum total utility that can be obtained by placing stations along the streets (edges) or over the intersections (vertices) so that each station keep spaced of, at least, a distance D from the others?*

SSMS is based on the following assumptions:

- The utility values yielded per station placed are constants;
- The distance between two street segments or street intersections are constants, calculated as the shortest path distance between their street intersections. In this shortest path calculation, the weight of each street segment is its length.

A decision version for this problem used hereafter is similar to the definition of Problem 4.1.1, changing only the question to "*is it possible to obtain a total utility of at least k' by placing stations in the graph $\mathbb{G}$ and spacing them by at least D meters?*". Theorem 4.1.1 and the following subsections shows the NP-Completeness of this problem.

Theorem 4.1.1. *The decision version of Spacing Shared Mobility Stations (SSMS) is NP-Complete.*

In order to prove that the decision version of SSMS is NP-Complete, one must show that the problem belongs to NP and is NP-Hard. The following subsection demonstrates that this problem belongs to NP and the subsection 4.1.2 proves its NP-Hardness.

4.1.1 The Decision Version of SSMS Belongs to NP

A problem belongs to NP if it is possible to verify in polynomial-time a certificate of an "YES" answer for a decision version of this problem [Garey and Johnson, 1979; Cormen et al., 2009]. Considering a certificate composed by the network of pedestrian ways being a graph $\mathbb{G}$ (built with the street segments as edges $e \in \mathbb{E}$ and the street intersections as vertices $v \in \mathbb{V}$) with all placed station $s \in \mathbb{S}$ attached to $\mathbb{G}$ in its respective locations on the network of pedestrian ways, the utility earned by each placed station, the minimum total number of stations k', and a distance threshold for spacing D, an algorithm for verifying an answer in polynomial-time can be:

1. check if the total utility is at least k'. If the total utility is fewer than k', the certificate must be verified as "NO". However, if the total utility is at least k', run the step two as follows.
2. run a shortest path algorithm starting from every station in $|\mathbb{S}|$. If another station is found at a distance shorter than D meters, the certificate must be verified as "NO". However, if no station is found at a distance shorter than D meters from any other station, the certificate must be verified as "YES".

Considering the Dijkstra's algorithm with Fibonacci heap to calculate the shortest paths [Cormen et al., 2009], the complexity time for each station run is $O(|\mathbb{V}|log|\mathbb{V}|+|\mathbb{E}|)$. Therefore, the complexity time for this whole procedure is $O(|\mathbb{S}|(|\mathbb{V}|log|\mathbb{V}|+|\mathbb{E}|))$.

The total time complexity of checking the certificate, after running these two steps, is $O(|\mathbb{S}|+|\mathbb{S}|(|\mathbb{V}|log|\mathbb{V}|+|\mathbb{E}|)) = O(|\mathbb{S}||\mathbb{V}|log|\mathbb{V}|+|\mathbb{S}||\mathbb{E}|)$. Therefore, the final time complexity order is polynomial and the problem belongs to NP. The following subsection proves that this problem is NP-Hard, and consequently, shows that the decision version of SSMS is NP-Complete.

4.1.2 The Decision Version of SSMS Is NP-Complete

A problem Π is NP-Hard if there is a transformation in polynomial-time which converts an instance of an already known NP-Hard problem into an instance of Π. An instance I_1 of a decision problem Π_1 is converted to an instance I_2 of another decision problem Π_2 if their answers are the same. In other words, it is converted if the answer of any I_1 for Π_1 is "YES" if and only if the answer of I_2 for Π_2 is also "YES" [Garey and Johnson, 1979; Cormen et al., 2009].

A transformation in polynomial-time of instances from the Maximum Independent Set problem is proposed in this work to prove that the decision version of the problem SSMS is NP-Hard. The Maximum Independent Set (MIS) problem consists in finding the maximum subset $\mathbb{M}$ of vertices from a graph so that there is no edge connecting any pair of vertices contained in $\mathbb{M}$ [Garey and Johnson, 1979; Cormen et al., 2009; Fleischner et al., 2010]. Given a positive integer threshold k, a decision version for the MIS problem can be: *is it possible to find an independent set with size* $|\mathbb{M}| \geq k$*?*

A polynomial-time transformation from a MIS instance I_1 into a SSMS instance I_2 consists in:

1. creating for I_2 a graph $\mathbb{G}'$ with (initially) the same structure of vertices and edges as the graph $\mathbb{G}$ from I_1. The set of vertices and edges of $\mathbb{G}'$ will be represented by $\mathbb{V}'$ and $\mathbb{E}'$, respectively. All edges in $\mathbb{E}'$ will have length equal to one, the same for the utility value of all vertices and edges. This procedure can be performed in $O(|\mathbb{V}|+|\mathbb{E}|)$ time and does not cause loss of generality for the MIS problem, since it does not consider length or utility value on edges.

2. including a gadget (subset of edges and vertices useful to the proof) at the middle of every edge in $\mathbb{E}'$. These gadgets are comprised by one red vertex connected to

two new paths of two other vertices each, as represented by Figure 4.1. All the red edges included will also have weight and utility value equal to one. This procedure can be performed in $O(|\mathbb{E}|)$ time.

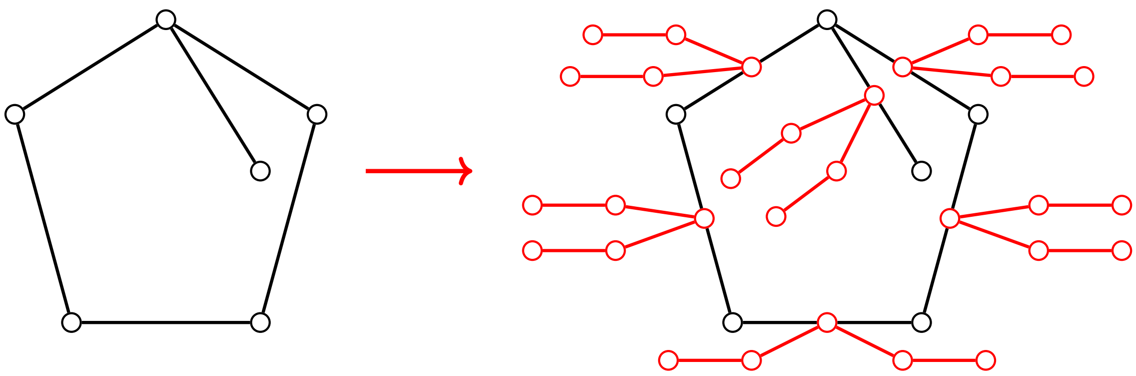


Figure 4.1: Polynomial Transformation of graph $\mathbb{G}$ for MIS into $\mathbb{G}'$ for SSMS

3. creating the threshold for total utility $k' \leftarrow k + 2|\mathbb{E}|$ and setting the minimum distance between stations $D \leftarrow 3$. This procedure can be performed in $O(1)$ time, considering that the number of edges $|\mathbb{E}|$ is known beforehand.

This transformation can be performed in $O(|\mathbb{V}| + |\mathbb{E}| + |\mathbb{E}| + 1)$ time, which has polynomial complexity of time $O(|\mathbb{V}| + |\mathbb{E}|)$. Provided that all I_2 generated instances will have edges length and utility value equal to one, the I_2 instances represent a more specific problem of SSMS in which all edges have the same length and utility value. Together with the parameters k' and D, every instance I_2 for SSMS will yield the same answer as the instance I_1 for MIS.

The following equations and lemma formalize the MIS and SSMS spacing constraints before proving that SSMS is NP-Hard. Table 4.1 and Table 4.2 define, respectively, the sets and values used in this proof of NP-Hardness.

Table 4.1: Sets used in the SSMS NP-Hardness proof

Element	Description
$u, v \in \mathbb{M} \subset \mathbb{V}$	Vertices u and v belonging to the independent set $\mathbb{M}$
$s, a, b \in \mathbb{S}$	Stations s, a and b placed along an edge or at a vertex

Table 4.2: Values used in the SSMS NP-Hardness proof

Constant	Description
D	Minimum distance allowed between stations
k	Minimum threshold for yielding "YES" in the MIS problem
k'	Minimum threshold for yielding "YES" in the SSMS problem
$\delta(u, v)$	Shortest distance between vertices u and v
$\delta(a, b)$	Shortest distance between stations a and b

In a connected unweighted graph $\mathbb{G}$, the shortest path between any two vertices u and v belonging the independent set $\mathbb{M}$ must have distance greater than or equal to two. This happens because if the shortest distance between a pair of different vertices u and v were shorter than two, u and v would be adjacent, and thus, would not belong to the same independent set. The following relation presents this independent set constraint, using $\delta(u, v)$ to indicate the shortest path distance between vertices u and v.

$$\nexists\ \delta(u, v) < 2 \qquad\qquad \forall u, v \in \mathbb{M} \subset \mathbb{V} \mid u \neq v \tag{4.1}$$

Similarly, a formal definition for the spacing constraint of SSMS is presented by Equation 4.2, where $\delta(a, b)$ is the shortest path distance between the stations a and b, and D is the distance threshold. Since every station must be spaced by at least distance $D = 3$ from the others, and all edges on I_2 instances have length equal to one, the closest any pair of stations can be from each other is three edges distant.

$$\nexists\ \delta(a, b) < D \qquad\qquad \forall a, b \in \mathbb{S} \mid a \neq b, D = 3 \tag{4.2}$$

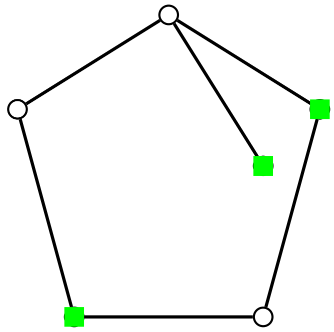
An optimal solution for I_2 instances of SSMS will not have a station on any edge originated from I_1 instance (black edges). Figure 4.2 illustrates this pattern with a non-optimal solution where a single station (green square) on Figure 4.2 (a) have already fulfilled all available length from the graph fragment due to the parameter $D = 3$, not allowing additional stations on that graph fragment. However, the Figure 4.2 (b) presents an optimal solution for the same graph fragment by placing a station on a black vertex (original from an I_2 instance) and, consequently, having enough space to include two more stations on the pendant red vertices.

(a) Non-optimal solution with no remaining space for additional stations

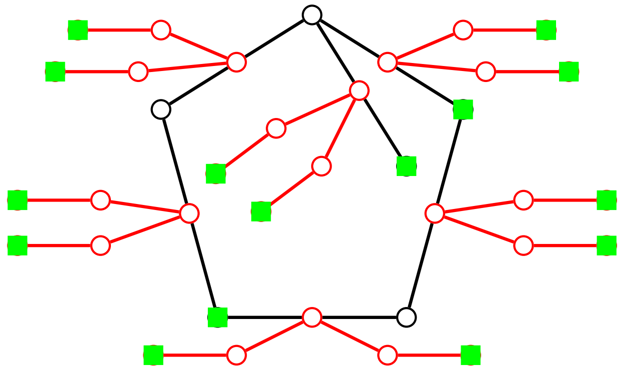
(b) Optimal solution with no remaining space for additional stations

Figure 4.2: Pattern of optimal solutions for I_2 instances of SSMS

Given the gadgets in I_2, no optimal solution for SSMS will place any station along black edges. This happens because moving a station s from along a black edge to a black vertex will enable the allocation of two additional stations on any gadget, as shown in Figure 4.2. Since the utility value obtained for placing stations in any vertex or edge of I_2 is the same, these two additional stations will provide greater total utility value than allocating s anywhere along a black edge. This pattern will be present in all optimal solutions for the I_2 instances of SSMS, assuring that no station will be placed on black edges. Figure 4.3 (b) illustrates it by presenting an optimal solution for the graph $\mathbb{G}'$ previously shown in Figure 4.1.



(a) Optimal solution for $\mathbb{G}$ with $k = 3$ stations



(b) Optimal solution for $\mathbb{G}'$ with $k' = k + 2|\mathbb{E}|$

Figure 4.3: Optimal solution for the $\mathbb{G}$ and $\mathbb{G}'$ example

Regardless of the instance, every gadget will have two stations. Provided that the gadgets have stations on their pendant vertices, the red vertices connected to the black ones are not spaced enough to have an additional station. Since the polynomial transformation places gadgets at the middle of every black edge from I_1, the shortest distance between any pair of black vertices on I_2 is an even number: one hop to reach the red vertex from a gadget and an additional hop to reach other black vertex, always performing sequences of two hops between black vertices. The following assertion clarifies the pattern of distance between stations on black vertices.

Assertion 4.1.1. Every pair of black vertices with stations are spaced by a distance of four or greater even number.

The proof that SSMS is NP-Hard consists in showing that every transformed instance I_2 for SSMS will yield the same answer of an original instance I_1 for MIS. This can be proved by showing that for all instance I_1 whose decision version of the MIS problem yields "YES", a transformed instance I_2 for a decision version of SSMS will also yield "YES"; and also showing the inverse: for every instance I_2 for SSMS yielding "YES", its original instance I_1 for MIS will also yield "YES". The proof is shown as follows.

Proof. Suppose, for the sake of contradiction, that the MIS instance I_1 would yield an answer "YES" while its transformed SSMS instance I_2 would yield an answer "NO". This would mean that MIS could arrange a set $\mathbb{M}$ from $\mathbb{G}$ with at least k vertices while SSMS could not place at least $k' = k + 2|\mathbb{E}|$ stations on $\mathbb{G}'$. That is false because the polynomial transformation doubled the shortest distance between two black vertices by including the gadgets at the middle of every black edge. Therefore, all equivalent vertices of $\mathbb{M}$ in $\mathbb{V}'$ are at least four edges spaced from each other.

Provided that three edges of distance between every station is enough, as presented by Equation 4.2, the graph $\mathbb{G}'$ will have at least as many stations on black vertices as the entire $\mathbb{M}$ has of vertices. Furthermore, every gadget in $\mathbb{G}'$ will yield two stations regardless of which black vertices also have stations. Since there is one gadget in $\mathbb{G}'$ for every edge from $\mathbb{G}$, and there will be at least k stations on black vertices from $\mathbb{G}'$, if an I_1 instance can arrange a set $\mathbb{M}$ with at least k vertices, every instance I_2 will place at least $k' = k + 2|\mathbb{E}|$ stations. Therefore, if an I_1 instance for MIS yield an answer "YES", its transformed I_2 instance for SSMS will also yield an answer "YES".

Now suppose, for the sake of contradiction, that the SSMS instance I_2 would yield an answer "YES" while the MIS instance I_1 would yield an answer "NO". This would mean that SSMS could place at least $k' = k + 2|\mathbb{E}|$ stations on $\mathbb{G}'$ while MIS could not arrange a set $\mathbb{M}$ from $\mathbb{G}$ with at least k vertices. This is false because as stated by Assertion 4.1.1, the black vertices with stations are spaced from each other by a distance of four or greater even number. Since the black vertices from I_2 are equivalent to the vertices from I_1 spaced by the double of the distance (due to the gadgets included), and a distance equal to two between vertices in $\mathbb{M}$ is enough (as defined by Equation 4.1), the number of vertices in $\mathbb{M}$ will be at least as much as the number of black vertices with a station in $\mathbb{G}'$.

Now considering the two stations placed in every gadget of $\mathbb{G}'$, if I_2 can place $k' = k + 2|\mathbb{E}|$ stations, I_1 will arrange a set $\mathbb{M}$ with k vertices, as shown by the example in Figure 4.3. Therefore, if an instance I_2 for SSMS yield an answer "YES", its original instance I_1 for MIS will also yield an answer "YES". ■

Since the decision version of the SSMS problem is in NP and is NP-Hard, the decision version of the SSMS problem is NP-Complete. Consequently, the optimization version of the SSMS problem is NP-Hard. Given the NP-Hardness of the MIS problem, this proof still holds even for simple SSMS instances when:

- the utility value got per station placed is the same for all the network of pedestrian ways (edges and vertices);
- street segments have the same length;
- the graph obtained by the network of pedestrian ways is planar with maximum degree equal to four [Fleischner et al., 2010].

Therefore, this proof is applicable to the majority of networks of pedestrian ways, including the assessed Metropolitan Region of São Paulo. Besides, it applies to any other optimization problem that consists in selecting places along edges of a graph, having a minimum space between these places. The next section proposes a MILP formulation to solve the SSMS problem.

4.1.3 MILP Formulation

This formulation considers the following assumptions:

1. the network of pedestrian ways is expressed as a non-directed graph;
2. every edge length is shorter than the maximum allowed spacing distance between stations;
3. every edge has some information distinguishing which of its connected vertices is at the start and which vertex is at the end of the edge;
4. placing a station in a vertex yields the greatest utility value among its adjacent edges.

Although these assumptions may not be promptly satisfied by a graph representing a road network, the four assumptions can be easily implemented in the graph by removing specific directions of edges, splitting long edges, arbitrarily defining in which vertices the edge starts and ends, and creating artificial edges to hold a different vertex utility value. The first assumption is needed because SSMS focuses in the distance walked by pedestrians, which are not restricted to walk in only one direction along the sidewalks. The second assumption assures that there will be at maximum one station per edge. The third assumption simplifies the position calculation of the placed stations. And the fourth assumption simplifies the formulation to avoid dealing with stations on vertices.

Table 4.3 defines the sets used to deal with edges in the MILP formulation, Table 4.4 presents the indices used to identify variables, Table 4.5 presents the variables, and Table 4.6 presents the model parameters of the formulation.

Table 4.3: Sets of the SSMS MILP Formulation

Set	Description
$A_e \subseteq \mathbb{E}$	Edges entirely within distance $D/2$ from edge e
$\Omega_e \subset \mathbb{E}$	Edges that do not belong to A_e, but are within distance D from edge e

Table 4.4: Indices of the SSMS MILP Formulation

Index	Description
$e_1, e_2 \in \mathbb{V}$	Vertices of the network. Edge e begins at e_1 and ends at e_2
$e = 1, 2, 3, ..., m$	Edge identifier. The same pair of vertices e_1, e_2 may have multiple edges

Figure 4.4 illustrates the formulation's symbols. Figure 4.4 (a) clarifies the difference between the sets A_e and Ω_e. Edge e is at the graph's bottom, represented in green.

Table 4.5: Variables of the SSMS MILP Formulation

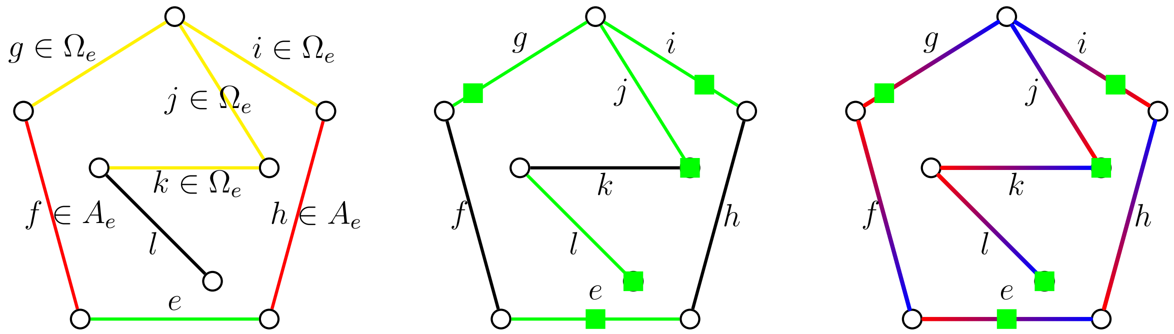
Variable	Description
x_e	Boolean variable indicating if edge e have one station along it
p_e	Continuous variable with the station's position (if it exists) in edge e

Table 4.6: Values of the SSMS MILP Formulation

Model Parameter	Description
u_e	Utility value obtained by placing a station in edge e
L_e	Length of edge e
$\delta(i, u)$	Shortest distance between vertices i and u
D	Minimum distance allowed between stations

Regardless of where a station is placed in e, the edges in red will not be able to have another station due to the minimum distance threshold of $D = 4$. Therefore, if e has a station, edges f and h, must not have another one because they belong to A_e. In this case, edges e, f and h are so close to each other that it is not even needed to check the p_e, p_f and p_h values to discover that these edges together may have at maximum one station.

A threshold $D/2$ is used as distance to assure that even edges in the opposite direction from e, but also in A_e, are nearby each other. An edge is in A_e if its entire length is distanced by at most $D/2$ from the entire length of e. In this case, any two edges in A_e reach each other by distance smaller than or equal to D by passing through e (a path with maximum distance $D/2+D/2$). Therefore, if one edge in A_e has one station, neither e nor other edge in A_e may have another station because they are nearby each other.



(a) $D = 4$
$A_e = \{f, h\}$
$\Omega_e = \{g, i, j, k\}$

(b) $D = 1.53$
$x_f = x_h = x_k = 0$
$x_e = x_g = x_j = x_i = x_l = 1$

(c) $D = 1.53$
$p_e = 0.5,\ p_i = 0.72$
$p_g = 0.81,\ p_j = 1,\ p_l = 0$

Figure 4.4: Illustration of SSMS sets, variables and model parameters

Every edge in all three graphs have length equal to one. Figure (a) distinguishes edges in A_e and Ω_e by colors red and yellow, respectively. Figure (b) shows the binary variables of each edge indicating whether it has or not a station on it. And Figure (c) illustrates each station position's value, ranging from zero to one. The bluish part of the edge is its start, and the reddish part is where the edge ends.

However, edges g, i, j and k, represented in yellow at Figure 4.4 (a), are not that close to e and the decision of placing more than one station on them may require analysing their position values p_e, p_g, p_i, p_j and p_k. Since edges g, i, j k do not belong to A_e and they are within distance D from e, they belong to Ω_e. Nevertheless, edge l is far enough from edge e, not being affected by any station that could be placed in e. Thus, edge l is neither in A_e nor in Ω_e.

The A_e set is useful for reducing the number of constraints and simplifying them in the MILP formulation. Preventing e and all edges in A_e to have together more than one station can be expressed using only one linear constraint, and without needing to apply a Big–M. Therefore, using constraints based on A_e will probaly make this MILP formulation lighter and faster to be solved than dealing with the exact position of any possible station nearby edge e, as it is needed to be made for edges in Ω_e.

Figure 4.4 (b) and Figure 4.4 (c) illustrate, respectively, variables x and p in an optimal solution considering $D = 1.53$. The binary variables x have value equal to 1 if there is a station in that edge, otherwise the variables will have value equal to 0. The variables p have the station position along the edge. The position is actually the distance between start of the edge and the station location. Figure 4.4 (c) represents the start of the edge as blue and the end of the edge as red. Since the edges on this figure have length equal to 1, the station at the middle of edge e has set p_e to 0.5, the stations at almost the end of the edges i and g have positions respectively of 0.72 and 0.81, the station at the extreme end of edge j made the p_j have value equal to 1, and the station at the extreme beginning of edge l made p_l have value equal to 0.

The distance between stations can be calculated using the positions p and the information of which vertex an edge starts or ends at. If the shortest path from a station to another one includes the segment between the start of the edge and the station location, that edge segment length is equal to edge's p. Taking as example the stations in Figure 4.4 (c), the shortest distance from the station in edge g to the station in edge i can be calculated by simply summing $p_g + p_i = 0.81 + 0.72 = 1.53$. The same happens for calculating the shortest distance between any pair of stations in the edges g, j and i.

However, if the shortest path from a station to another one includes the segment between the station location and the edge end, that edge segment length is equal to the total edge's length minus the edge's p. The reason of this subtraction is not to consider the segment between the start of the edge up to the station, represented as the bluish part of the edge, which is not present in this shortest distance. As an example also from Figure 4.4 (c), the shortest distance from edge i to edge e can be calculated by $(L_i - p_i) + L_h + p_e = (1 - 0.72) + 1 + 0.5 = 1.78$, where $(L_i - p_i)$ is the length of edge i minus the start segment of edge i. The other terms $L_h + p_e$ are the next steps of the path, composed by the whole length of edge h plus the beginning of edge e up to its station.

This rationale of distance calculation is used in the proposed SSMS MILP formulation. This formulation is presented by Equations 4.3 to 4.10. Equation 4.3 defines the objective function of maximizing the total utility obtained by placing the stations. Inequality 4.4 assures that at most one station can be placed in either edge e or in any edge in A_e. Inequalities 4.5 to 4.8 assure the spacing between edges in Ω_e. Inequality 4.9 and Equation 4.10 define, respectively, the domains for p_e and x_e.

$$\arg\max_{x_e} \sum_{e=1}^{m} u_e x_e \tag{4.3}$$

Subject to:

$$x_e + \sum_{f=1}^{m} x_f \leq 1 \qquad \forall e \in \mathbb{E} \mid f \in A_e \tag{4.4}$$

$$p_e + \delta(e_1, f_1) + p_f \geq D - D(2 - x_e - x_f) \qquad \forall e \in \mathbb{E}, \forall f \in \mathbb{E} \mid f \in \Omega_e \wedge e > f \tag{4.5}$$

$$(L_e - p_e) + \delta(e_2, f_1) + p_f \geq D - D(2 - x_e - x_f) \qquad \forall e \in \mathbb{E}, \forall f \in \mathbb{E} \mid f \in \Omega_e \wedge e > f \tag{4.6}$$

$$p_e + \delta(e_1, f_2) + (L_f - p_f) \geq D - D(2 - x_e - x_f) \qquad \forall e \in \mathbb{E}, \forall f \in \mathbb{E} \mid f \in \Omega_e \wedge e > f \tag{4.7}$$

$$(L_e - p_e) + \delta(e_2, f_2) + (L_f - p_f) \geq D - D(2 - x_e - x_f) \qquad \forall e \in \mathbb{E}, \forall f \in \mathbb{E} \mid f \in \Omega_e \wedge e > f \tag{4.8}$$

$$0 \leq p_e \leq L_e \qquad \forall e \in \mathbb{E} \tag{4.9}$$

$$x_e \in \{0, 1\} \qquad \forall e \in \mathbb{E} \tag{4.10}$$

Inequalities from 4.5 to 4.8 mean that the distance between the position of a station in edge e and the position of another station in edge f must not be shorter than D, if such stations exist. Constraints from Inequality 4.5 are regarding pairs of edges whose shortest path includes the segment between the station and the start of the edges e and f, being calculated using directly its p_e and p_f values. Therefore, the left-hand side of Inequality 4.5, composed by the expression $p_e + \delta(e_1, f_1) + p_f$, represents the distance inside edge e from its station, plus the shortest distance between vertices e_1 and f_1, plus the distance inside edge f up to its station. Inequalities 4.6, 4.7 and 4.8 calculate the same, but for situations where the shortest distance includes the edge segment between the station and end of the edge, indicated by $(L_e - p_e)$ and $(L_f - p_f)$. Since the network of pedestrian ways is a non-directed graph, the shortest distance from e to f is the same as from f to e. Therefore, there is no reason to repeat the same constraint inverting edges e and f. The $e > f$ condition in the constraints' domain from Inequality 4.5 to 4.8 avoids this repetition.

The Big–M method was used to relax the right-hand side of the constraints from Inequality 4.5 to Inequality 4.8 if the edge has no station on it. In this case, the Big–M

method consisted in subtracting the D threshold if edges e or f do not have a station on them. By doing so, if at least one edge of these do not have a station, the constraint will not bring any effects to the optimization. However, if both edges e and f have a station, the right-hand side will not be relaxed because it will be subtracted by zero: $D - D(2 - x_e - x_f) \rightarrow D - D(2 - 1 - 1) = D - D \times (0) = D$.

The following section presents the Stations Allocation in Streets Segments problem, proves it to be NP-Complete and proposes a MILP formulation to solve it.

4.2 Stations Allocation in Street Segments

The Stations Allocation in Street Segments (SASS) problem also aims to maximize the total utility got by selecting where stations should be allocated in a network of pedestrian ways. However, although SSMS defines the exact locations to place stations, SASS defines which street segments should have a station.

The main drawback of using SASS instead of SSMS is the lower precision on the stations' location. Provided that this problem only selects street segments, the location precision to allocate stations becomes highly influenced by the street segments length, which vary and can be long. However, this issue may not be severe in urban populated areas, since city blocks usually are not longer than a maximum distance pedestrians are flexible to walk. Besides, long street segments can be split in two or more smaller segments, mitigating this lower precision issue.

Nevertheless, it is expected that in a real-world scenario many locations suggested by an optimal solution for SSMS will not end up being used as stations. Provided that not all addresses in the assessed city might be available for renting or buying, shared mobility companies may be forced to use sub-optimal locations by choosing nearby available addresses to become stations. Therefore, the endeavor invested in solving the SSMS problem may not worth it since there is a faster MILP formulation that reasonably approaches the same real-world task.

Similarly to SSMS, SASS decides where to allocate stations without placing them near each other. The SASS problem is defined as follows.

Problem 4.2.1. *Given a network of pedestrian ways $\mathbb{G}$ (non-directed graph) composed by a set $\mathbb{V}$ (vertices) of street intersections, and a set $\mathbb{E}$ (edges) of street segments which can have different lengths and yield different utility value by allocating stations on them, what is the maximum total utility that can be obtained by allocating at most one station per street segment so that all supplied street segments keep spaced of, at least, a distance*

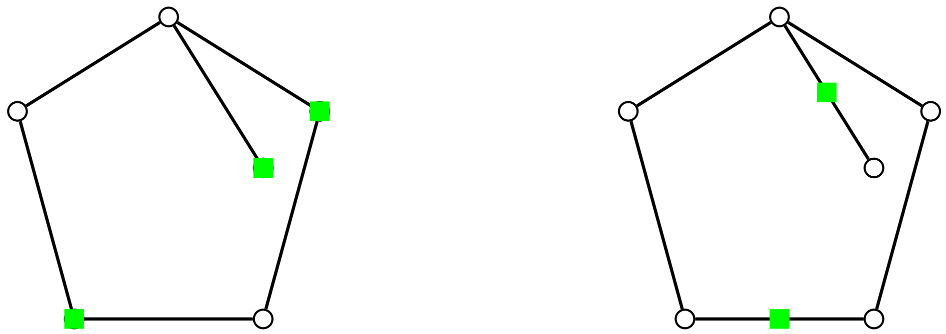
D from the others?

SASS is based on the following assumptions:

- The utility values yielded per station placed are constants;
- The distance between two street segments are constants calculated as the shortest path distance between their street intersections. In this shortest path calculation, the weight of each street segment is its length.

In order to simplify the descriptions, hereafter in this work two street segments are described as edges nearby or edges nearby each other if the shortest distance between their vertices is lower than or equal to D. Similarly, if a street segment e can be reached within distance D from a street intersection v, it will be described as an edge e being nearby a vertex v.

Considering the instance of Figure 4.5 and $D = 2$, the only pair of edges that are not nearby each other are the edge connected to the pendant vertex, and the edge at the bottom of the graph. Therefore, the only optimal solution for SASS on this instance consists in placing one station in both of these edges. In comparison with the optimal solution for the SSMS problem, as shown by Figure 4.5 (a) using the same input (graph $\mathbb{G}$ and $D = 2$), the SASS problem obtained one station less. This happens because SASS is more restrictive than SSMS, neither allowing stations on vertices nor considering the distance inside the supplied edges while calculating the spaced distance. This rationale leads to the following assertion.



(a) Optimal solution for SSMS (b) Optimal solution for SASS

Figure 4.5: Optimal solutions with parameter $D = 2$

Assertion 4.2.1. The objective function value from an optimal solution for SASS is a lower bound for the objective function value from an optimal solution for SSMS, when considered the same instance for both problems.

Assertion 4.2.1 is important for a company interested in solving the SASS problem since it can be assured that no investment would be wasted by allocating more stations than needed. Companies can also solve the SASS problem after splitting lengthy edges into smaller ones, reducing the effect of parameter D on the optimal solution. However, the more edges are split, the larger the graph becomes. Thus, holding the graph in memory and solving this new SASS instance will demand more computational efforts than solving the original SASS instance.

Although SASS is simpler than SSMS, a decision version of SASS is also NP-Complete. The decision version for SASS used hereafter is similar to the definition of Problem 4.2.1, changing only the question to "*is it possible to obtain a total utility of at least k' by selecting edges to have one station each, and only selecting edges spaced by at least distance D from each other?*". Theorem 4.2.1 and the following subsections demonstrates the NP-Completeness of this problem.

Theorem 4.2.1. *The decision version of Stations Allocation in Street Segments (SASS) is NP-Complete.*

As made for SSMS in section 4.1, proving that a decision version of SASS is NP-Complete consists in showing that it belongs to NP and is NP-Hard. The following subsection demonstrates that this problem belongs to NP and the subsection 4.2.2 proves its NP-Hardness.

4.2.1 The Decision Version of SASS Belongs to NP

It is possible to verify in polynomial-time an answer for this decision version of SASS similarly to the algorithm proposed for SSMS. The total time complexity of checking the certificate, after running those two steps, is $O(|\mathbb{S}| + |\mathbb{S}|(|\mathbb{V}|log|\mathbb{V}| + |\mathbb{E}|)) = O(|\mathbb{S}||\mathbb{V}|log|\mathbb{V}| + |\mathbb{S}||\mathbb{E}|)$. Therefore, the final time complexity order is polynomial and the problem belongs to NP. The following subsection proves that this problem is NP-Hard, and consequently, shows that a decision version of SASS is NP-Complete.

4.2.2 The Decision Version of SASS Is NP-Complete

A transformation in polynomial-time of instances from the Maximum Induced Matching problem is proposed in this work to prove that a decision version of SASS is NP-Hard. The Maximum Induced Matching (MIM) problem consists in finding the maximum subset $\mathbb{M}$ of edges from a graph $\mathbb{G}$ so that:

- no edges in $\mathbb{M}$ share any vertex (it is a matching);
- and no edges in $\mathbb{M}$ are connected by any other edge in $\mathbb{G}$ (so all vertices from edges in $\mathbb{M}$ are at least two hops distant).

By doing so, every pair of vertices in $\mathbb{M}$ is already connected to their edges from $\mathbb{G}$. Thus, it forms an induced subgraph of $\mathbb{G}$ [Cameron, 1989; Ko and Shepherd, 2003]. Given a positive integer threshold k, a decision version for the MIM problem can be: *is it possible to find an induced matching with size* $|\mathbb{M}| \geq k$*?*

A polynomial-time transformation from a MIM instance I_1 into a SASS instance I_2 consists in:

1. creating for I_2 a graph $\mathbb{G}'$ with the same structure of vertices $\mathbb{V}$ and edges $\mathbb{E}$ as the graph $\mathbb{G}$ from I_1. The set of vertices and edges of $\mathbb{G}'$ will be represented by $\mathbb{V}'$ and $\mathbb{E}'$, respectively. Every edge in $\mathbb{E}'$ will have length and utility value equal to one. This procedure can be performed in $O(|\mathbb{V}| + |\mathbb{E}|)$ time and does not cause loss of generality for the MIM problem, since it does not consider length or utility value on edges.
2. setting the parameter $D \leftarrow 2$. This procedure can be performed in $O(1)$ time.

This transformation can be performed in $O(|\mathbb{V}|+|\mathbb{E}|+1)$ time, which has polynomial complexity of time $O(|\mathbb{V}| + |\mathbb{E}|)$. Together with the parameters k' and D, every instance I_2 for SASS will yield the same answer as the instance I_1 for MIM. Table 4.7 presents the values used in this NP-Hardness proof.

Table 4.7: Values used in the SASS NP-Hardness proof

Constant	Description
D	Minimum distance allowed between edges with stations
k	Minimum threshold for yielding "YES" in the MIM problem
k'	Minimum threshold for yielding "YES" in the SASS problem

As made for SSMS, the proof that SASS is NP-Hard consists in showing that for all instance I_1 whose decision version of the MIM problem yields "YES", a transformed instance I_2 for a decision version of SASS will also yield "YES"; and also showing the inverse: for every instance I_2 for SASS yielding "YES", its original instance I_1 for MIM will also yield "YES". The proof that SASS is NP-Hard is shorter and simpler than the SSMS's proof and is shown as follows.

Proof. Suppose, for the sake of contradiction, that the MIM instance I_1 would yield an answer "YES" while its transformed SASS instance I_2 would yield an answer "NO". This would mean that MIM could arrange a set $\mathbb{M}$ from $\mathbb{G}$ with at least k edges while SASS could not select at least k' edges for having stations on $\mathbb{G}'$. That is false because every two edges in the induced matching $\mathbb{M}$ are neither incident nor share any incident edge in $\mathbb{G}$, thus, all edges in $\mathbb{M}$ are at least two hops distant from each other in $\mathbb{G}$. Since the polynomial transformation set every edge length to one and $D = 2$, the SASS instance I_2 is enabled to select edges spaced by two or more hops of distance. Therefore, for every MIM instance I_1 that yields "YES", its transformation I_2 for SASS will also yields "YES".

Now suppose, for the sake of contradiction, that the SASS instance I_2 would yield an answer "YES" while the MIM instance I_1 would yield an answer "NO". This would mean that SASS could select at least k' edges for having stations on $\mathbb{G}'$ while MIM could not arrange a set $\mathbb{M}$ from $\mathbb{G}$ with at least k edges. This is false because any combination of selected edges SASS can arrange using $D = 2$ is also an induced matching for its equivalent instance I_1 for MIM. This happens because a first hop spaced between every two selected edges is enough to define these selected edges as a matching, and a second spacing hop among those edges is enough for having at least one vertex, outside the matching, separating every two selected edges in the matching. These separating vertices assure that the matching is already an induced subgraph, because the vertices in the matching are not connected by any other edges of the graph $\mathbb{G}'$. Since the matching is an induced subgraph, the matching is induced. Therefore, if an instance I_2 for SASS yield an answer "YES", its original instance I_1 for MIM will also yield an answer "YES". ■

Since the decision version of SASS is in NP and is NP-Hard, the decision version of SASS is also NP-Complete. Consequently, the optimization version of the SASS problem is NP-Hard. Given the NP-Hardness of the MIM problem, this proof still holds even for simple SASS instances when:

- the utility value got per station allocated is the same for all the network of pedestrian ways (edges and vertices);
- street segments have the same length;
- the graph obtained by the network of pedestrian ways is planar with maximum degree equal to four [Ko and Shepherd, 2003].

Similarly to SSMS, this proof is applicable to the majority of networks of pedestrian ways, including the assessed Metropolitan Region of São Paulo. Also, it applies to any other optimization problem of selecting a subset of edges with a minimum space between them. The next subsection explains the proposed MILP formulation for solving SASS.

4.2.3 MILP Formulation

Table 4.8 and Table 4.9 describe, respectively, the variables and model parameters defined in this MILP formulation.

Table 4.8: Variables of the SASS MILP Formulation

Variable	Description
x_i	Boolean variable indicating if edge $i \in \mathbb{E}$ has one station

Table 4.9: Values used in the SASS MILP Formulation

Model Parameter	Description
u_i	Utility value obtained by allocating a station in edge i
m	Number of edges, equivalent to $\lvert\mathbb{E}\rvert$
$\delta(i,j)$	Shortest distance from any vertex of edge i to any vertex of edge j
D	Minimum distance allowed between stations

The following equations define this MILP formulation. Equation 4.11 defines the objective function and Inequalities from 4.12 to Equation 4.14 define the constraints. The objective function maximizes the sum of utility values obtained by placing stations on edges. Inequality 4.12 assures that at most one station can be placed in either edge i or in any edge j distanced up to $D/2$ from i. This constraint has a rationale similar to the set A_e and Inequality 4.4 from the SSMS MILP formulation, being also used for making this MILP formulation lighter and faster to be solved. Inequality 4.13 avoids edge i and any edge in j distanced up to D to have, together, more than one station. And finally, Equation 4.14 defines x_i as binary variables.

$$\arg\max_{x_i} \sum_{i=1}^{m} u_i x_i \tag{4.11}$$

Subject to:

$$x_i + \sum_{\substack{j=1 \\ j \neq i \\ \delta(i,j) < D/2}}^{m} x_j \leq 1 \qquad \forall i = 1, 2, ..., m \tag{4.12}$$

$$x_i + x_j \leq 1 \qquad \forall i = 1, 2, ..., m;\ \forall j = i, ..., m \mid D/2 \leq \delta(i,j) \leq D \tag{4.13}$$

$$x_i \in \{0, 1\} \qquad \forall i = 1, 2, ..., m \tag{4.14}$$

The next section defines the Carsharing Flow Network problem.

4.3 Carsharing Flow Network

The Carsharing Flow Network problem (CFN) consists in solving a LP to simulate the performance of the carsharing business models proposed at subsection 3.4.4. This LP is formulated as a flow network problem, similarly to the Min-Cost Flow described in Wolsey [1998] and Bazaraa et al. [2010]. The CFN problem is defined as follows.

Problem 4.3.1. *Let $\mathbb{G}$ be a flow network composed by vertices indicating stations and clients, and edges indicating costs and profits of a carsharing service. Considering that clients are flexible to be served in different nearby stations, all vehicles must be returned to their initial stations at the end of the simulated period, and stations have a minimum and maximum number of parking slots allowed, what is the flow over $\mathbb{G}$ that maximizes the company's total profit?*

Regarding the operational details, the CFN is based on the following assumptions:

- Clients are flexible not to have all of their trip demands served by this carsharing service;
- Clients are flexible to walk to or from any station located within a predefined maximum walking distance;
- Clients do not have an option to wait for being served. Each trip demand is promptly served or not served;
- If electric vehicles are used, the time they stay idle at stations is enough for recharging them;
- If combustion engine vehicles are used, some clients will be flexible to take the rented vehicle to a gas station for refueling (the carsharing company will pay for the gas) in return for some benefits, such as price discount.

Figure 4.6 presents an example of a flow network $\mathbb{G}$ for CFN. From left to right, the flow network represents the time passing along a routine period. Each row of nodes and arcs represents the evolution over time of each station, from when it receives its resources (vehicles and parking slots), up to when the simulated period finishes. In the simulated scenarios of this work, the period represents the weekly routine of trips made by the possible clients.

The gray nodes A, B and C represent three carsharing stations either at the start of the week, or at the end of the week. White nodes with numbers indicate trip profiles to be served. When the same trip may start or finish in different stations, its white nodes appear more than once in the flow network. That is the case of both trips 1 and 2. Trip

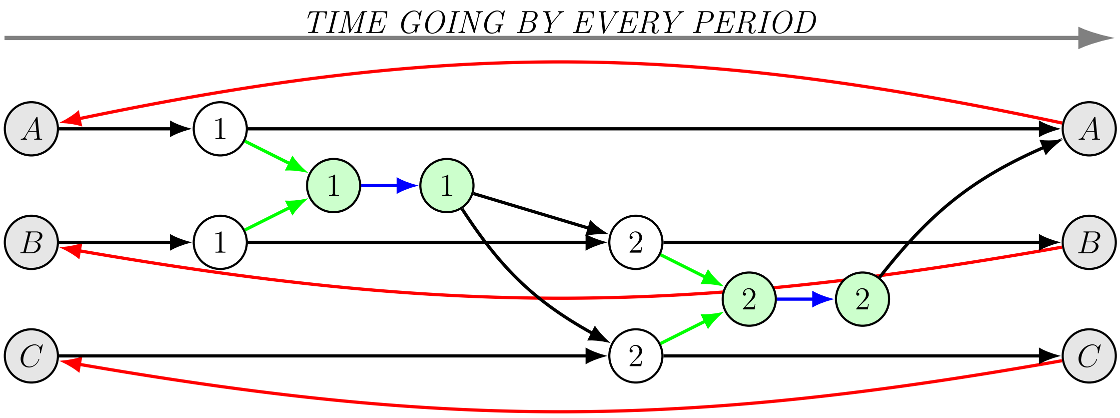


Figure 4.6: Example of Flow Network $\mathbb{G}$

1 may start either at station A or at station B, and may finish at station B or station C. In both stations B and C, the following trip to be served is trip 2. Although the origin of trip 2 is near stations B or C, its destination is near station A only, being the last trip of the period made at that station.

In this flow network, arcs can represent flows with values that are greater than one. Thus, each arc represent a mobility profile of various similar people instead of only one specific person. This allows the LP formulation to be more computationally scalable than other formulations having one variable per client. Since each trip profile may indicate trips starting or finishing in different stations, the CFN must assure that the same individual served trip is not counted multiple times. Therefore, for every set of white nodes there will be two green nodes used to filter the served trips. By doing so, the served trips are funneled through the blue arc, assuring that the correct number of served trips will flow through the network.

Green arcs indicate revenue obtained by serving trips; thus, they yield positive utility values. Blue arcs only assure the funneling of the flow of vehicles, without yielding revenue or cost, and thus, they have utility value equal to zero. The flow that passes by green and blue arcs is constrained by the demand of each trip as an upper bound, avoiding unrealistically profitable scenarios. Black arcs only direct the flow of vehicles after a blue arc, or indicate idle vehicles in the stations. They have utility values equal to zero and do not have a predefined upper bound, since their flow is already constrained by the other arcs.

There is no staff relocating vehicles. All fleet movements are made by the clients, while performing their own trips along the period. Red arcs do not indicate a movement of vehicles. They indicate the costs involved in having vehicles to rent in each station; therefore, red arcs have negative utility values. Red arcs also ensure that the initial number of vehicles at each station will be available at the beginning of every period. It is assured because the red arcs tie the end to the beginning of the period, forming a “cycle” for each station. Red arcs have upper and lower bounds that depends on the business model being evaluated. The CFN network does not describes the exact time when a trip was finished. However, as showed in Figure 4.6, it is possible to notice that every trip certainly has finished before its arc reaches the next node.

Although the CFN problem may also represent the dynamics of an electric fleet when the assumptions mentioned earlier are obeyed, this research does not include scenarios regarding electric vehicles. The decision of not simulating electric vehicles is due to lack of charging infrastructure in Brazil at present, and the high costs of having an electric vehicle in Brazil when compared to combustion engine ones [Yamamura et al., 2022].

Besides, recharging electric vehicles is slower than refueling combustion engine vehicles, and current vehicle batteries have a range of driven distance smaller than the range of gas vehicles with a full tank [Yamamura et al., 2022; Turoń et al., 2022]. It was simulated in Turoń et al. [2022] that using electric vehicles for carsharing may be worse than using combustion engine vehicles. This result was obtained due to the recharging time and high costs for purchasing a fleet of electric vehicles. As discussed in the work, additional vehicles would be needed to replace the electric vehicles while recharging. In this case, electric vehicles will not be shared enough to become the optimal fleet choice.

The next subsection presents the LP formulation to solve the CFN problem.

4.3.1 LP Formulation

Tables 4.10, and 4.11 present, respectively, the variables and model parameters used in this LP formulation. The objective function and constraints are described from Equation 4.15 to Inequality 4.18.

Table 4.10: Variables of the LP Formulation

Variable	Description
$x_{i,j} \in \mathbb{A}$	Arc (i, j) indicating the flow from node i to node j

Table 4.11: Model Parameters of the LP Formulation

Model Parameter	Description
n	Number of nodes
$u_{i,j}$	Utility value of the arc (i, j)
$\lfloor b_{i,j} \rfloor$	Upper bound of the arc (i, j), if applicable
$l_{i,j}$	Lower bound of the arc (i, j), if applicable

Equation 4.15 defines the objective function of this LP. In summary, this function sums up all profits obtained by serving clients (fuelling costs are already deducted) including the vehicle costs (negative utility values from red arcs, which also comprehend the rental of parking slots). Equation 4.16 represents the constraints of flow conservation. These constraints ensure that the total flow incoming to the node is equal to the total flow outgoing this node. By doing so, the rationale of the network is guaranteed, since there is no “leakage” of flow along the network. Equation 4.17, in the case of blue and green arcs, defines the upper bound of the expected number of clients that want to make that trip. Regarding red arcs, that same constraint defines the maximum number of parking slots available to rent in each station. In turn, Equation 4.18 ensures that those scheduled clients previously selected will be served in the mixed-demand business models, namely Mixed Free-Floating and Mixed Partial-Floating; and ensures that all other arcs will not have negative flow.

$$\underset{x_{i,j}}{\arg\max} \sum_{i=1}^{n}\sum_{j=1}^{n} u_{i,j}x_{i,j} \tag{4.15}$$

Subject to:

$$\sum_{j=1}^{n} x_{i,j} - \sum_{j=1}^{n} x_{j,i} = 0 \qquad \forall i = 1, 2, ..., n \tag{4.16}$$

$$x_{i,j} \leq \lfloor b_{i,j} \rfloor \qquad \forall (i,j) \in \mathbb{A} \mid \exists\, b_{i,j} \tag{4.17}$$

$$x_{i,j} \geq \begin{cases} l_{i,j}, & \text{if business model of mixed-demand} \\ 0, & \text{otherwise.} \end{cases} \tag{4.18}$$

This formulation is suitable for optimizing the fleet of the three carsharing business models defined in subsection 3.4.4. They only differ on the values of the right-hand side of the constraints, and if it is necessary to solve the LP twice. Solving the LP model twice is required for the business models of mixed-demand. Since these business models serve on-demand carsharing clients using idle vehicles and parking slots from the scheduled clients, it is only possible to optimize the fleet of such mixed-demand business models after the scenario with scheduled clients is already optimized. By doing so, the constants $l_{i,j}$ are obtained from the first optimization run, and used in the second (final) optimization run.

This LP formulation does not consider maintenance operations. These tasks are expected to be made by third parties or by some clients rewarded with benefits such as price discounts. The following subsection demonstrates that this LP is Totally Unimodular, and consequently, the CFN problem belongs to P, the class of problems solvable in polynomial-time.

4.3.2 CFN belongs to P

The simple structure of the proposed LP allowed its matrix of constraints to be Totally Unimodular (TU), making it possible to solve the CFN problem in polynomial-time. That happens because as the matrix of constraints is TU and the vector of constants b of all constraints have an integer value, all optimal solutions found by the LP will have only integer variable values. Therefore, it is not necessary to run algorithms like Branch-and-Bound to make the solutions integral [Tamir, 1976; Wolsey, 1998; Bazaraa et al., 2010].

A matrix of constraints is TU if all determinants of its square sub-matrices are -1, 0 or 1. It can be assured that, regardless of the instance, the matrix of constraints for this LP will only have such determinants because it follows sufficient conditions presented in Wolsey [1998]. These conditions are:

- every coefficient of the matrix is one of these elements: $\{-1, 0, 1\}$;
- every column contains at most two nonzero coefficients;
- the matrix can be divided in two sets of lines $\mathbb{L}_1$ and $\mathbb{L}_2$ so that the column-based sum of $\mathbb{L}_1$, subtracted by the column-based sum of $\mathbb{L}_2$, results in a line composed by only zeros. Equation 4.19 formalizes this condition.

$$\sum_{i \,\in\, \mathbb{L}_1} a_{i,j} - \sum_{i \,\in\, \mathbb{L}_2} a_{i,j} = 0 \qquad \forall j \in \{1, 2, ..., m\} \tag{4.19}$$

The flow conservation constraints defined by Equation 4.16 satisfy these conditions because every arc appear twice: one time with value 1 and another time with value -1. Therefore, every coefficient belongs to $\{-1, 0, 1\}$, there are no more than two nonzero coefficients in each row, and the sum of them results in zero.

The last two constraints, represented by Inequalities 4.17 and 4.18, maintains the total unimodularity of this LP because they are equivalent to identity matrices. Since one matrix TU concatenated to an identity matrix results in another matrix TU [Wolsey, 1998], the LP including Inequalities 4.17 and 4.18 is also TU. Consequently, if the right-hand side (vector b) has only constants, this proposed LP can be solved in polynomial-time, and the CFN problem belongs to P. Next subsection presents the algorithm used to discover which street intersections are nearby each other and which stations are close to the places visited by the possible clients.

4.4 Shortest Distance Algorithm

The SSMS, SASS and CFN problems depends on the shortest distance between street intersections or stations to build their optimization models. A reasonable method for discovering which locations are nearby each other consists in calculating the shortest distance between all pairs of vertices from the road network and selecting only those with distance shorter than D.

One option consists in running the Floyd-Warshal algorithm, with time complexity order of $O(\mathbb{V}^3)$ [Cormen et al., 2009], and then filtering the pairs of vertices with distance up to D, spending an additional time complexity of $O(\mathbb{V}^2)$. This will result in a total complexity time of $O(\mathbb{V}^3+\mathbb{V}^2)$. Another option consists in running the Dijkstra algorithm for all vertices, having total time complexity order of $O(\mathbb{V}^2 log \mathbb{V} + \mathbb{E}\mathbb{V})$ [Cormen et al., 2009], and discovering which pairs have distance shorter than D while the shortest distance is calculated.

In this case, the Dijkstra algorithm for all vertices is expected to perform faster than the Floyd-Warshal algorithm. That is expected because vertices in pedestrian ways and road networks usually have low degree, and the D threshold can be applied to interrupt iterations of search in the graph. Algorithm 1 presents this modification of the Dijkstra's shortest path algorithm by halting the search in the graph depending on the distance reached so far, checked at line 19. An array L is also used to avoid re-initializing all distances to infinity. Provided that only a small subset of vertices are expected to be reached due to the threshold D, L stores the vertices reached so far, allowing the "for" loop at line 3 to only iterate over vertices reached by the last source vertex.

The first lines of Algorithm 1 initialize L, the array *distances* and the heap Q. Dijkstra's shortest path search goes from line line 9 to line 20. Edges are included in graph $\mathbb{D}$, at line 24, after the search have been finished for each source. The edges' length are set at line 25. The time complexity of this algorithm keeps being $O(\mathbb{V}^2 log \mathbb{V} + \mathbb{E}\mathbb{V})$, the same as simply running the Dijkstra's shortest path for each vertex.

After running Algorithm 1, the data on the returned graph $\mathbb{D}$ is stored in tables "VerticesPairsNearby" and "StationsNearbyPlaces", presented in section 5.2. Shortest distance calculations are also useful to define the utility value of edges in SSMS and SASS and to estimate the monthly parking costs for solving the CFN problem. Next chapter explains how this work was planned to be developed.

Algorithm 1: AllPairsNearby($\mathbb{G}$: graph, D: real)

Input: Network of pedestrian ways and the places of trip composing the graph $\mathbb{G}$, and minimum distance allowed between edges D

Output: Graph $\mathbb{D}$ in which vertices nearby each other are adjacent, and edges' lengths are the shortest distances between vertices

```
L ← 𝕍;
for each source in 𝕍 do
    for each v in L do
        distances[v] ← ∞;
    distances[source] ← 0;
    L ← {source};
    Q ← ∅;
    Insert(Q, ⟨0, source⟩);
    while Q ≠ ∅ do
        u ← ExtractMin(Q);
        for each e ← (u, v) in 𝔾.Adj[u] do
            if distances[v] > distances[u] + e.length then
                distances[v] ← distances[u] + e.length;
                if v ∉ L then
                    L ← L ∪ v;
                    Insert(Q, ⟨distances[v], v⟩);
                else
                    DecreaseKey(Q, ⟨distances[v], v⟩);
        if distances[u] ≥ D then
            break;
    𝔻.𝕍 ← 𝔻.𝕍 ∪ source;
    for each v in L do
        𝔻.𝕍 ← 𝔻.𝕍 ∪ v;
        𝔻.𝔼 ← 𝔻.𝔼 ∪ ⟨source, v⟩;
        𝔻.𝔼.⟨source, v⟩.length ← distances[v];
return 𝔻;
```

Chapter 5

Resources and Methods

This chapter describes the computational resources and models used to solve the problems described in the last chapter. The hardware and software specifications are described in the following section, the conceptual schema of the spatial database used is presented in section 5.2 and the data preprocessing tasks are described in 5.3.

5.1 Computational Resources

The optimization problems and the simulations run in a computer with the specifications described in Table 5.1, and using software listed in Table 5.2.

Table 5.1: Hardware Specifications

Resource	Specification
Model	Apple Mac mini (Late 2012)
Processor	Intel Core i7 2.3 GHz, 4 cores
RAM	16 GB, DDR3 1666 MHz
Hard Disk	1 TB, 5400 RPM

The next section presents the diagram of the spatial database used in the experiments, and summarizes the trips dataset used in this work.

5.2 Conceptual Schema

The conceptual schema of the database used in this work was created following the Object Modeling Technique for Geographic Applications (OMT-G), proposed by Borges

Table 5.2: Software Specifications

Resource	Specification
Operational System	macOS Catalina 10.15.6
Database Management System	PostgreSQL 13.2
Spatial Database Extension	PostGIS 3.1
Scripting Language	Python 3.7.4
Graph Algorithms Library	NetworkX 2.5
Parallel Data Analysis Library	Koalas 1.8.0
Optimization solver	Gurobi 9.1.2, Individual Academic License

et al. [2001]. The diagram was materialized using the OMT-G Designer tool, proposed by Lizardo and Davis Jr. [2014, 2017] and available online[1]. The data definition and data integrity statements were also generated by OMT-G Designer.

The road network data was obtained from OpenStreetMap[2] on August 8, 2021 and the transport demand was obtained from the Origin-Destination Survey of São Paulo[3], performed along the years of 2017 and 2018. This survey has produced a dataset composed by 183,092 rows and 126 columns. Each row indicates a trip made by a family member of the interviewed person. Each column indicates attributes about the family, about the person in the trip, or about the trip itself. The metadata about these 126 attributes is presented in the Annex, at Table A.1.

In summary, this dataset presents attributes regarding the socioeconomic situation of families and characteristics of their trips. The data about trips are: their purpose, mode of transport, parking cost (if applicable), location and timestamp of origin and destination, and identification within the same dataset of which persons of the family shared the vehicle during the trip (if applicable). Some visualizations and queries using this dataset can be made on the Ciclocidade's public dashboard[4].

Figure 5.1 shows the OMT-G class diagram. Classes "StreetIntersection" and "StreetSegment" compose the network of pedestrian ways of the São Paulo Metropolitan Area. The attributes on "StreetSegment" indicate an id for each street segment, its origin and destination vertices, its length in meters, a textual flag from OpenStreetMap describing the driving right-of-way of this street segment, the utility value used in SSMS and SASS problems, and the estimated monthly parking costs for one vehicle in this street segment. The "idEdge" attribute is needed instead of using the origin and destination vertices as a primary key because some pairs of vertices in the road network have more than one edge between them.

[1] http://aqui.io/omtg-designer/

[2] https://www.openstreetmap.org/relation/2661855

[3] http://www.metro.sp.gov.br/pesquisa-od/

[4] https://public.tableau.com/profile/ciclocidade#!/vizhome/OD2017/MAPA

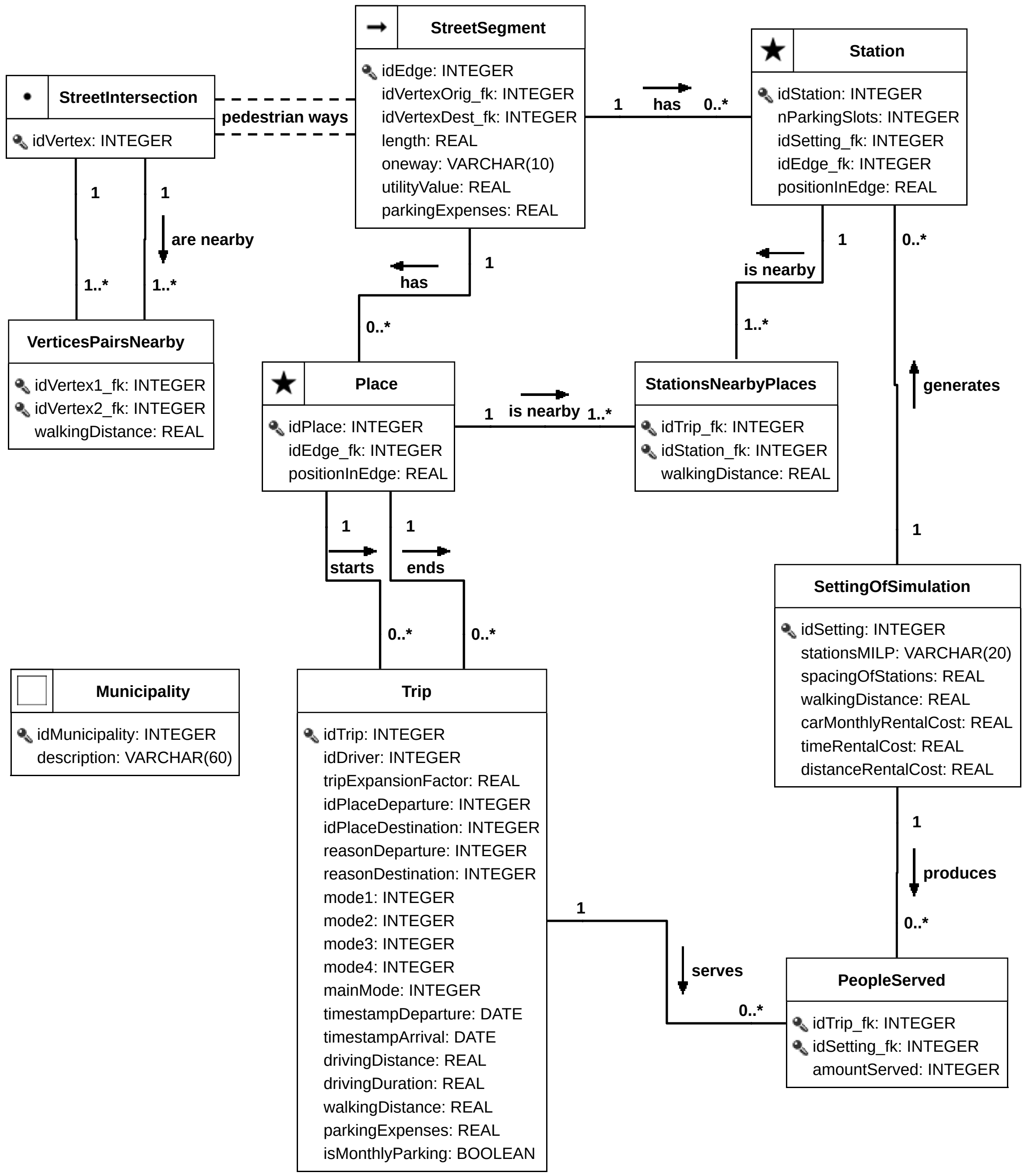


Figure 5.1: OMT-G Diagram

Class "VerticesPairsNearby" indicates which pairs of street intersections are nearby each other. Having this information materialized in the database instead of calculated dynamically makes it easier and faster to preprocess the input data for solving the SSMS problem. In this work, pairs of street intersections for which the streetwise shortest distance between them is larger than 500 meters are not included in the "VerticesPairsNearby" class, because they are not considered to be spatially close. Since this class is about walking distances, the driving right-of-way is not considered while calculating the shortest distance between the pairs of vertices. Class "Municipality" defines the frontiers of the municipalities within the São Paulo Metropolitan Area.

Class “Station” maintains the data about the stations generated by SMSS or SASS and their attributes used in the simulations of CFN. Its attributes are, respectively: the station identification, the number of needed parking slots calculated by the CFN simulation, the id of the simulation setting which generated that station with such properties, the id of the edge in which this station is positioned, and the position of this station at the edge. This position is a real number varying from zero to one, where zero means that the station is placed exactly at the edge’s origin vertex, one means that the station is placed exactly at the edge’s destination vertex, and any intermediate value means that the station is positioned at that proportion of the edge length.

All drivers included in the dataset have their origin and destination places stored in the “Place” class. The places are stored regardless of which transport mode that person used for that trip in the day of the interview. In this case, we assume that if some person is able to drive and is the driver of at least one of his/her trip demands, this person may also be flexible to drive in his/her other trip demands. The “Place” class includes the place identification, the id of the closest street segment, and also stores what position in this street segment is the closest to the stored place, similarly to the “positionInEdge” attribute of the “Station” class described previously.

Class “StationNearbyPlaces” references the stations near the origin and destination trip locations stored in the “Place” relation. Similarly to the “VerticesPairsNearby” relation, the threshold for two locations to be considered nearby is 500 meters. By doing so, every place will be at most about 250 meters away from a station (250 meters of street segments to its left and another 250 meters of street segments to its right). This relation makes it easier to load data for the CFN simulation. Using this relation, it is possible to easily obtain the generated stations that are close to each origin or destination place.

The important data about each trip of the dataset is stored in the “Trip” class. This class holds the identification of the trip, identification of the driver, the expansion factor of the trip, the reference to the places of departure and destination, the purpose of the driver attending the origin and the destination places (such as going from home to work), up to four transport mode used by the person to get to his/her destination, the main transport mode used in this trip, the time of departure and arrival, the driving distance calculated as the shortest distance according with the driving right-of-way, the duration of that trip considering the average speed of other trips on the same weekday and hour, the walking distance as the shortest path not considering the driving right-of-way, the driver’s expenses with parking, and a flag indicating if the parking expenses are regarding fixed monthly or other parking costs.

Except for the attributes “drivingDistance”, “drivingDuration”, and “walkingDistance”, the data used in the “Trip” class comes from the dataset used, whose metadata is presented on Table A.1, in the Annex. The identification of the driver was obtained from the column 44, after selecting only people who drive a car. This selection is made by removing all people from the dataset which do not have at least one line with column 107 having value 9, indicating that this person drives an automobile. The trip's expansion factor, extracted from column 80, indicates how many trips from the general population are represented by this sample trip. This trip's expansion factor is used in Inequality 4.17 as the upper bound for the number of similar clients that can be served, and in the objective function of SSMS and SASS formulations as the utility value of the street segments.

Since clients are expected to be flexible to walk short distances before or after their trips, the utility value used when stations are allocated in SSMS or SASS is calculated according with the demand nearby a street segment. To do so, the utility value of a street segment is calculated as the sum of expansion factors from trips that have started or finished in that street segment or in nearby segments. In this case, a street segment is considered near a place of origin or destination if it can be entirely traversed in 500 meters of walk from the place of origin or destination. Considering as nearby only those street segments entirely traversed in 500 meters avoids setting utility values for lengthy street segments with no demand of trips. By doing so, the utility values are concentrated either on the exact street segment of the trip, or on the specific surroundings of the trip origin or destination.

The estimated monthly parking expenses are calculated as the average of monthly parking costs in nearby street segments. The concept of nearby street segments here is the same as in the utility value calculation, considering only street segments that can be entirely traversed in 500 meters. If a street segment has an utility value but there is no data on monthly parking costs nearby, its estimated monthly parking expenses is set as the same as the closest street segment with an average monthly parking expense calculated.

In the "Trip" class, the coordinates (x and y) of places where trips started were obtained from columns 85 and 86, and the coordinates (x and y) where trips finished were obtained from columns 89 and 90. The transport modes was extracted from columns 107 to 110, and the main transport mode was obtained from column 118. The departure times (hours and minutes) come from columns 111 and 112, and the arrival times (hours and minutes) were obtained from columns 114 and 115. The parking expenses value was extracted from column 122 and the flag indicating if the parking was paid as a fixed monthly fee was obtained from column 121.

Class "SettingOfSimulation" records the parameters used in the generation of stations and the CFN simulation. Similarly to the other relations, an id column is also defined for each setting. The parameters used to generate stations are a brief description of the MILP formulation used, and which distance threshold D was used. The CFN parameters were the maximum distance clients are flexible to walk to get to a carsharing station (first-mile) or to get to their destination (last-mile), the expected cost the carsharing company will have per monthly rented car, how much the carsharing clients will be charged per minute, and how much they will be charged per driven kilometer while using the carsharing service. Trips that can be served according to some simulation setting are stored in the “PeopleServed” class. In this class, the attribute “amountServed” holds the number of similar trips, limited by the person’s expansion factor, that could be served according with the parameters defined in the “SettingOfSimulation” class.

A physical schema is generated from the conceptual schema in OMT-G by OMT-G Designer for PostGIS. The tool automatically produces data definition statements in SQL, along with assertions and triggers that implement all spatial integrity constraints contained in the class diagram. Primary and foreign key constraints are also created, following the indications in the OMT-G class diagram. Additionally, a GiST spatial index was generated for every spatial column defined, and a B-Tree index was generated for every primary key. In the tables created from classes “VerticesPairsNearby” and “StationsNearbyPlaces”, the B-Tree indices were generated as multi-columns, composed by the first column of the primary key and the “walkingDistance” column. By doing so, queries using both columns, such as loading stations nearby a specific place but up to x meters distant, can benefit from the multi-column index[5].

Next section describes the preprocessing tasks performed before running the optimization methods and the instance of pedestrian ways network.

5.3 Data Preprocessing

The first database tables to be populated were “StreetIntersection” and “StreetSegment”. The data of both tables had to be preprocessed in order to avoid redundancy or wrong shortest path distance calculations. The following subsection describes the preprocessing tasks applied.

5.3.1 Data Cleaning

The network of pedestrian ways obtained from OpenStreetMap contains subsets of vertices that cannot be reached from any other edges and vertices of the rest of the network, even when considering undirected edges for walking distances. After manually

[5]https://www.postgresql.org/docs/13/indexes-multicolumn.html

analyzing some samples of such vertices, we verified that these are streets inside gated communities, private properties or anomalies caused by missing street segments missing on OpenStreetMap. These cases generated smaller unconnected components in the network of pedestrian ways, making it impossible to find paths and distances between locations in disconnected components of the graph.

The total number of street intersections is 364,521. The largest component of the network of pedestrian ways has 323,736 street intersections, equivalent to 88.81% of the total amount. The other 40,785 street intersections compose 6,521 smaller graphs. Among these smaller components, the largest one (which is the second largest if considering all components) had 2,142 street intersections, equivalent to 0.59% of the total amount. After a manual analysis using Google Maps[6], it was found that this component (second largest among all) was located in the Terra Preta district, inside the municipality of Mairiporã, about 50 km to the North from São Paulo's downtown. Most of the street segments in this component are not paved. The second largest component had 828 street intersections, equivalent to only 0.23% of the total, and was located in an urban area inside the São Paulo's municipality, in the Jardim Gaivotas neighborhood. This neighborhood is in an opposite direction of the Terra Preta district, being 30 km to the South from downtown São Paulo, and about 80 km distant to Terra Preta. The other 6,519 smaller components are also spread along the São Paulo Metropolitan Area.

As mentioned before, these smaller components are streets inside gated communities, private properties or were caused by missing street segments on OpenStreetMap. Due to their private nature, street segments inside gated communities and other private properties are not realistic locations for having shared mobility services publicly available for anyone nearby. Therefore, those street segments are not expected to receive shared mobility stations and thus should be skipped from the optimization tasks of this work. Regarding the missing street segments in OpenStreetMap, their absence will likewise not cause significant impact to the data. That happens because the origin and destination locations occur either on the largest component of the graph, or close to an edge that belongs to the largest component of the graph. Therefore, deleting every street segment and street intersection that do not belong to the largest component will not significant impact the overall graph since other nearby streets that belong to the largest component can be used instead.

Besides deleting the smaller components, other four preprocessing tasks were applied to the graph of pedestrian ways. These tasks were: deleting street segments with length equal to zero, skipping segments not nearby any origin or destination of trips in order to reduce the number of variables in the optimization of stations location, splitting lengthy street segments into shorter ones for solving the SSMS MILP (assumption mentioned in subsection 4.1.3), and splitting lengthy street segments for increasing the SASS

[6]https://google.com/maps

precision. Table 5.3 presents the variation of the number of street segments after each preprocessing task.

Table 5.3: Number of Vertices and Edges After Preprocessing Tasks

Preprocessing Task	Vertices		Edges	
	Number	Variation	Number	Variation
Original Number	364,521	-	454,680	-
Deleting Smaller Components	323,736	-11.19%	414,234	-8.90%
Deleting Edges with Length = 0	323,736	0%	414,188	-0.01%
Skipping Far Vertices and Edges	240,874	-25.60%	285,186	-31.15%
Splitting Edges for SSMS	252,348	+4.76%	296,660	+4.02%
Splitting Edges for SASS	2,807,548	+1,012.57%	2,850,186	+861.32%

The task of splitting lengthy street segments into smaller ones was performed using Algorithm 2. The following subsection discusses this algorithm and presents statistics of its use on the network of pedestrian ways.

5.3.2 Splitting Lengthy Street Segments

Algorithm 2 was applied to split edges for both problems SSMS and SASS. The algorithm takes as arguments the $\mathbb{G}$ graph to have its edges split, and the distance D (optional) for limiting the splits when the maximum length of the graph reaches the D threshold. The D parameter is useful to avoid unnecessary splits for SSMS instances, since the SSMS problem does not benefit from edge lengths that are much shorter than the maximum allowed distance between stations. Algorithm 2 can be divided into three major steps:

1. initializing a priority queue using the length of edges in $\mathbb{G}$, from line 1 to 3;
2. iteratively decreasing the maximum value in the heap by dividing it at maximum $nIterations$ times, from line 4 to 10;
3. changing $\mathbb{G}$ by removing the lengthy edges and replacing them with shorter ones, from line 11 to 23.

Since the number of new edges is limited to nine times the number of edges, as defined in line 4, and it is possible to perform the operations Insert, FindMax and

DecreaseKey in $O(1)$ time by using a Fibonacci Heap, Algorithm 2 has time-complexity of $O(\mathbb{E})$.

Algorithm 2: SplitStreetSegments($\mathbb{G}$: graph, $D = 0$: real)

Input: Network of pedestrian ways $\mathbb{G}$, and minimum distance allowed between edges D

Output: Graph $\mathbb{G}$ with lengthy edges split into shorter ones

$Q \leftarrow$ InitializeMaxHeap();
for *each e in* $\mathbb{G}.\mathbb{E}$ **do**
    Insert($Q, \langle e.length, 1, e\rangle$);
$nIterations \leftarrow 9 \times |\mathbb{G}.\mathbb{E}|$;
for $i \leftarrow 1$ **to** $nIterations$ **do**
    $\langle oldLength, nSplits, e\rangle \leftarrow$ FindMax(Q);
    if $oldLength \leq D$ **then**
        break;
    $nSplits \leftarrow nSplits + 1$;
    DecreaseKey($Q, \langle e.length/nSplits, nSplits, e\rangle$);
for *each tuple in Q* **do**
    $\langle lengthSplit, nSplit, e\rangle \leftarrow tuple$;
    if $nSplit \geq 2$ **then**
        $\langle source, destination\rangle \leftarrow e$;
        $\mathbb{G}.\mathbb{E} \leftarrow \mathbb{G}.\mathbb{E} - e$;
        for $i \leftarrow 1$ **to** $nSplit - 1$ **do**
            $newV \leftarrow createVertex()$;
            $\mathbb{G}.\mathbb{V} \leftarrow \mathbb{G}.\mathbb{V} \cup newV$;
            $\mathbb{G}.\mathbb{E} \leftarrow \mathbb{G}.\mathbb{E} \cup \langle source, newV\rangle$;
            $\mathbb{G}.\mathbb{E}.\langle source, newV\rangle.length \leftarrow lengthSplit$;
            $source \leftarrow newV$;
        $\mathbb{G}.\mathbb{E} \leftarrow \mathbb{G}.\mathbb{E} \cup \langle newV, destination\rangle$;
        $\mathbb{G}.\mathbb{E}.\langle newV, destination\rangle.length \leftarrow lengthSplit$;
return $\mathbb{G}$;

Table 5.4 presents the results of applying the Algorithm 2 on the graph of pedestrian ways in São Paulo. The second column describes the initial distribution of edge lengths. The third column describes the edge lengths after running Algorithm 2 with $D = 200$, for solving the SSMS problem. The fourth column presents the edge lengths when only $|\mathbb{E}|$ splitting iterations were run, making the total number of edges become the double of the initial amount. The fifth column describes the edge lengths when $4\times|\mathbb{E}|$ splitting iterations were run, making the total number of edges become five times the initial. And the last column describes the edge lengths distribution when all $nIterations = 9\times|\mathbb{E}|$ were run, making the number of edges become ten times the initial.

Next chapter presents the experimental results of this work.

Table 5.4: Distribution of Edges' Length (in meters) After Splitting

			Number of Splits		
Statistics	Initial	$D = 200$	$1 \times \|\mathbb{E}\|$	$4 \times \|\mathbb{E}\|$	$9 \times \|\mathbb{E}\|$
Minimum	0.02	0.02	0.02	0.02	0.02
Q_1	28.04	28.04	25.67	12.43	6.64
Median	58.42	58.42	34.32	14.07	7.08
Mean	71.22	65.93	31.91	13.22	6.78
Q_3	96.56	96.56	41.05	14.99	7.30
Maximum	6,703.57	200.00	49.08	15.82	7.50
Standard Deviation	66.79	45.73	11.86	2.59	0.85
Number of Edges	285,186	296,660	570,372	1,425,930	2,851,860

Chapter 6

Experimental Results

This chapter presents the experimental results from optimizing the stations location, and simulating the carsharing business models. The following section discusses the solutions and run times for solving the SSMS and SASS problems.

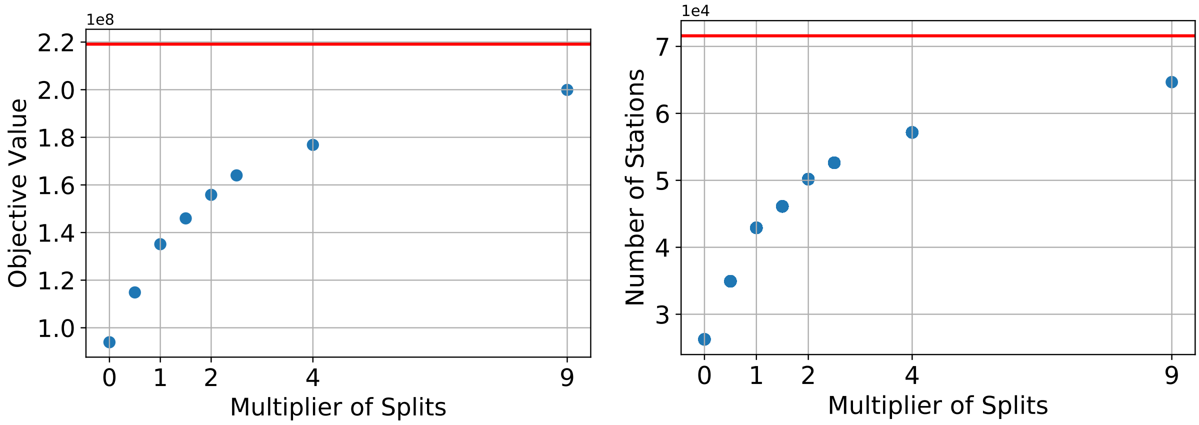
6.1 Optimization of Stations Location

As defined in Chapter 4, the objective function of both SSMS and SASS is simply the sum of the utility values got by placing stations on the street segments. In this work, the utility value of a street segment is the expected number of trips that starts or finishes nearby that street segment. This demand of trips was obtained from the Origin-Destination Survey of São Paulo, considering only the trips taken by someone who drives. And it was considered that two street segments are nearby each other if the shortest distance between them is shorter than or equal to the maximum distance carsharing clients are flexible to walk.

In this work, that maximum distance was set to 200 meters. By doing so, the more trips are nearby a street segment, the greater is its utility value. The following subsection presents the best solutions found for both SSMS and SASS problems.

6.1.1 Best Solutions Found

Figures 6.1 (a) and (b), shows respectively, the difference between the objective values and number of stations obtained by optimizing the SSMS and SASS problems. Both ordinate axis presents values in scientific notation due to high values expressed. Regarding Figure 6.1 (a), the red line indicates the objective value of the best solution found by optimizing the SSMS problem. This objective value is 2.1912 e+08, about 9.61% higher than the optimum solution for SASS with $9 \times |\mathbb{E}|$ splits made. A similar patter was found on Figure 6.1 (b), where SSMS obtained 71,585 stations while SASS with $9 \times |\mathbb{E}|$ splits got 64,669 stations, that is about 9,66% fewer stations.



(a) Optimal solution for SASS (points) and SSMS (red line)

(b) Number of stations for SASS (points) and SMSS (red line)

Figure 6.1: Objective Values and Number of Stations by solving the MILP models

This similarity was present not only between the SSMS solution and the greatest SASS solution, but also along all optimum solutions got from SASS. Such pattern was expected since all presented SASS solutions has optimality guarantee, and thus, they are already the best combination of edges for the given instance. Thus, increasing the precision of SASS with more splits will allow SASS to avoid wasting space, and then, allow selecting more edges to have stations, increasing its objective value. Although the SASS solutions have optimality guarantee, with a gap of 1e-4 (0.0001), the SSMS optimization was interrupted with a gap of 14% from the upper bound. Therefore, it is possible that the SSMS problem can achieve even better solutions. The following subsection discusses the run times of these optimizations.

6.1.2 Run times

Both SSMS and SASS with $9 \times |\mathbb{E}|$ splits took about one week of computer processing to obtain the solutions presented in Figure 6.1. The fact that all evaluated scenarios of SASS were solved to optimality, including the scenario with $9 \times |\mathbb{E}|$ splits (ten times more edges than original), was expected since the SASS's formulation is simpler than the SSMS's formulation and do not uses Big-M in the constraints. Although the more splits are made the better solutions SASS finds, the objective value gained by increasing the number of splits reduces as new splits are made. Besides, as increasing the splits also increases the instance size, solving SASS with a high number of splits becomes cumbersome due to the amount of memory needed to hold the model.

Figure 6.2 and Table 6.1 presents, respectively, the boxplots and descriptive statistics of the time spent by solving the SASS MILP model with multiplier of splits up to 2.5. This multiplier threshold of 2.5 was chosen empirically as the highest multiplier yielding an instance that fits in the 16 GB of RAM of the computer used, without requiring swap operations. Every presented boxplot comprises 40 run times. These run times had a low standard variation, making the boxes of Figure 6.2 (b) very narrow.

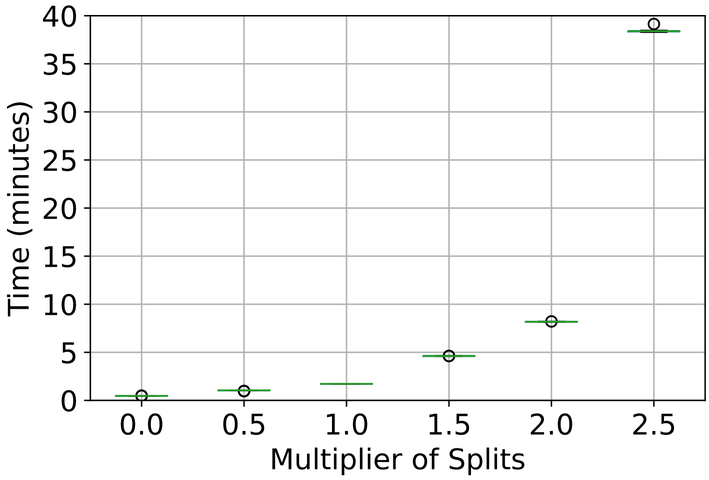


Figure 6.2: Run time for SASS

Table 6.1: Statistics of time spent (in seconds) for solving the SASS

	Multiplier					
Statistics	0.0	0.5	1.0	1.5	2.0	2.5
Min.	28.484	58.458	103.145	490.095	276.287	2298.301
Q_1	28.572	62.770	103.335	490.645	276.541	2301.350
Median	28.603	62.884	103.457	491.078	276.666	2303.406
Mean	28.621	62.557	103.443	491.074	276.743	2304.276
Q_3	28.637	62.956	103.534	491.352	276.844	2305.101
Max.	28.955	63.065	103.680	492.678	277.465	2348.290
S. D.	0.085	1.175	0.140	0.518	0.248	7.496

No boxplot among the evaluated scenarios shared the same range of values. Therefore, there is significant difference between the run time of the scenarios evaluated [Krzywinski and Altman, 2014]. This difference is maintained even considering multiplier values close each other, such as 0.0 and 0.5. Thus, varying the multiplier parameter impacts the solver run time.

The following section presents the carsharing simulations for the whole Metropolitan Region of São Paulo.

6.2 Large-Scale Simulations

This section presents the large-scale simulations of rental prices and maximum distances that clients would be willing to walk. All results shown correspond to the global optimum solution for the applied parameters setting. Therefore, the optimization model made the most from the available vehicles, parking slots, allowed walking distances and expected demand to maximize the company's profits.

Figure 6.3 and 6.4 are regarding the Scheduled Free-Floating. This business model was chosen because it is the simplest among the evaluated ones. Therefore, it will allow assessing the business viability with the lowest possible rental prices. Figure 6.3 presents the monthly expected profits of a carsharing company. These profits are based on different proportions of the Uber base price used to define the rental, and varying maximum distances walked by clients.

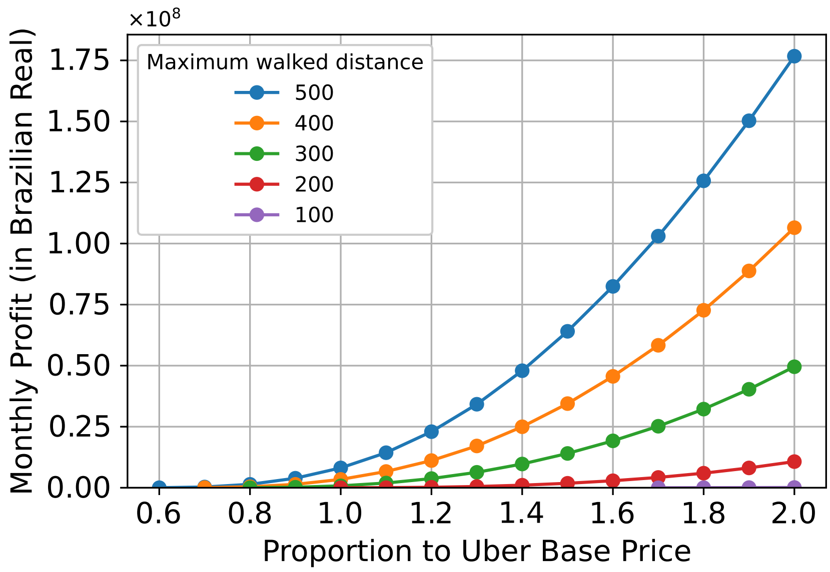


Figure 6.3: Expected profits on the Scheduled Free-Floating

Even on Scheduled Free-Floating, no scenario was profitable considering a maximum walking distance of 100 meters, and prices lower or equal to the Uber base price. The profitable scenario with cheapest prices considered up to 500 meters of walking and 60% of the Uber base price. This scenario is expected to yield R$16,457.24 per month for operating a fleet of 102 vehicles. Even if all clients are willing to deliver the rented vehicles to gas stations for refueling without requiring additional benefits, the monthly profit of only R$16,457.24 provides a small margin for covering other eventual costs, such as cleaning and maintaining the 102 vehicles.

Therefore, this scenario was not the chosen one for the more detailed analyses presented in this work. The more detailed analyses regarding the Scheduled Free-Floating were based on the scenario with also up to 500 meters of walking distance, but charging 70% of the Uber base price. This scenario is expected to yield a monthly profit of R$304,801.88 for operating 720 vehicles. Although the number of vehicles in this scenario increased by 7.06 times, the monthly profit increased by 18.52 times. Thus, the company will have a profit margin to cover eventual costs and clients will pay 30% less when compared to the Uber base price.

Figure 6.4 presents the percentage of trips that would be served by each carsharing rental price evaluated. The chosen scenario for Scheduled Free-Floating is expected to serve about 0.92% of the possible trips. No scenario, even considering the ones charging higher prices, could serve more than 28% of the possible trips. Although the percentage of served demand is small, the subset of trips selected to be served represents well the population of possible carsharing clients.

Figure 6.4: Percentage of served trips on the Scheduled Free-Floating

Figure 6.5 and Figure 6.6 illustrate, respectively, the location of stations and the locations where trips start in the chosen Scheduled Free-Floating scenario. The number of places where trips start is higher than the number of stations because these simulations are based on free-floating mode. Therefore, although every vehicle has its parking slot and comes back to it when the week ends, any vehicle is free to be used in any trip at any place within the studied area along the week.

The colors used in both Figure 6.5 and Figure 6.6 indicate ranges of the number of occurrences at that location. Each class has about the same number of occurrences. There

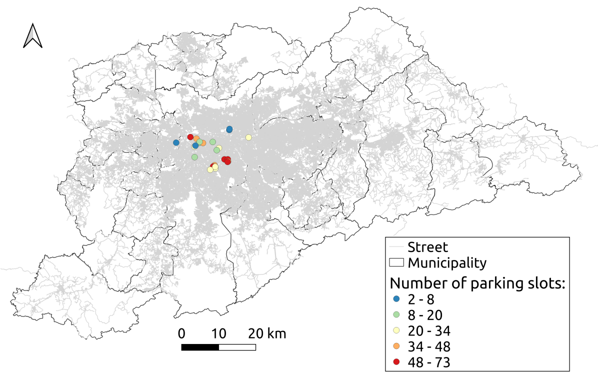


Figure 6.5: Parking slots in the Scheduled Free-Floating

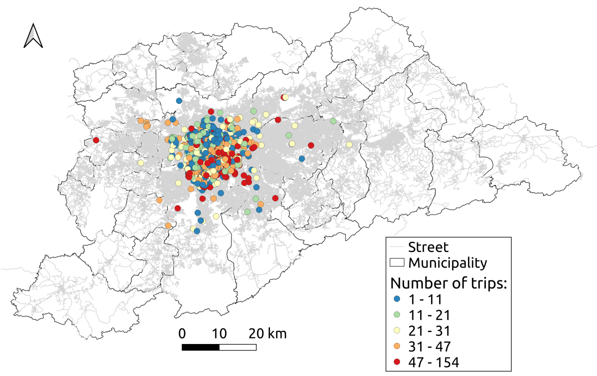


Figure 6.6: Locations of trips starting in the Scheduled Free-Floating

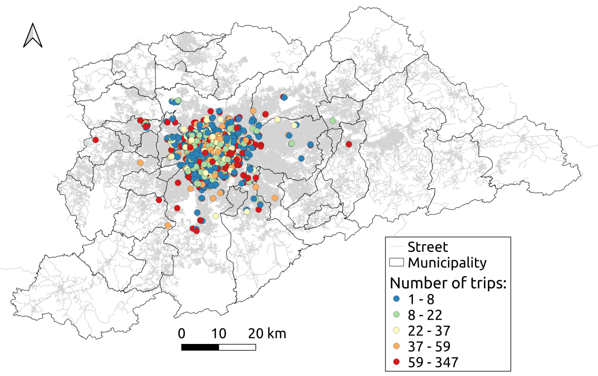


Figure 6.7: Parking slots in the Mixed Partial-Floating

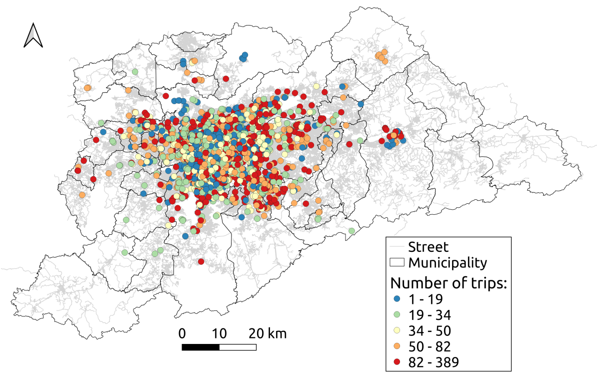


Figure 6.8: Locations of trips starting in the Mixed Free-Floating

is no clear pattern of colors found exclusively in specific regions of the map. However, classes depicted in red are more frequent in São Paulo's central zone. This is expected due to the higher demand of short trips in that region. Another pattern shown in Figures 6.5 and 6.6 is the concentration of selected trips near downtown. Such pattern happens because of the short-term nature of carsharing rentals, that is associated to trips that are more frequent in areas close to downtown. Thus, offering the service in regions with a lower flow of vehicles requires higher rental prices to make the service profitable.

Figure 6.7 and 6.8 illustrate the locations of trips starting in the two mixed-demand business models. In these both scenarios, the scheduled trips were also charged 70% of the Uber base price, but the on-demand trips were charged 90% of Uber base price. Therefore, all these three scenarios charged less than what Uber would usually charge. Except for the pattern of higher concentration of trips in the São Paulo's central zone, all other patterns from Figures 6.5 and 6.6 are also observed in Figures 6.7 and 6.8. Besides, it is clearer on Figure 6.8 that small clusters of locations for starting trips are formed in cities far from São Paulo's downtown.

As matter of comparison, Figures from 6.9 to 6.12 present the location of every trip demand (served and not served) grouped by quarter of the year. Although the quarters have different number of trip demands, any quarter has more trip demands than

Figure 6.8, which has trip demand of all quarters together.

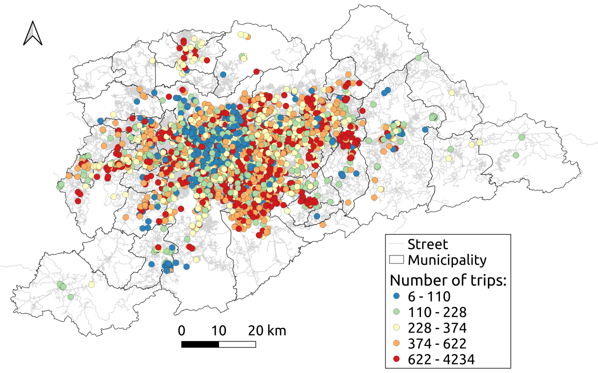


Figure 6.9: Locations of trips starting on the first quarter of the year

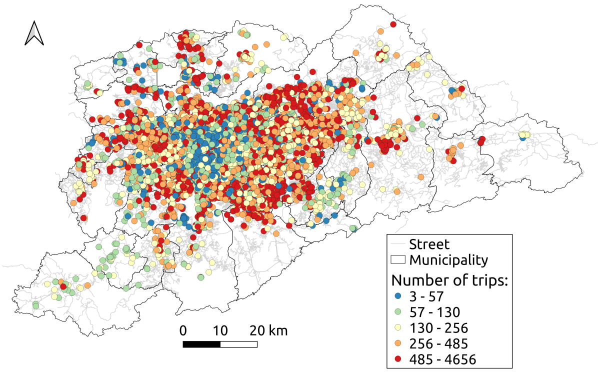


Figure 6.10: Locations of trips starting on the second quarter of the year

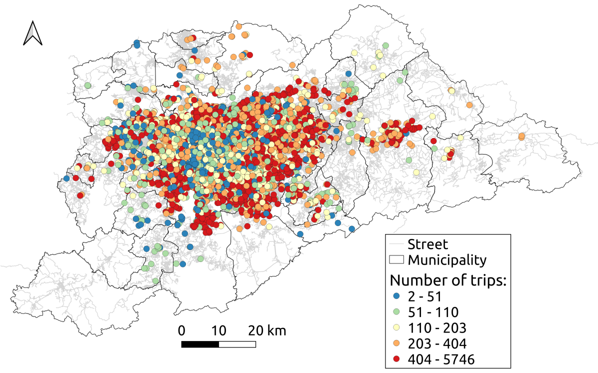


Figure 6.11: Locations of trips starting on the third quarter of the year

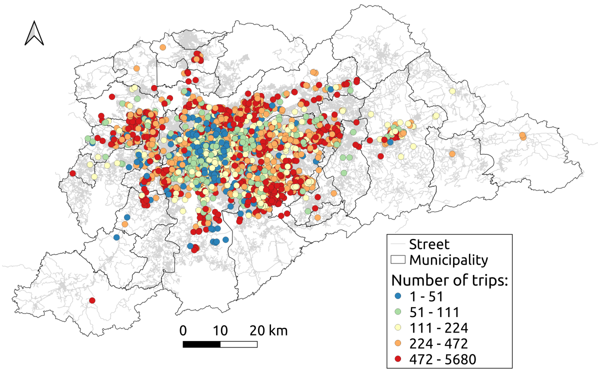


Figure 6.12: Locations of trips starting on the fourth quarter of the year

The following subsection compares the evaluated business models.

6.2.1 Business models comparison

Although carsharing rentals are expected to be cheaper than ridesourcing [Schwieterman and Bieszczat, 2017; Monteiro et al., 2021a], it is possible that carsharing clients allow paying more than the Uber base price in some situations:

1. the clients demand for trips is higher than the local supply of Transport Network Companies (TNC) drivers at that moment;

2. the client demand for trips in a region is so low that there are no TNC drivers nearby;

3. carsharing clients are not flexible to wait for a driver from a TNC company to accept the trip;

4. carsharing clients feel safer or more comfortable when driving by themselves than having a stranger as driver.

The increase in prices mentioned in situation (1) is usually the same as used in this work to represent price proportions higher than 1.0. By doing so, TNC companies increase the charged price in levels of 10% to incentive drivers to accept that trip. The same happens for situation (2), when TNC companies increase the base price to attract other drivers in order to serve trips from that region. In these four situations, carsharing clients may either decide to pay higher prices because they have no cheap competing option, or because they do not have other shared mobility option to choose.

The following analysis compares the performance of the chosen scenario for Scheduled Free-Floating, to a scenario with the same business model but considering 200% of Uber base price, and the other two business models. Figures 6.13 and 6.14 show that the actual share of demand, indicated by the points in both figures, does not differ much from its share among the served trips. Therefore, the profile of served trips is similar to the common profile of trips made by people who usually drive in the Metropolitan Region of São Paulo.

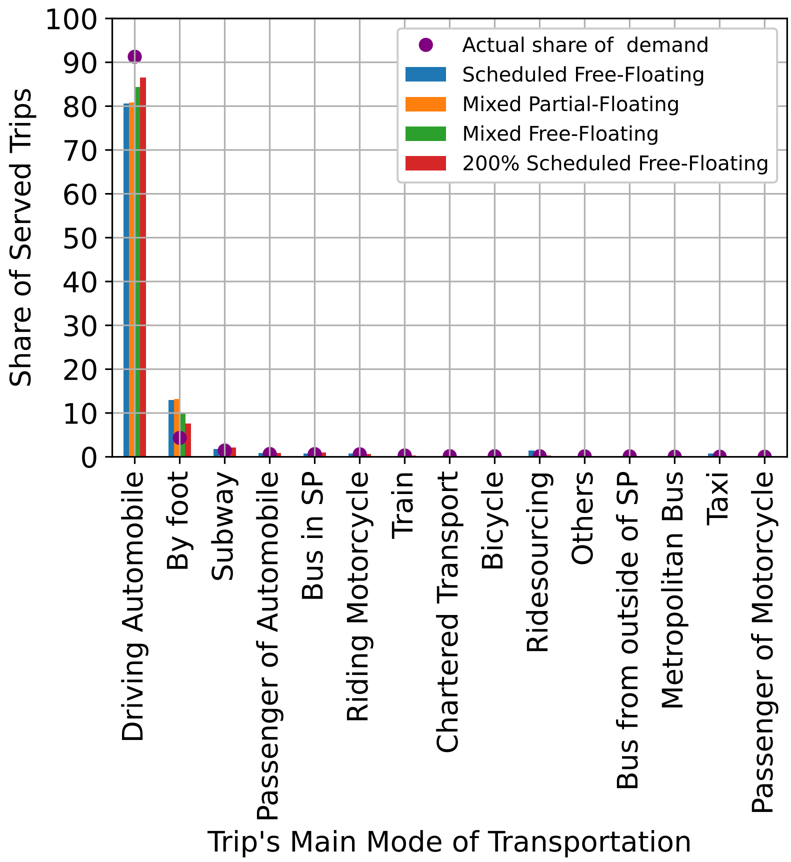


Figure 6.13: Share of served demand per mode

The actual demands of departure and destination are similar, since the green and red points in Figure 6.14 are close to perfectly overlapping. This happens due to the

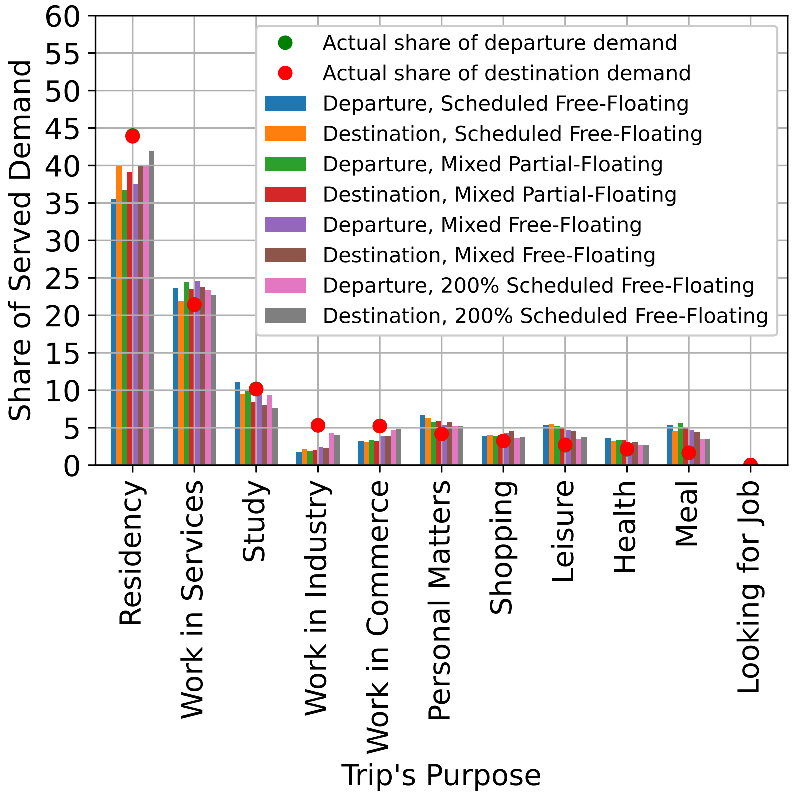


Figure 6.14: Share of served demand per purpose

"round-trip" nature of the day-to-day commuting. Regarding the trip profiles, although the scenario with 200% of Uber base price would serve almost 30 times more trips than the scenario with 70%, as shown by Figure 6.4, all business models are similar. However, the scenario with 200% tends to follow the actual demand of the trips more closely, while the other scenarios are more prone to serve short-term trips, such as those made by foot and for having a meal. As matter of comparison, while the average distance travelled by all possible trips is 12.5 kilometers, the scheduled served trips with 70% of Uber base price had an average of 7 kilometers.

Figure 6.15 shows this pattern by comparing the percentage of possible trips that were served. The trips' main mode of transportation that had the highest served percentage were trips made by taxi, ridesourcing services and on foot. Neither of these three modes are usual for long trips, due to the costs involved or walking effort. Besides, the mode "Driving Automobile" that represented more than 90% of all possible trips (actual demand shown in Figure 6.13) had a participation that is even lower than the trips made by foot in Figure 6.15. The scenario with 200% of Uber base price does not stand out for trips that were made by ridesourcing or taxi.

Figure 6.16 compares the departure and destination shares of trips' purposes. All evaluated scenarios had a pattern of more trips served for departure than destination about "Personal Matters", but the inverse when the purpose is "Shopping". Such situations are expected due to the spatial and temporal imbalance in demand along different parts of

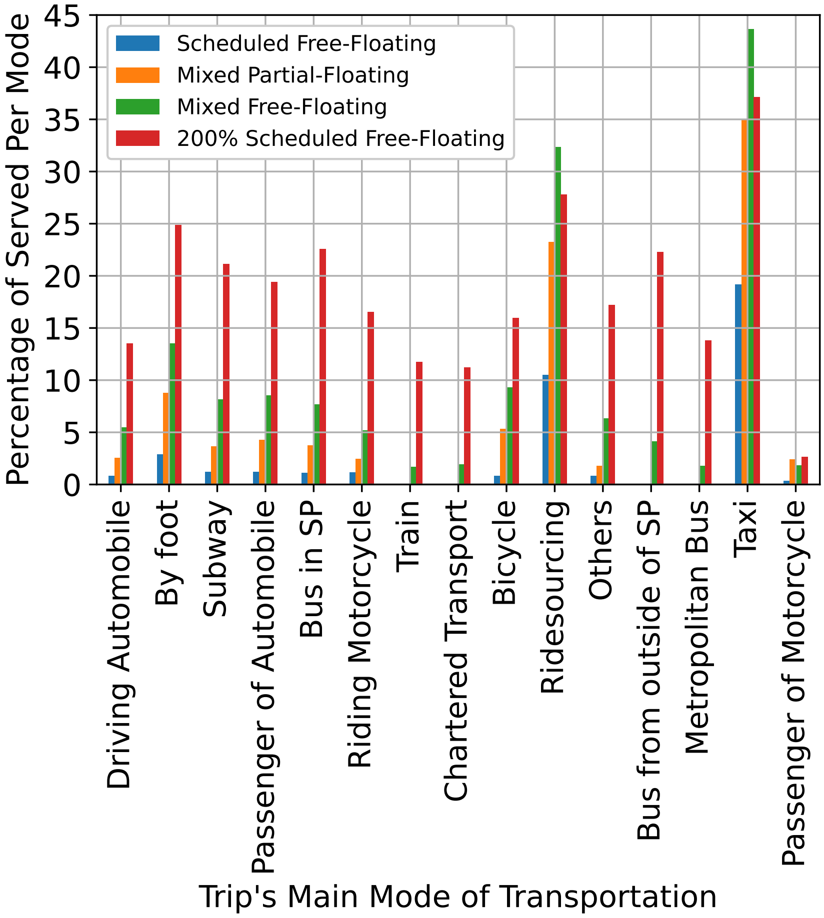


Figure 6.15: Percentage of served trips per mode

any city.

Although these results showed improvements in the carsharing performance by including on-demand trips into the business model, the more clients are served, the more resources are required to keep the service. The following subsection discusses the challenge of assuring enough parking slots to shelter the increasing number of vehicles.

6.2.2 The parking space issue

As mentioned in 3.4.4, this work also considers some parking slots for free-floating vehicles. Figures from 6.17 to 6.20 present the variation of the number of parked carsharing vehicles along the week, considering the Scheduled Free-Floating.

The points indicate any place where carsharing trips start and finish. The blue points are partially transparent to make it easier to notice which parts of the graph concentrate more points. Therefore, the bluer an area is, the more points exist in that

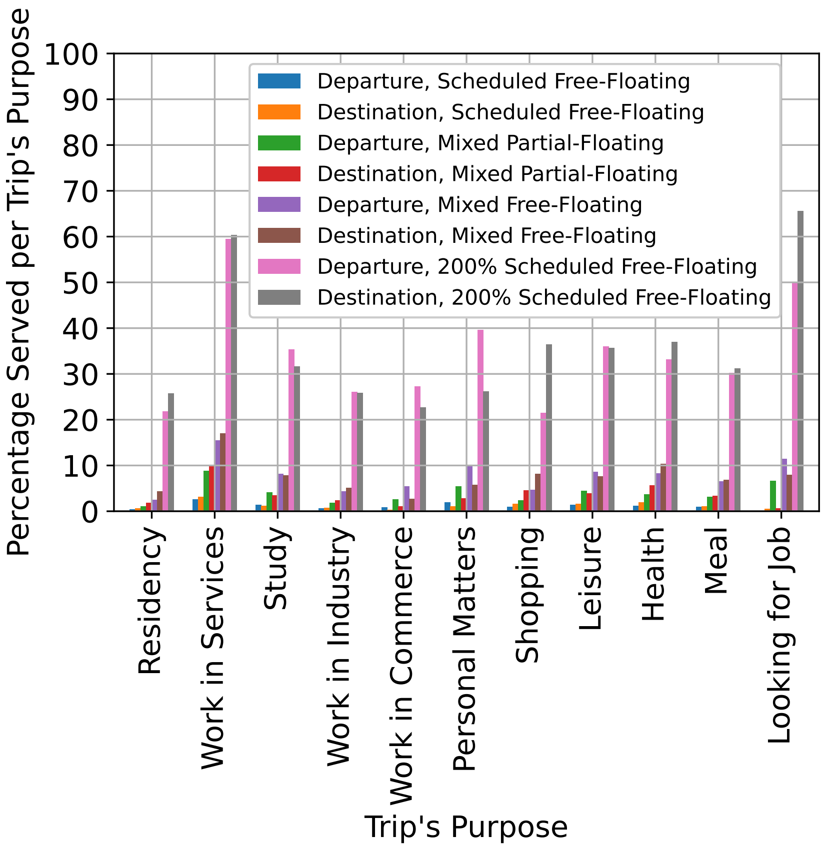


Figure 6.16: Percentage of served trips per purpose

area of the graph. Each point indicates the number of vehicles parking at some place minus the number of carsharing parking slots at that same place. Therefore, positive values indicate places with carsharing vehicles parked on the street because there are not enough parking slots. Negative values indicate parking slots that are rented but not used at that time. Points at zero mean that there are exactly as many vehicles as there are parking slots. Red points indicate the highest or lowest surplus of vehicles expected at that day.

The high positive values on Figure 6.17 represents a risk to the carsharing operators since it may be unrealistic to expect that there will be public spaces enough to park more than 100 free-floating vehicles in a same area. Although part of the profits may be used for renting additional parking slots, this solution will not scale well because that issue is generalized to multiple areas of the Metropolitan Region of São Paulo. As examples, Figure 6.18 shows the same business model as Figure 6.17 but reinvesting 5% of the profits for renting additional parking slots to critical areas. This would reduce the highest red point of Figure 6.17 from 153 to 115, a reduction of 25%. However, as shown by Figures 6.19 and 6.20, reinvesting 50% of the profits for renting additional parking slots would reduce the highest point to 80, a reduction of 48%; and reinvesting 100% of the profits would reduce the highest point to 66, a reduction of only 57%.

Nevertheless, negative surpluses are rare. Thus, the carsharing service will be

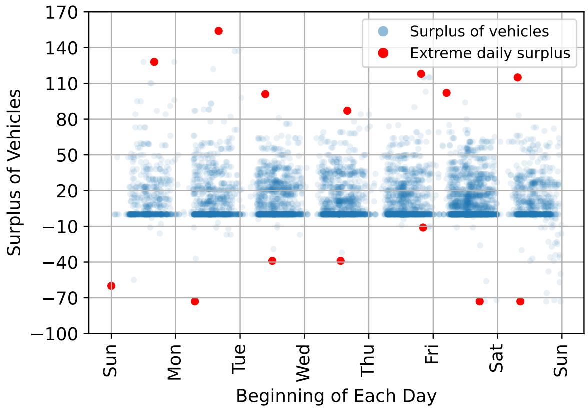


Figure 6.17: Balance of vehicles in the Scheduled Free-Floating

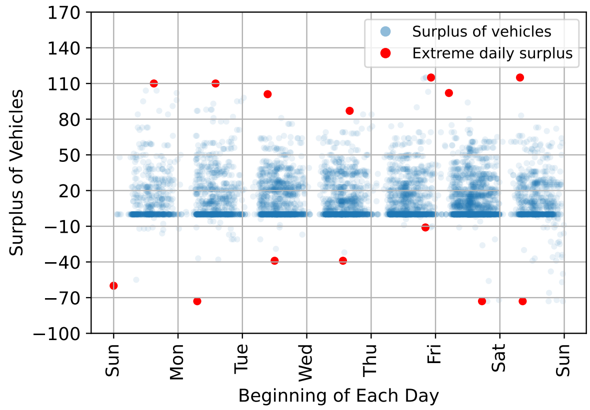


Figure 6.18: Balance of vehicles after 5% of reinvestment in parking slots

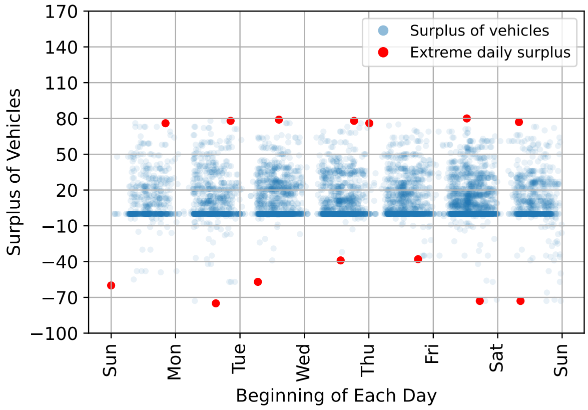


Figure 6.19: Balance of vehicles after 50% of reinvestment in parking slots

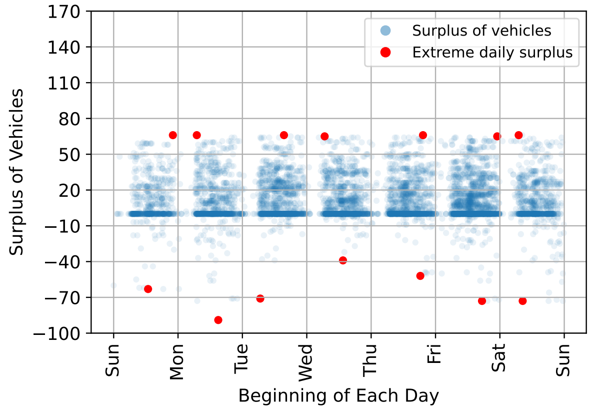


Figure 6.20: Balance of vehicles after 100% of reinvestment in parking slots

robust against unexpected events on the fleet, such as car crashes, that could disrupt the carsharing flow.

Chapter 7

Conclusion and Future Works

This section presents the final remarks for this PhD thesis. The following list summarizes the accomplishments of this work:

1. the carsharing fleet-sizing and shared mobility siting problems were contextualized;
2. analyses of the expected carsharing costs and profits;
3. discussion of benefits in clients being flexible to walk to nearby stations to get an available vehicle or station;
4. evaluation of the computational performance of carsharing fleet-sizing using different MILP formulations;
5. the NP-Completeness of optimizing the exact location of shared mobility facilities was proved;
6. the NP-Completeness of selecting an optimal subset of street segments to settle shared mobility facilities was proved;
7. three profitable carsharing business models solvable in polynomial-time by linear optimization were proposed;
8. comparison of performance and discussion of challenges faced by the low-cost carsharing service simulated for the whole Metropolitan Region of São Paulo.

The outcomes of this PhD thesis include eight works. The following list presents the already published ones:

1. Cristiano Martins Monteiro, Cláudia Aparecida Soares Machado, Mariana de Oliveira Lage, Fernando Tobal Berssaneti, Clodoveu Augusto Davis Jr., and José Alberto Quintanilha. Maximizing Carsharing Profits: An Optimization Model to Support the Carsharing Planning. In *25th International Conference on Production Research 2019 (ICPR 2019)*, volume 39, pages 1968–1976, Chicago, USA, aug 2019a. Elsevier

2. Mariana de Oliveira Lage, Claudia Aparecida Soares Machado, Cristiano Martins Monteiro, Fernando Tobal Berssaneti, and José Alberto Quintanilha. Location Suitable for the Implementation of Carsharing in the City of São Paulo. In *25th International Conference on Production Research 2019 (ICPR 2019)*, volume 39, pages 1962–1967, Chicago, USA, aug 2019. Elsevier

3. Cristiano Martins Monteiro, Geraldo Robson Mateus, and Clodoveu Augusto Davis Jr. Computational Performance of Carsharing Fleet-Sizing Optimization. In *XX Brazilian Symposium on GeoInformatics (GEOINFO)*, pages 111–122, 2019b

4. Cristiano Martins Monteiro, Cláudia Aparecida Soares Machado, Mariana de Oliveira Lage, Fernando Tobal Berssaneti, Clodoveu Augusto Davis Jr., and José Alberto Quintanilha. Optimization of carsharing fleet size to maximize the number of clients served. *Computers, Environment and Urban Systems*, 87:101623, 2021a

5. Cristiano Martins Monteiro, Cláudia Aparecida Soares Machado, Adelaide Cassia Nardocci, Fernando Tobal Berssaneti, José Alberto Quintanilha, and Clodoveu Augusto Davis Jr. Shared Mobility Opportunities and Their Computational Challenges for Improving Health-Related Quality of Life. In Roger Vickerman, editor, *International Encyclopedia of Transportation*, pages 376–383. Elsevier, Oxford, 2021b. ISBN 978-0-08-102672-4. doi: https://doi.org/10.1016/B978-0-08-102671-7.10754-7. URL https://www.sciencedirect.com/science/article/pii/B9780081026717107547

6. Mariana de Oliveira Lage, Cláudia Aparecida Soares Machado, Cristiano Martins Monteiro, Clodoveu Augusto Davis Jr., Charles Lincoln Kenji Yamamura, Fernando Tobal Berssaneti, and José Alberto Quintanilha. Using Hierarchical Facility Location, Single Facility Approach, and GIS in Carsharing Services. *Sustainability*, 13(22):12704, 2021

7. Cristiano Martins Monteiro and Clodoveu A Davis. Polynomial-Time Carsharing Optimization: Linear Formulation and Large-Scale Simulations. *IEEE Transactions on Intelligent Transportation Systems*, 24(4):4428–4437, 2023. doi: 10.1109/TITS.2022.3232149

Except for the second and sixth publications, the author of this PhD thesis was the first author of the mentioned publications. Another work comprising the results from Chapter 4 is expected to be published in an international journal.

The following section concludes this work and section 7.2 suggests future works.

7.1 Conclusion

The large-scale simulations made represent more than 2.5 millions inhabitants of São Paulo who are able to drive. Together, they make close to 9 million trips per day, indicating an average of less than four trips per person per day. This reflects how inefficient owning private vehicles can be if no shared mobility modes are used. Besides, private vehicles that keep parked most of the time may also hamper the performance of emerging shared mobility services that could use those parking spaces to serve much more clients and trips per day.

Better and cheaper shared mobility services are benefiting more people with each passing day. Indeed, the more clients carsharing companies have, the more resources those companies get to share and widen their services to even more people. However, providing good services to multiple clients for low prices requires significant efforts of planning and operation. Although it was shown in Chapter 6 that it is possible to offer a profitable low-cost carsharing service without fleet relocations, such service will be restricted to niches of demand. A broader carsharing will either depend on fleet relocations or charging fees higher than ridesourcing services to serve high percentages of demand.

Distance walked by the clients was shown to contribute to reduce the impact of lacking parking slots and the need of relocation operations. However, as the maximum distance that clients are flexible to walk increases, the percentage of clients who would need to walk to be served also increases. In real situations, it is likely that requiring clients to walk lengthy distances would decrease the demand. Also, unexpected events as car crashes or external events such as traffic and rainfall can slow down the sharing dynamics, hampering the flow of clients between stations.

However, results showed that the proposed business models are resilient against small unavailability of vehicles, because often there are more vehicles than parking slots in the stations. But the challenge in the number of required parking slots still remains. It is possible that few workers focused in relocating parked vehicles may diminish critical situations of having more vehicles than public parking spaces, and reduce the longest distance walked by clients. However, the optimization of this task is not guaranteed to be solved in polynomial-time.

The polynomial-time formulation proposed in this work was important for enabling the simulation of large-scale scenarios. In a real-world environment, carsharing companies using such efficient formulation are able to re-optimize their fleet often, since the fleet-sizing problem will not be another challenge to be solved. Also, different formulations have a significant impact in time required to solve a problem when it is not solved in polynomial-time. As shown in section 3.2, the formulation without Big-M not only reduced the computational time to solve the model but also had more stable run times than the formulation using the Big-M. Therefore, a carsharing company could properly plan that its operation tasks will start after the solver finish running, since the optimization problem will not take too long to be solved.

However, other problems may not justify intensive efforts of solving them to the global optimum. In the case of SMSS and SASS defined in Chapter 4, because they are NP-Hard and their instance is large, the gap of 14% between the best solution found and the upper bound was considered good enough to interrupt the optimization. Indeed, the solution found without optimality guarantee was suitable for the analysis and performance comparison of carsharing business models. The following section presents the suggested future works.

7.2 Future Work

The suggested future work includes:

1. proposing a real-time model, making trips completely on-demand and with dynamic pricing (rare events of demand, weather);
2. including price discounts into the optimization to reward clients who agree to walk. Also, analyzing the expected profits after applying such discounts;
3. comparing investment requirements and expected profits on buying vehicles instead of renting them. This includes considering the vehicle depreciation along multiple years and additional maintenance costs;
4. evaluating if it is possible for on-demand shared mobility services to replace mass public transportation;
5. comparing with autonomous vehicles;
6. evaluating formulations based on the duration of recharging electric vehicles;
7. assessing the impact of time spent on recharging batteries for partial or free-floating carsharing services. In those cases, charging batteries can become an issue since clients can finish their rentals in places without a charging spot, not recharging the vehicle for the next client;
8. evaluating longer trips and the battery range of electric vehicles;

9. solving SSMS and SASS problems considering locations with parking slots currently available for renting;

10. evaluating different functions to generate utility values for both SSMS and SASS problems;

11. applying additional preprocessing tasks to reduce the instance size for both SSMS and SASS problems;

12. optimizing the location of stations in the same optimization phase of fleet-sizing and profit maximization;

13. including non-deterministic situations on the objective function and constraint set, such as inclusion of on-demand trips, traffic, weather conditions, and adversities along the trip, such as flat tire, broken windshield, accident or mechanical problem;

14. considering different client profiles, such as allowing different price limits, being flexible to use public transport in part of the trip, being flexible to split the ride with strangers or other people that are almost strange (e.g., another student from the same university);

15. adjusting parameters according to the data publicly available by competing companies, such as the data shared on the Uber Movement website[1];

16. predicting operational issues that might emerge when offering a cheap carsharing service that, ocasionally, will not have all of its assumptions satisfied;

17. assessing how a small number of staff dedicated to relocating the fleet can reduce critical scenarios of lacking available parking slots, vehicles, or even electric chargers, that could force clients to walk lengthy distances to be served;

18. evaluating the use of meta-heuristics to solve the relocation problem to avoid critical scenarios of parking slots in real-time;

19. proposing novel algorithms and formulations for solving both SSMS and SASS problems;

20. comparing the computational performance of different parameter settings of Gurobi, while solving instances of SSMS and SASS problems with different sizes and topologies;

21. evaluating parallelism and distributed processing techniques during the optimization.

[1] https://movement.uber.com/

Bibliography

Álvaro Aguilera-García, Juan Gomez, and Natalia Sobrino. Exploring the adoption of moped scooter-sharing systems in Spanish urban areas. *Cities*, 96:102424, 2020.

Szymon Janusz Albinski. A branch-and-cut method for the Vehicle Relocation Problem in the One-Way Car-Sharing. Master's thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 5 2015.

Andy S Alic, Jussara M Almeida, Wagner Meira Jr, Dorgival O Guedes, Walter dos Santos, Ignacio Blanquer, Sandro Fiore, Nádia P Kozievitch, Nazareno Andrade, Tarciso Braz, et al. GIS and Data: Three applications to enhance Mobility. In *XIX Brazilian Symposium on GeoInformatics (GEOINFO)*, pages 1–12, 2018.

María Ampudia-Renuncio, Begona Guirao, Rafael Molina-Sánchez, and Cristina Engel de Alvarez. Understanding the spatial distribution of free-floating carsharing in cities: Analysis of the new Madrid experience through a web-based platform. *Cities*, 98:102593, 2020.

Mikołaj Bartłomiejczyk. Driving Performance Indicators of Electric Bus Driving Technique: Naturalistic Driving Data Multicriterial Analysis. *IEEE Transactions on Intelligent Transportation Systems*, 20(99):1–10, 2018.

Mokhtar S Bazaraa, John J Jarvis, and Hanif D Sherali. *Linear Programming and Network Flows*. John Wiley & Sons, 4 edition, 2010.

Gabriel Bortoleto Benetti. CARROS COMPARTILHADOS: VIABILIDADE NA CIDADE DE SÃO PAULO, 2019. Monography (Bachelor of Civil Engineering), Federal Institute of Education, Science and Technology of São Paulo (IFSP), Brazil.

Kateřina Böhmová, Yann Disser, Matúš Mihalák, and Rastislav Šrámek. Scheduling transfers of resources over time: Towards car-sharing with flexible drop-offs. In *LATIN 2016: Theoretical Informatics*, pages 220–234. Springer, 2016.

Karla AV Borges, Clodoveu Augusto Davis Jr., and Alberto HF Laender. OMT-G: An Object-Oriented Data Model for Geographic Applications. *GeoInformatica*, 5(3):221–260, 2001.

Burak Boyacı and Konstantinos G Zografos. Investigating the effect of temporal and spatial flexibility on the performance of one-way electric carsharing systems. *Transportation Research Part B: Methodological*, 129:244–272, 2019.

Georg Brandstätter, Markus Leitner, and Ivana Ljubić. Location of charging stations in electric car sharing systems. *Transportation Science*, 54(5):1408–1438, 2020.

Maurizio Bruglieri, Ferdinando Pezzella, and Ornella Pisacane. A two-phase optimization method for a multiobjective vehicle relocation problem in electric carsharing systems. *Journal of Combinatorial Optimization*, 36(1):162–193, 2018.

Kathie Cameron. Induced matchings. *Discrete Applied Mathematics*, 24(1-3):97–102, 1989.

Jeffrey D Camm, Amitabh S Raturi, and Shigeru Tsubakitani. Cutting big m down to size. *Interfaces*, 20(5):61–66, 1990.

Michele Cocca, Danilo Giordano, Marco Mellia, and Luca Vassio. Free Floating Electric Car Sharing: A Data Driven Approach for System Design. *IEEE Transactions on Intelligent Transportation Systems*, 20(12):4691–4703, 2019.

Boyd Cohen and Jan Kietzmann. Ride on! Mobility business models for the sharing economy. *Organization & Environment*, 27(3):279–296, 2014.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 3 edition, 2009.

Gonçalo Homem de Almeida Correia and António Pais Antunes. Optimization approach to depot location and trip selection in one-way carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):233–247, 2012.

Gonçalo Homem De Almeida Correia, Diana Ramos Jorge, and David Marques Antunes. The added value of accounting for users' flexibility and information on the potential of a station-based one-way car-sharing system: An application in Lisbon, Portugal. *Journal of Intelligent Transportation Systems*, 18(3):299–308, 2014.

Alexandru Cristian, Luke Marshall, Mihai Negrea, Flavius Stoichescu, Peiwei Cao, and Ishai Menache. Multi-Itinerary Optimization as Cloud Service (Industrial Paper). In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 279–288, 2019.

Lôys Leine Alegre da Silva and Paulo André de Oliveira. MÉTODOS DE DEPRECIAÇÃO DE VEÍCULOS NACIONAIS ATÉ MIL CILINDRADAS: UM ESTUDO DE CASO. *Tekhne e Logos*, 9(2):9–19, 2018.

Antoine Deza, Kai Huang, and Michael R Metel. Charging station optimization for balanced electric car sharing. *Discrete Applied Mathematics*, 2020.

Bowen Du, Yongxin Tong, Zimu Zhou, Qian Tao, and Wenjun Zhou. Demand-aware charger planning for electric vehicle sharing. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1330–1338, 2018.

Herbert Fleischner, Gert Sabidussi, and Vladimir I Sarvanov. Maximum independent sets in 3-and 4-regular Hamiltonian graphs. *Discrete Mathematics*, 310(20):2742–2749, 2010.

Michael R Garey and David S Johnson. *Computers and intractability: A guide to the theory of NP-completeness.* W. H. Freeman, 1 edition, 1979.

Konstantinos N Genikomsakis, Ignacio Angulo Gutierrez, Dimitrios Thomas, and Christos S Ioakimidis. Simulation and Design of Fast Charging Infrastructure for a University-based e-Carsharing System. *IEEE Transactions on Intelligent Transportation Systems*, 19(99):1–10, 2017.

Rüdiger Hahn, Felix Ostertag, Adrian Lehr, Marion Büttgen, and Sabine Benoit. "I like it, but I don't use it": Impact of carsharing business models on usage intentions in the sharing economy. *Business Strategy and the Environment*, 29(3):1404–1418, 2020.

Yusuke Hara and Eiji Hato. A car sharing auction with temporal-spatial OD connection conditions. *Transportation research procedia*, 23:22–40, 2017.

Raj Jain. *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling.* Wiley professional computing. Wiley, 1991. ISBN 978-0-471-50336-1.

Diana Jorge, Goncalo HA Correia, and Cynthia Barnhart. Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1667–1675, 2014.

Diana Jorge, Cynthia Barnhart, and Gonçalo Homem de Almeida Correia. Assessing the viability of enabling a round-trip carsharing system to accept one-way trips: Application to Logan Airport in Boston. *Transportation Research Part C: Emerging Technologies*, 56:359–372, 2015.

Ed Klotz and Alexandra M Newman. Practical guidelines for solving difficult mixed integer linear programs. *Surveys in Operations Research and Management Science*, 18 (1-2):18–32, 2013.

CW Ko and F Bruce Shepherd. Bipartite domination and simultaneous matroid covers. *SIAM Journal on Discrete Mathematics*, 16(4):517–523, 2003.

Martin Krzywinski and Naomi Altman. Points of Significance: Visualizing samples with box plots. *Nature Methods*, 11(2):119–120, 2014.

Mariana de Oliveira Lage, Claudia Aparecida Soares Machado, Cristiano Martins Monteiro, Fernando Tobal Berssaneti, and José Alberto Quintanilha. Location Suitable for the Implementation of Carsharing in the City of São Paulo. In *25th International Conference on Production Research 2019 (ICPR 2019)*, volume 39, pages 1962–1967, Chicago, USA, aug 2019. Elsevier.

Mariana de Oliveira Lage, Cláudia Aparecida Soares Machado, Cristiano Martins Monteiro, Clodoveu Augusto Davis Jr., Charles Lincoln Kenji Yamamura, Fernando Tobal Berssaneti, and José Alberto Quintanilha. Using Hierarchical Facility Location, Single Facility Approach, and GIS in Carsharing Services. *Sustainability*, 13(22):12704, 2021.

Minghui Lai, Qian Hu, Ying Liu, and Zekun Lang. A rolling-horizon decision framework for integrating relocation and user flexibility in one-way electric carsharing systems. *Transportation Research Part C: Emerging Technologies*, 144:103867, 2022.

Albert YS Lam, Yiu-Wing Leung, and Xiaowen Chu. Electric vehicle charging station placement: Formulation, complexity, and solutions. *IEEE Transactions on Smart Grid*, 5(6):2846–2856, 2014.

Quentin Lamour, Adriano M Morelli, and Karin R de C Marins. Improving walkability in a TOD context: spatial strategies that enhance walking in the Belém neighbourhood, in São Paulo, Brazil. *Case studies on transport policy*, 7(2):280–292, 2019.

Zhiyong Liu, Ruimin Li, and Jingchen Dai. Effects and feasibility of shared mobility with shared autonomous vehicles: An investigation based on data-driven modeling approach. *Transportation Research Part A: Policy and Practice*, 156:206–226, 2022.

Luís Eduardo Oliveira Lizardo and Clodoveu Augusto Davis Jr. OMT-G Designer: a Web tool for modeling geographic databases in OMT-G. In *International Conference on Conceptual Modeling*, pages 228–233. Springer, 2014.

Luís Eduardo Oliveira Lizardo and Clodoveu Augusto Davis Jr. A PostGIS extension to support advanced spatial data types and integrity constraints. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–10, 2017.

Pedro Logiodice, Renato Arbex, Diego Tomasiello, and Mariana A Giannotti. Spatial visualization of job inaccessibility to identify transport related social exclusion. In *XVI Brazilian Symposium on GeoInformatics (GEOINFO)*, pages 105–118, 2015.

Mengshi Lu, Zhihao Chen, and Siqian Shen. Optimizing the profitability and quality of service in carshare systems under demand uncertainty. *Manufacturing & Service Operations Management*, 20(2):162–180, 2018.

Xin Luan, Lin Cheng, Yang Zhou, and Fang Tang. Strategies of Car-Sharing Promotion in Real Market. In *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, pages 159–163. IEEE, 2018.

Cláudia Aparecida Soares Machado, Nicolas Patrick Marie Salles Hue, Fernando Tobal Berssaneti, and José Alberto Quintanilha. An Overview of Shared Mobility. *Sustainability*, 10(12):4342, 2018.

Roger L Mackett and Roselle Thoreau. Transport, social exclusion and health. *Journal of Transport & Health*, 2(4):610–617, 2015.

Elliot W Martin and Susan A Shaheen. Greenhouse Gas Emission Impacts of Carsharing in North America. *IEEE Transactions on Intelligent Transportation Systems*, 12(4): 1074–1086, 2011.

George Mavrotas. Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation*, 213(2): 455–465, 2009.

Konstantina Mellou, Luke Marshall, Krishna Chintalapudi, Patrick Jaillet, and Ishai Menache. Optimizing Onsite Food Services at Scale. In *Proceedings of the 28th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 618–629, 2020.

Cristiano Martins Monteiro and Clodoveu A Davis. Polynomial-Time Carsharing Optimization: Linear Formulation and Large-Scale Simulations. *IEEE Transactions on Intelligent Transportation Systems*, 24(4):4428–4437, 2023. doi: 10.1109/TITS.2022.3232149.

Cristiano Martins Monteiro, Flávio Vinícius Cruzeiro Martins, and Clodoveu Augusto Davis Jr. Optimization of Bus Stops, New Pick-up and Drop-off Locations for Public Transportation. *Journal of Information and Data Management*, 9(3):229–229, 2018.

Cristiano Martins Monteiro, Cláudia Aparecida Soares Machado, Mariana de Oliveira Lage, Fernando Tobal Berssaneti, Clodoveu Augusto Davis Jr., and José Alberto Quintanilha. Maximizing Carsharing Profits: An Optimization Model to Support the Carsharing Planning. In *25th International Conference on Production Research 2019 (ICPR 2019)*, volume 39, pages 1968–1976, Chicago, USA, aug 2019a. Elsevier.

Cristiano Martins Monteiro, Geraldo Robson Mateus, and Clodoveu Augusto Davis Jr. Computational Performance of Carsharing Fleet-Sizing Optimization. In *XX Brazilian Symposium on GeoInformatics (GEOINFO)*, pages 111–122, 2019b.

Cristiano Martins Monteiro, Cláudia Aparecida Soares Machado, Mariana de Oliveira Lage, Fernando Tobal Berssaneti, Clodoveu Augusto Davis Jr., and José Alberto Quintanilha. Optimization of carsharing fleet size to maximize the number of clients served. *Computers, Environment and Urban Systems*, 87:101623, 2021a.

Cristiano Martins Monteiro, Cláudia Aparecida Soares Machado, Adelaide Cassia Nardocci, Fernando Tobal Berssaneti, José Alberto Quintanilha, and Clodoveu Augusto Davis Jr. Shared Mobility Opportunities and Their Computational Challenges for Improving Health-Related Quality of Life. In Roger Vickerman, editor, *International Encyclopedia of Transportation*, pages 376–383. Elsevier, Oxford, 2021b. ISBN 978-0-08-102672-4. doi: https://doi.org/10.1016/B978-0-08-102671-7.10754-7. URL https://www.sciencedirect.com/science/article/pii/B9780081026717107547.

Abood Mourad, Jakob Puchinger, and Chengbin Chu. A survey of models and algorithms for optimizing shared mobility. *Transportation Research Part B: Methodological*, 123: 323–346, 2019.

Alan T Murray, Richard L Church, and Xin Feng. Single facility siting involving allocation decisions. *European Journal of Operational Research*, 284(3):834–846, 2020.

Mehdi Nourinejad, Sirui Zhu, Sina Bahrami, and Matthew J. Roorda. Vehicle relocation and staff rebalancing in one-way carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, 81:98–113, 2015.

Colin Pooley. Mobility, transport and social inclusion: Lessons from history. *Social Inclusion*, 4(3):100–109, 2016.

Martin Repoux, Burak Boyacı, and Nikolas Geroliminis. Simulation and optimization of one-way car-sharing systems with variant relocation policies. In *hEART 2014-3rd Symposium of the European Association for Research in Transportation*, 2014.

Ulrike Ritzinger, Jakob Puchinger, and Richard F Hartl. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54 (1):215–231, 2016.

Joyashree Roy, Debalina Chakravarty, Shyamasree Dasgupta, Debrupa Chakraborty, Shamik Pal, and Duke Ghosh. Where is the hope? Blending modern urban lifestyle with cultural practices in India. *Current Opinion in Environmental Sustainability*, 31: 96–103, 2018.

Gonçalo Gonçalves Duarte Santos and Gonçalo Homem de Almeida Correia. Finding the relevance of staff-based vehicle relocations in one-way carsharing systems through the use of a simulation-based optimization tool. *Journal of Intelligent Transportation Systems*, 23(6):583–604, 2019.

Maximilian Schiffer, Gerhard Hiermann, Fabian Rüdel, and Grit Walther. A polynomial-time algorithm for user-based relocation in free-floating car sharing systems. *Transportation Research Part B: Methodological*, 143:65–85, 2021.

Joseph P Schwieterman and Alice Bieszczat. The cost to carshare: A review of the changing prices and taxation levels for carsharing in the United States 2011–2016. *Transport Policy*, 57:1–9, 2017.

Social Exclusion Unit S.E.U. Making the connections: final report on transport and social exclusion. Technical report, Office of the Deputy Prime Minister, London, England, 2003. URL https://www.ilo.org/wcmsp5/groups/public/---ed_emp/---emp_policy/---invest/documents/publication/wcms_asist_8210.pdf.

Átila M Silva Jr, Humberto T Marques-Neto, and Jussara MA Gonçalves. Custos e benefícios de viagens compartilhadas em um sistema de transporte multimodal. In *Anais do II Workshop de Computação Urbana*. SBC, 2018.

Mitja Stiglic, Niels Agatz, Martin Savelsbergh, and Mirko Gradisar. Enhancing urban mobility: Integrating ride-sharing and public transit. *Computers & Operations Research*, 90:12–21, 2018.

Philipp Ströhle, Christoph M Flath, and Johannes Gärttner. Leveraging customer flexibility for car-sharing fleet optimization. *Transportation Science*, 53(1):42–61, 2019.

A Tamir. On totally unimodular matrices. *Networks*, 6(4):373–382, 1976.

Christopher Tchervenkov, Milos Balać, Sebastian Hörl, Henrik Becker, and Kay W Axhausen. How much parking space can carsharing save? *Arbeitsberichte Verkehrs-und Raumplanung*, 1374, 2018.

Diego Bogado Tomasiello, Mariana Giannotti, and Flávia F Feitosa. ACCESS: An agent-based model to explore job accessibility inequalities. *Computers, Environment and Urban Systems*, 81:101462, 2020.

Katarzyna Turoń, Andrzej Kubik, and Feng Chen. What Car for Car-Sharing? Conventional, Electric, Hybrid or Hydrogen Fleet? Analysis of the Vehicle Selection Criteria for Car-Sharing Systems. *Energies*, 15(12), 2022. ISSN 1996-1073. doi: 10.3390/en15124344. URL https://www.mdpi.com/1996-1073/15/12/4344.

Peter van der Waerden, Harry Timmermans, and Marloes de Bruin-Verhoeven. Car drivers' characteristics and the maximum walking distance between parking facility and final destination. *Journal of transport and land use*, 10(1):1–11, 2017.

M Veloso-Poblete, P Palominos, L Quezada, and M Alfaro. Modelo para la planeación y asignación de viajes interurbanos a buses, considerando mantenciones preventivas. In *9th International Conference on Production Research–Americas*, volume 2018, pages 1–6, 2018.

J Viegas and L Martinez. Transition to shared mobility: How large cities can deliver inclusive transport services. Technical report, Corporate Partnership Board Report, International Transport Forum, Paris, France, Tech. Rep, https://www.itf-oecd.org/sites/default/files/docs/transition-shared-mobility.pdf, 2017.

Fábio Luciano Violin. Motivators for the use of vehicle sharing services on demand in Brazil. *Journal of Globalization, Competitiveness and Governability*, 15(3), 2021.

Laurence A Wolsey. *Integer Programming*. John Wiley & Sons, 1 edition, 1998.

Charles Lincoln Kenji Yamamura, Harmi Takiya, Cláudia Aparecida Soares Machado, José Carlos Curvelo Santana, José Alberto Quintanilha, and Fernando Tobal Berssaneti. Electric Cars in Brazil: An Analysis of Core Green Technologies and the Transition Process. *Sustainability*, 14(10):6064, 2022.

Sicheng Zhang, Xiyuan Zhao, Xiang Li, and Haitao Yu. Heterogeneous fleet management for one-way electric carsharing system with optional orders, vehicle relocation and on-demand recharging. *Computers & Operations Research*, page 105868, 2022.

Appendix A

Dataset's Metadata of the São Paulo's OD Survey

Table A.1 describes the columns of the dataset sourced from the Origin-Destination Survey of São Paulo[1] in the years of 2017 and 2018. This data was drawn from the file "LAYOUT OD2017.xlsx", which is written in portuguese.

Since the relation "OD_2017" presented in the Figure 5.1 represents the data Origin-Destination dataset, every line in the Table A.1 indicates one attribute in the relation "OD_2017".

The specifications of the coordinates are:

- Coordinates system: UTM - SAD69
- Horizontal datum: Córrego Alegre (23) [EPSG:22523]
- Digitizing: Planimetric
- Allowable error: up to 10 (ten) meters

[1] http://www.metro.sp.gov.br/pesquisa-od/

Table A.1: LAYOUT PESQUISA ORIGEM DESTINO 2017

	Variável	Conteúdo	Início	Fim	Compr	Códigos
1	ZONA	Zona do Domicílio	1	3	3	1 a 517
2	MUNI_DOM	Município de Domicílio	4	5	2	1 a 39
3	CO_DOM_X	Coordenada X Domicílio	6	17	12	12 dígitos
4	CO_DOM_Y	Coordenada Y Domicílio	18	29	12	12 dígitos
5	ID_DOM	Identifica Domicílio	30	37	8	
6	F_DOM	Identifica Primeiro Registro do Domicílio	38	38	1	0 - Demais Registros
						1 - Primeiro Registro do Domicílio
7	FE_DOM	Fator de Expansão do Domicílio	39	49	11	11 dígitos 6 casas decimais
8	DOM	Número do Domicílio	50	53	4	
9	CD_ENTRE	Código de Entrevista	54	54	1	1 - Completa com Viagem
						2 - Completa sem Viagem
10	DATA	Data da Entrevista	55	62	8	
11	TIPO_DOM	Tipo de Domicílio	63	63	1	1 - Particular
						2 - Coletivo
12	AGUA	Possui água encanada?	64	64	1	0 - Não
						1 - Sim
13	RUA_PAVI	A rua é pavimentada?	65	65	1	0 - Não
						1 - Sim
14	NO_MORAD	Total de Moradores no Domicílio	66	67	2	
15	TOT_FAM	Total de Familias no Domicílio	68	68	1	
16	ID_FAM	Identifica Família	69	77	9	

	Variável	Conteúdo	Início	Fim	Compr	Códigos
17	F_FAM	Identifica Primeiro Registro da Família	78	78	1	0 - Demais Registros
						1 - Primeiro Registro da Família
18	FE_FAM	Fator de Expansão da Família	79	89	11	11 dígitos 6 casas decimais
19	FAMILIA	Número da Família	90	90	1	
20	NO_MORAF	Total de Moradores na Família	91	92	2	
21	CONDMORA	Condição de Moradia	93	93	1	1 - Alugada
						2 - Própria
						3 - Cedida
						4 - Outros
						5 - Não Respondeu
22	QT_BANHO	Banheiros	94	94	1	
23	QT_EMPRE	Empregados Domésticos	95	95	1	
24	QT_AUTO	Automóveis	96	96	1	
25	QT_MICRO	Microcomputadores	97	97	1	
26	QT_LAVALOU	Máquinas de Lava louça	98	98	1	
27	QT_GEL1	Geladeiras de 1 porta	99	99	1	
28	QT_GEL2	Geladeiras de 2 portas	100	100	1	
29	QT_FREEZ	Freezer	101	101	1	
30	QT_MLAVA	Máquinas de Lavar	102	102	1	
31	QT_DVD	DVDs	103	103	1	
32	QT_MICROON	Microondas	104	104	1	
33	QT_MOTO	Motos	105	105	1	

	Variável	Conteúdo	Início	Fim	Compr	Códigos
34	QT_SECAROU	Secadora de roupas	106	106	1	
35	QT_BICICLE	Bicicletas	107	107	1	
36	NAO_DCL_IT	Código de Declaração de Itens de Conforto	108	108	1	0 - Não Declarou Itens de Conforto
						1 - Declarou Itens de Conforto
37	CRITERIOBR	Critério de Classificação Econômica Brasil	109	109	1	1 - A
						2 - B1
						3 - B2
						4 - C1
						5 - C2
						6 - D - E
38	PONTO_BR	Pontos Critério Brasil	110	111	2	
39	ANO_AUTO1	Ano Fabricação - Auto 1	112	115	4	
40	ANO_AUTO2	Ano Fabricação - Auto 2	116	119	4	
41	ANO_AUTO3	Ano Fabricação - Auto 3	120	123	4	
42	RENDA_FA	Renda Familiar Mensal	124	132	9	9 dígitos 2 casa decimais
43	CD_RENFA	Código de Renda Familiar	133	133	1	1 - Renda Familiar Declarada e Maior que Zero
						2 - Renda Familiar Declarada como Zero
						3 - Renda Atribuída pelo Critério Brasil

	Variável	Conteúdo	Início	Fim	Compr	Códigos
						4 - Renda Atribuída pela Média da Zona
44	ID_PESS	Identifica Pessoa	134	144	11	
45	F_PESS	Identifica Primeiro Registro da Pessoa	145	145	1	0 - Demais registros
						1 - Primeiro registro da pessoa
46	FE_PESS	Fator de Expansão da Pessoa	146	156	11	11 dígitos 6 casas decimais
47	PESSOA	Número da Pessoa	157	158	2	
48	SIT_FAM	Situação Familiar	159	159	1	1 - Pessoa Responsável
						2 - Cônjuge/Companheiro(a)
						3 - Filho(a)/Enteado(a)
						4 - Outro Parente
						5 - Agregado
						6 - Empregado Residente
						7 - Parente do Empregado Residente
49	IDADE	Idade	160	161	2	(anos)
50	SEXO	Gênero	162	162	1	1 - Masculino
						2 - Feminino
51	ESTUDA	Estuda Atualmente?	163	163	1	1 - Não
						2 - Creche/Pré-Escola
						3 - 1º Grau /Fundamental
						4 - 2º Grau/Médio

	Variável	Conteúdo	Início	Fim	Compr	Códigos
						5 – Superior/Universitário
						6 - Outros
52	GRAU_INS	Grau de Instrução	164	164	1	1 - Não-Alfabetizado/ Fundamental I Incompleto
						2 - Fundamental I Completo/ Fundamental II Incompleto
						3 - Fundamental II Completo/ Médio Incompleto
						4 - Médio Completo/ Superior Incompleto
						5 - Superior Completo
53	CD_ATIVI	Condição de Atividade	165	165	1	1 - Tem trabalho regular
						2 - Faz bico
						3 - Em Licença Médica
						4 - Aposentado/Pensionista
						5 - Sem Trabalho
						6 - Nunca Trabalhou
						7 - Dona de Casa
						8 - Estudante
54	CO_REN_I	Condição de Renda Individual	166	166	1	1 - Tem Renda
						2 - Não Tem Renda
						3 - Não Respondeu
55	VL_REN_I	Renda Individual	167	175	9	9 dígitos 2 casa decimais

	Variável	Conteúdo	Início	Fim	Compr	Códigos
56	ZONA_ESC	Zona da Escola	176	178	3	1 a 517 ou 999 quando for zona externa
57	MUNIESC	Município da Escola	179	180	2	1 a 39 ou 99 quando for município externo
58	CO_ESC_X	Coordenada X Escola	181	192	12	12 dígitos
59	CO_ESC_Y	Coordenada Y Escola	193	204	12	12 dígitos
60	TIPO_ESC	Tipo de Escola	205	205	1	1 - Pública
						2 - Particular
61	ZONATRA1	Zona do Primeiro Trabalho	206	208	3	1 a 517 ou 999 quando for zona externa
62	MUNITRA1	Município do Primeiro Trabalho	209	210	2	1 a 39 ou 99 quando for município externo
63	CO_TR1_X	Coordenada X 1º Trabalho	211	222	12	12 dígitos
64	CO_TR1_Y	Coordenada Y 1º Trabalho	223	234	12	12 dígitos
65	TRAB1_RE	Primeiro Trabalho é igual a Residência ?	235	235	1	1 - Sim
						2 - Não
						3 – Sem endereço fixo
66	TRABEXT1	Realiza Trabalho Externo-1º Trabalho	236	236	1	1 - Sim
						2 - Não

	Variável	Conteúdo	Início	Fim	Compr	Códigos
67	OCUP1	Ocupação do 1º Trabalho	237	238	2	1 - Membros superiores do poder público, dirigentes de organizações de interesse público e de empresas e gerentes
						2 - Profissionais das ciências e das artes
						3 - Técnicos de nível médio
						4 - Trabalhadores de serviços administrativos
						5 - Trabalhadores dos serviços
						6 - Vendedores do comércio em lojas e mercados
						7 - Trabalhadores agropecuários, florestais e da pesca
						8 - Trabalhadores da produção de bens e serviços industriais
						9 - Trabalhadores em serviços de reparação e manutenção
						10 - Membros das Forças Armadas, Policiais e Bombeiros Militares
						11 - Outras ocupações

	Variável	Conteúdo	Início	Fim	Compr	Códigos
68	SETOR1	Setor de Atividade do 1º Trabalho	239	240	2	1 - Agrícola
						2 - Construção Civil
						3 - Indústria
						4 - Comércio
						5 - Serviço de Transporte de Carga
						6 - Serviço de Transporte de Passageiros
						7 - Serviço Creditício-financeiro
						8 - Serviço Pessoal
						9 - Serviço de Alimentação
						10 - Serviço de Saúde
						11 - Serviço de Educação
						12 - Serviço Especializado
						13 - Serviço de Administração Pública
						14 - Outros Serviços
69	VINC1	Vínculo Empregatício do 1º Trabalho	241	241	1	1 - Assalariado com carteira
						2 - Assalariado sem carteira
						3 - Funcionário público
						4 - Autônomo
						5 - Empregador

	Variável	Conteúdo	Início	Fim	Compr	Códigos
						6 - Profissional Liberal
						7 - Dono de negócio familiar
						8 - Trabalhor familiar
70	ZONATRA2	Zona do Segundo Trabalho	242	244	3	1 a 517 ou 999 quando for zona externa
71	MUNITRA2	Município do Segundo Trabalho	245	246	2	1 a 39 ou 99 quando for município externo
72	CO_TR2_X	Coordenada X 2º Trabalho	247	258	12	12 dígitos
73	CO_TR2_Y	Coordenada Y 2º Trabalho	259	270	12	12 dígitos
74	TRAB2_RE	Segundo Trabalho é igual a Residência ?	271	271	1	1 - Sim
						2 - Não
						3 – Sem endereço fixo
75	TRABEXT2	Realiza Trabalho Externo-2º Trabalho	272	272	1	1 - Sim
						2 - Não
76	OCUP2	Ocupação do 2º Trabalho	273	274	2	idem à 1º Ocupação
77	SETOR2	Setor de Atividade do 2º Trabalho	275	276	2	idem ao 1º Setor de Atividade
78	VINC2	Vínculo Empregatício do 2º Trabalho	277	277	1	idem ao 1º Vínculo Empregatício
79	N_VIAG	Número da Viagem	278	279	2	
80	FE_VIA	Fator de Expansão da Viagem	280	290	11	11 dígitos 6 casas decimais
81	DIA_SEM	Dia da Semana	291	291	1	2 - Segunda-feira
						3 - Terça-feira
						4 - Quarta-feira

	Variável	Conteúdo	Início	Fim	Compr	Códigos
						5 - Quinta-feira
						6 - Sexta-feira
82	TOT_VIAG	Total de Viagens da Pessoa	292	293	2	
83	ZONA_O	Zona de Origem	294	296	3	1 a 517
84	MUNI_O	Município de Origem	297	298	2	1 a 39
85	CO_O_X	Coordenada X Origem	299	310	12	12 dígitos
86	CO_O_Y	Coordenada Y Origem	311	322	12	12 dígitos
87	ZONA_D	Zona de Destino	323	325	3	1 a 517
88	MUNI_D	Município de Destino	326	327	2	1 a 39
89	CO_D_X	Coordenada X Destino	328	339	12	12 dígitos
90	CO_D_Y	Coordenada Y Destino	340	351	12	12 dígitos
91	ZONA_T1	Zona da 1ª Transferência	352	354	3	1 a 517
92	MUNI_T1	Município 1ª Transferência	355	356	2	1 a 39
93	CO_T1_X	Coordenada X 1ª Transferência	357	368	12	12 dígitos
94	CO_T1_Y	Coordenada Y 1ª Transferência	369	380	12	12 dígitos
95	ZONA_T2	Zona da 2ª Transferência	381	383	3	1 a 517
96	MUNI_T2	Município 2ª Transferência	384	385	2	1 a 39
97	CO_T2_X	Coordenada X 2ª Transferência	386	397	12	12 dígitos
98	CO_T2_Y	Coordenada Y 2ª Transferência	398	409	12	12 dígitos
99	ZONA_T3	Zona da 3ª Transferência	410	412	3	1 a 517
100	MUNI_T3	Município 3ª Transferência	413	414	2	1 a 39
101	CO_T3_X	Coordenada X 3ª Transferência	415	426	12	12 dígitos
102	CO_T3_Y	Coordenada Y 3ª Transferência	427	438	12	12 dígitos

	Variável	Conteúdo	Início	Fim	Compr	Códigos
103	MOTIVO_O	Motivo na Origem	439	440	2	1 - Trabalho Indústria
						2 - Trabalho Comércio
						3 - Trabalho Serviços
						4 - Escola/Educação
						5 - Compras
						6 - Médico/Dentista/Saúde
						7 - Recreação/Visitas/Lazer
						8 - Residência
						9 - Procurar Emprego
						10 - Assuntos Pessoais
						11 - Refeição
104	MOTIVO_D	Motivo no Destino	441	442	2	idem ao anterior
105	SERVIR_O	Servir Passageiro na Origem	443	443	1	1 - Sim
						2 - Não
106	SERVIR_D	Servir Passageiro no Destino	444	444	1	1 - Sim
						2 - Não
107	MODO1	Modo 1	445	446	2	01 - Metrô
						02 - Trem
						03 - Monotrilho
						04 - Ônibus/micro-ônibus/ perua do município de São Paulo

	Variável	Conteúdo	Início	Fim	Compr	Códigos
						05 - Ônibus/micro-ônibus/ perua de outros municípios
						06 - Ônibus/micro-ônibus/ perua metropolitano
						07 - Transporte Fretado
						08 - Transporte Escolar
						09 - Dirigindo Automóvel
						10 - Passageiro de Automóvel
						11 - Táxi Convencional
						12 - Táxi não Convencional
						13 - Dirigindo Moto
						14 - Passageiro de Moto
						15 - Bicicleta
						16 - A Pé
						17 - Outros
108	MODO2	Modo 2	447	448	2	idem ao anterior
109	MODO3	Modo 3	449	450	2	idem ao anterior
110	MODO4	Modo 4	451	452	2	idem ao anterior
111	H_SAIDA	Hora Saída	453	454	2	Hora de Saída
112	MIN_SAIDA	Minuto Saída	455	456	2	Minuto de Saída
113	ANDA_O	Tempo Andando na Origem	457	458	2	Em minutos
114	H_CHEG	Hora Chegada	459	460	2	Hora de Chegada
115	MIN_CHEG	Minuto Chegada	461	462	2	Minuto de Chegada

	Variável	Conteúdo	Início	Fim	Compr	Códigos
116	ANDA_D	Tempo Andando no Destino	463	464	2	Em minutos
117	DURACAO	Duração da Viagem (em minutos)	465	467	3	
118	MODOPRIN	Modo Principal	468	469	2	idem ao MODO1
119	TIPOVG	Tipo de Viagem	470	470	1	1 - Coletivo
						2 - Individual
						3 - A pé
						4 - Bicicleta
120	PAG_VIAG	Quem Pagou a Viagem	471	471	1	1 - Você/Sua família
						2 - Empregador
						3 - Isento
						4 - Outros
						5 - Não respondeu
121	TP_ESAUTO	Tipo de Estacionamento do Automóvel ou Moto	472	472	1	1 - Não estacionou
						2 - Zona azul
						3 - Patrocinado
						4 - Proprio
						5 - Meio-fio
						6 - Avulso
						7 - Mensal
						8 - E-fácil
						9 - Não respondeu

	Variável	Conteúdo	Início	Fim	Compr	Códigos
122	VL_EST	Valor do Estacionamento do Automóvel ou Moto	473	478	6	6 dígitos, 2 casas decimais
123	PE_BICI	Por Que Viajou A Pé ou Bicicleta	479	479	1	1 - Pequena distância
						2 - Condução cara
						3 - Ponto/Estação distante
						4 - Condução demora para passar
						5 - Viagem demorada
						6 - Condução lotada
						7 - Atividade física
						8 - Outros motivos
124	VIA_BICI	Se viajou de bicicleta, usou via segregada?	480	480	1	1 - Sim
						2 - Não
125	TP_ESBICI	Estacionamento Bicicleta	481	481	1	1 - Bicicletário gratuito
						2 - Bicicletário pago
						3 - Local privado
						4 - Rua/Local público
						5 - Guardador de rua
						6 - Estação de bicicleta
						7 - Paraciclo público
						8 - Outros
126	ID_ORDEM	Número de Ordem do Registro	482	487	6	De 1 a 183.092