A Modular Approach to Compare Optimization Methods for Bike Sharing Systems

Thomas Barzola- Poma Hild

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : I-MEP2 - Ingénierie - Matériaux, Mécanique, Environnement, Energétique, Procédés, Production
Spécialité : GI : Génie Industriel : conception et production
Unité de recherche : Laboratoire des Sciences pour la Conception, l'Optimisation et la Production de Grenoble

Une approche modulaire pour la comparaison des méthodes d'optimisation des systèmes de vélos en libre-service

A Modular Approach to Compare Optimization Methods for Bike Sharing Systems

Présentée par :

Thomas BARZOLA-POMA HILD

Direction de thèse :

Van-dat CUNG Professeur des Universités, Grenoble INP	Directeur de thèse
Vincent JOST CNRS	Co-encadrant de thèse
Nicolas GAST chercheur, INRIA	Co-encadrant de thèse

Rapporteurs :

Tal RAVIV
PROFESSEUR ASSOCIE, Tel Aviv University
Frederic MEUNIER
PROFESSEUR DES UNIVERSITES, Ecole des Ponts ParisTech CERMICS

Thèse soutenue publiquement le **20 mars 2023**, devant le jury composé de :

Van-dat CUNG PROFESSEUR DES UNIVERSITES, Université Grenoble Alpes, G-SCOP	Directeur de thèse
Christine SOLNON PROFESSEUR DES UNIVERSITES, INSA Lyon, CITI	Présidente
Etienne COME CHARGE DE RECHERCHE, Univ. Gustave Eiffel, GRETTIA/COSYS	Examinateur
Nadia BRAUNER PROFESSEUR DES UNIVERSITES, Université Grenoble Alpes, G-SCOP	Examinatrice
Tal RAVIV PROFESSEUR ASSOCIE, Tel Aviv University	Rapporteur
Frederic MEUNIER PROFESSEUR DES UNIVERSITES, Ecole des Ponts ParisTech CERMICS	Rapporteur

Invités :

Vincent Jost
CHARGE DE RECHERCHE, CNRS, G-SCOP

Contents

1 Introduction **5**
- 1.1 History of Bike Sharing Systems . . . 5
  - 1.1.1 Shared Mobility . . . 5
  - 1.1.2 Bike Sharing Systems . . . 6
  - 1.1.3 Main Problems of Bike Sharing Systems . . . 7
- 1.2 Particular Needs for Optimization of Management Strategies. . . . 8
- 1.3 Organization of this Document . . . 9

2 Critical literature review **10**
- 2.1 Optimization of BSS Management Strategies . . . 10
  - 2.1.1 Strategic Decision Level Problems . . . 11
  - 2.1.2 Management Strategies . . . 11
  - 2.1.3 Existing Works are Hard to Compare and Assess . . . 16
  - 2.1.4 Concluding Remarks on the Optimization of BSS Management Strategies 17
- 2.2 Data Handling in BSS . . . 18
  - 2.2.1 Importance of Real-life Data . . . 18
  - 2.2.2 Few Publications on Data Cleaning and Preparation . . . 18
  - 2.2.3 Variety of Data . . . 19
  - 2.2.4 Data Sources . . . 19
  - 2.2.5 Concluding Remarks on Data Handling . . . 20
- 2.3 Literature Review on Demand Estimation . . . 20
  - 2.3.1 Departure/Arrival Estimation . . . 20
  - 2.3.2 Bike Stock Prediction . . . 20
  - 2.3.3 User Trip Prediction . . . 21
  - 2.3.4 Demand Estimation . . . 21
  - 2.3.5 Concluding Remarks on Demand Estimation . . . 22
- 2.4 Literature Review on BSS Simulators . . . 22
  - 2.4.1 Simulators in a Simulation-Optimization Process . . . 22
  - 2.4.2 Publications on Simulation . . . 23
  - 2.4.3 Concluding Remarks on BSS Simulation . . . 23

3 A Modular Approach to Bike Sharing Systems Optimization **24**
- 3.1 Design and Specification of the Modular Approach . . . 24
- 3.2 Modules Description . . . 28

3.2.1 BSS Data . . . 28
3.2.2 Data Cleaning Module and Cleaned Data . . . 28
3.2.3 Modal Share Estimation Module . . . 29
3.2.4 Scenarios Generation Module and Scenario Data . . . 29
3.2.5 Optimization Module . . . 29
3.2.6 Simulation Module and Realization Data . . . 30
3.2.7 Scenario Visualization Module . . . 30
3.3 Examples of Module Descriptions . . . 31
3.4 Conclusion . . . 32

4 Data Handling and Reconciliation 34
4.1 Introduction . . . 34
4.2 Data Description . . . 35
4.2.1 User Trips Data . . . 35
4.2.2 Bike Stocks Data . . . 37
4.3 Inconsistencies Between Bike Stocks and User Trips Data . . . 40
4.3.1 Number of Stations . . . 40
4.3.2 Relative Bike Stocks and Relative Stock Balance . . . 41
4.4 Enriching Data With Bike Relocations . . . 42
4.4.1 Bike Path Reconstruction Method . . . 43
4.4.2 General Characteritics of the Inferred Relocations (periodicity, tide, balance) . . . 44
4.4.3 Assessing the Number of Inferred Relocations Found . . . 47
4.5 Matching Inferred Relocations and Bike Stocks Data . . . 50
4.5.1 Constructive Alogrithm . . . 50
4.6 Comparison of the Inferred and Given Bike Stocks . . . 52
4.6.1 Visualization of the Rebuilt Stocks for Different Stations . . . 53
4.6.2 Time Precision of New Inferred Stock . . . 54
4.6.3 Towards a Model of the Stock Reconciliation Problem . . . 55
4.7 Conclusion . . . 56

5 Demand Estimation 57
5.1 Introduction . . . 57
5.2 Formal Definition of the Problem . . . 58
5.3 Comparing Estimation Methods . . . 59
5.3.1 General Idea . . . 59
5.3.2 Description of the Simulator . . . 59
5.3.3 Error Measure . . . 59
5.4 Estimation of Time-Independent Parameters . . . 60
5.4.1 Non-Zero-Vehicle-Time Method (NZVT) . . . 60
5.4.2 Excess Demand Rate Method (EDR) . . . 60
5.4.3 Ignore Method . . . 61
5.4.4 Numerical Experiments . . . 61
5.5 Estimation of Time-Dependent Parameters . . . 63
5.5.1 Time Splitting Methods . . . 63

5.5.2 Parameters of the Experiments . . . 64
5.5.3 Comparison of Time Division Methods . . . 64
5.5.4 Impact of the Number of Observations . . . 66
5.6 Conclusion . . . 68
5.7 Limits of the Approach . . . 69
5.7.1 Exogenous Parameters . . . 69
5.7.2 Direct Censor . . . 69
5.7.3 Single Station . . . 70
5.7.4 Poisson Hypothesis . . . 70
5.7.5 Variation on the Comparison Methodology . . . 70

6 Optimization Algorithms and Reproducibility 72
6.1 Objective and Method . . . 72
6.2 Description of the System . . . 74
6.2.1 System Considered . . . 74
6.2.2 Journey Dissatisfaction Function . . . 74
6.2.3 User Behavior Model . . . 74
6.3 Methodology . . . 75
6.3.1 Multi-Agent Simulator . . . 75
6.3.2 Optimization Process . . . 76
6.3.3 Our Assumptions on Unclear Points . . . 78
6.4 Data Provided and Scenario Generation . . . 78
6.4.1 Descriptive Statistics of the Data . . . 79
6.4.2 Note on Scenario Generation . . . 80
6.5 Optimization Results . . . 81
6.5.1 Performance Before the Local Search Optimization . . . 83
6.5.2 Performance After the Local Search Optimization . . . 85
6.6 Evolution of the Optimization Process . . . 87
6.7 Conclusion . . . 88

7 Simulation 89
7.1 Verification Models . . . 89
7.1.1 Need for a Verification Model . . . 89
7.1.2 Requirements of Verification Models . . . 90
7.1.3 Three Examples of Verification Models From the Literature . . . 90
7.2 Simulators . . . 92
7.2.1 Requirement of Simulators . . . 92
7.2.2 Modular Design for a Shared Simulator . . . 93
7.2.3 Advantages of Simulator as a Verification Model . . . 94
7.3 Case Study: Starling Simulator . . . 95
7.3.1 Presentation of the Starling Simulator . . . 95
7.3.2 Verification Process . . . 95
7.3.3 Description of the Verification . . . 96
7.4 Effects of the Modification of the User Behavior Model . . . 96
7.4.1 Changes in the Definitions of the Performance Indicators . . . 96

7.4.2 Changes in the Values of the Performance Indicators 97
7.4.3 Conclusion on the Modification of the User Behavior Model 98
7.4.4 Performance of the Optimization Methods 99
7.5 Conclusion 100

8 General Conclusion and Perspectives **101**
8.1 Conclusion on the Work Carried Out 101
8.2 Perspectives 102

References **105**

Chapter 1

Introduction

In this chapter we present a brief history of shared mobility and bike sharing systems. Then we detail the functioning of the optimization for management strategies of bike sharing systems. Finally, we present the organization of this document.

1.1 History of Bike Sharing Systems

1.1.1 Shared Mobility

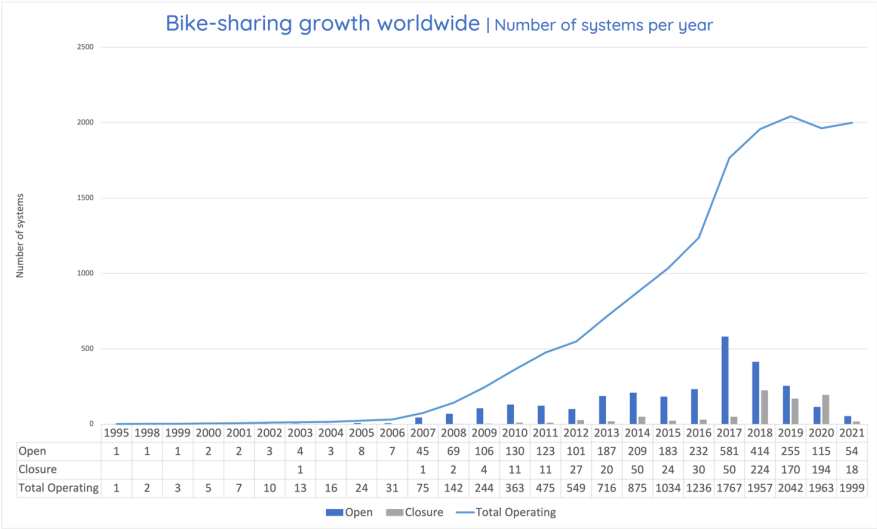
Shared mobility is often defined as a transportation service where resources and infrastructures are shared among users. This includes traditional public transportation, but also newer services, such as bike sharing, rides on demand or carpooling. In the last decades, shared mobility has received continuous attention, since it offers many advantages for all parties. For users, it offers access to a transportation service on an as-needed basis, reducing the cost of personal vehicle ownership (Shaheen et al., 2015). For urban planning, shared mobility can be a solution to the last mile problem and can reduce congestion in dense cities. Shared mobility is often divided into three categories:

- Public transit, the most common shared mobility system.
- Automobile-based modes, where the users share the usage of a car (by car-sharing, carpooling, or by calling a ride).
- Micro-mobility, where the users share small, low-speed vehicles for personal transportation such as bikes or scooters.

In this thesis, we focus on this last category of shared mobility, and more specifically on Bike Sharing Systems (BSS).

1.1.2 Bike Sharing Systems

Bike sharing is a type of shared mobility system, where users share a pool of bikes, using one of the bike at need for short trips. The first bike sharing system was experimented in 1965 in Amsterdam, where the Provo group painted fifty bicycles in white and offered free usage. Even if the program is still active in some parts of the Netherlands, at the time, most of the bikes were stolen or damaged. During the following years, multiple bike sharing systems were experimented but with limited success. At the time, the principal limitation was user identification in order to reduce theft and vandalism. Around 1995, systems were smartened with a variety of technological improvements, which allowed a slight growth for bike sharing systems. Some small systems were implemented, such as Rennes's ''Vélo à la Carte" and Munich ''Call a Bike''(DeMaio, 2009).
In 2005, Velo'v was launched in Lyon, followed quickly by Velib in Paris. Both of those systems were operated by JCDecaux. They represented a mini-revolution for bike sharing systems. For instance, Velib was launched with 20,000 bicycles and registered 26 million rentals in the first year. The systems in Lyon and in Paris are denominated as *Station-Based Bike Sharing Systems* due to their functioning with fixed docking stations.
In a station based bike sharing system, bikes are placed in automated stations with an integrated rental terminal. Any user of the system can identify at a station and ask for a bike, if there is a bike available at this station. Once the authentication is done the bike is automatically released from the docking station, allowing the user to travel. Once the trip is completed, the user must return the bike to a station. She locks the bike to the docking pole, thus ending her rental. The fee paid by the user depends on the kind of subscription purchased and on the duration of the trip made. Such a system solves partially the problems encountered previously. Indeed, the authentication of the users reduces the risk of theft and vandalism, while the automation of the stations saves the labor cost of staffed stations and offers a 24h availability of the system. This management nonetheless implies a bigger investment in infrastructure and has a bigger impact on the city planning.
In 2016, private companies started large-scale *Free-Floating Bike Sharing Systems* in a dozen of Chinese cities (O'Brien et al., 2021): Bikes are not attached to any station, a lock is usually integrated into the frame of the bike, and after the user authenticates, the bike is released to allow the trip. In the German system''Call a bike" in 2000, users could send a text message in order to receive a code to unlock the bike. The recent technological improvements made in connected and mobile devices allowed the systems to grow in terms of number of bikes. While offering an easier mobility experience for their users by removing some constraints of the station-based systems, free-floating systems created new urban management problems, such as irregular parking behavior or difficulty in picking-up for maintenance, leading to 'bike cemeteries' where the bikes are abandoned (BBC, 2018; Chen et al., 2020). As we see in Figure 1.1, today the number of functioning systems is growing and many of them are still growing in terms of number of bikes. China is the country with the biggest number of active systems (see Figure 1.1) (O'Brien et al., 2021) and the biggest number of bikes in circulation (Shanghai system has more than 500,000 actives bikes) (DeMaio & Meddin, 2020).



	1995	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
Open	1	1	1	2	2	3	4	3	8	7	45	69	106	130	123	101	187	209	183	232	581	414	255	115	54
Closure							1				1	2	4	11	11	27	20	50	24	30	50	224	170	194	18
Total Operating	1	2	3	5	7	10	13	16	24	31	75	142	244	363	475	549	716	875	1034	1236	1767	1957	2042	1963	1999

Figure 1.1: Figure extracted from (O'Brien et al., 2021)

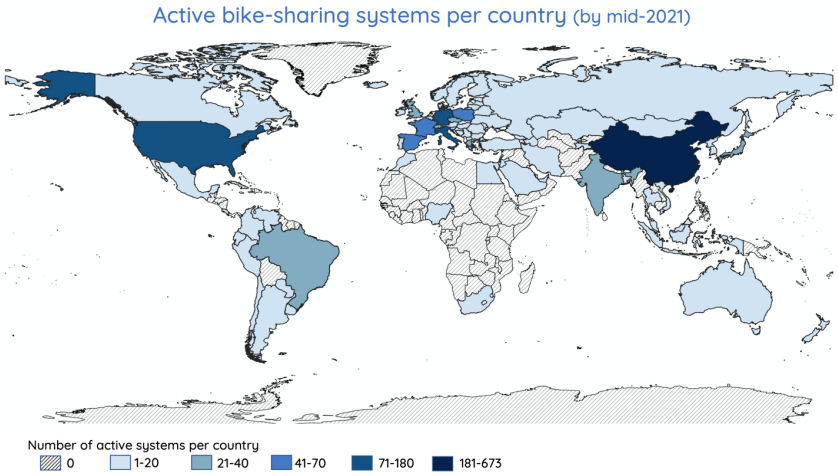


Figure 1.2: Figure extracted from (O'Brien et al., 2021)

1.1.3 Main Problems of Bike Sharing Systems

The implementation of a bike sharing system in a city is a complex process, as so, it faces different critical problems. First, before any further analysis, it is important to understand the impacts that a bike sharing system will have. Depending on the type of bike sharing system, the result on health, mobility patterns or sustainability will be different (Li et al., 2021). This relates easily to urban planning problems, where the objective is to determine how to insert the system into the city (Chen et al., 2020). The business model of the system must also be studied. Transportation is often considered a public service, benefits or profitability must not be ignored and public-private partnership can lead to better performance (Nikitas, 2019). Another problem that must be faced is making the possible users actually use the system. This relates to providing a safe environment for bike usage but also to pricing (Fishman et al., 2014). On a more operational topic, it is fundamental to ensure the usability of the

system. This means making the bikes available to the users when and where they are needed (Laporte et al., 2018) but also designing bikes and infrastructure that meet the user needs (Meireles et al., 2013). On this thesis we focus on the study of optimization method for management strategies. Briefly, management strategies are the way to improve availability of the bikes where they are needed while respecting some operational constraints. A more precise definition of management strategy and a literature review of the methods proposed for optimization is proposed in Chapter 2.

1.2 Particular Needs for Optimization of Management Strategies.

The optimization of management strategies, while being an independent subject, requires two main results from other scientific fields, a demand estimation and a evaluation process.

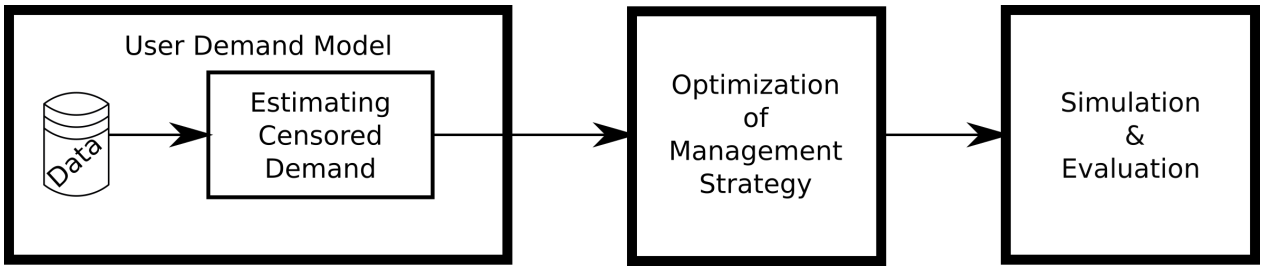


Figure 1.3: Minimal workflow of the optimization of management strategies in BSS

First, as any optimization work, data is required. In the case of the optimization of BSS management strategies, we need to have access to a representation of the modal share of the system (the number of travelers that have chosen the system). This modal share, corresponds to the users for whom the system must be designed, it is represented by the first block of Figure 1.3.
Second, there is a need for a way to assess a management strategy. Since the system is complex and involves lots of different interactions between agents (users, bikes, stations), it cannot accurately be modeled within the optimization process. Thus, a "solution checker" must be designed to assess the performance of a management strategy, represented by the third block of Figure 1.3. This adds to the fact that various criteria must be considered to qualify a management strategy, but only few can be chosen to do the optimization. A "measure instrument" is thus needed to assess correctly the impact of the new management strategy on the system and on it's environment.

This needs (demand model, optimization method, assessment method) have been identified by the authors in the optimization literature but, because it is not the main subject in their work, few studies have been dedicated to those subjects.

1.3 Organization of this Document

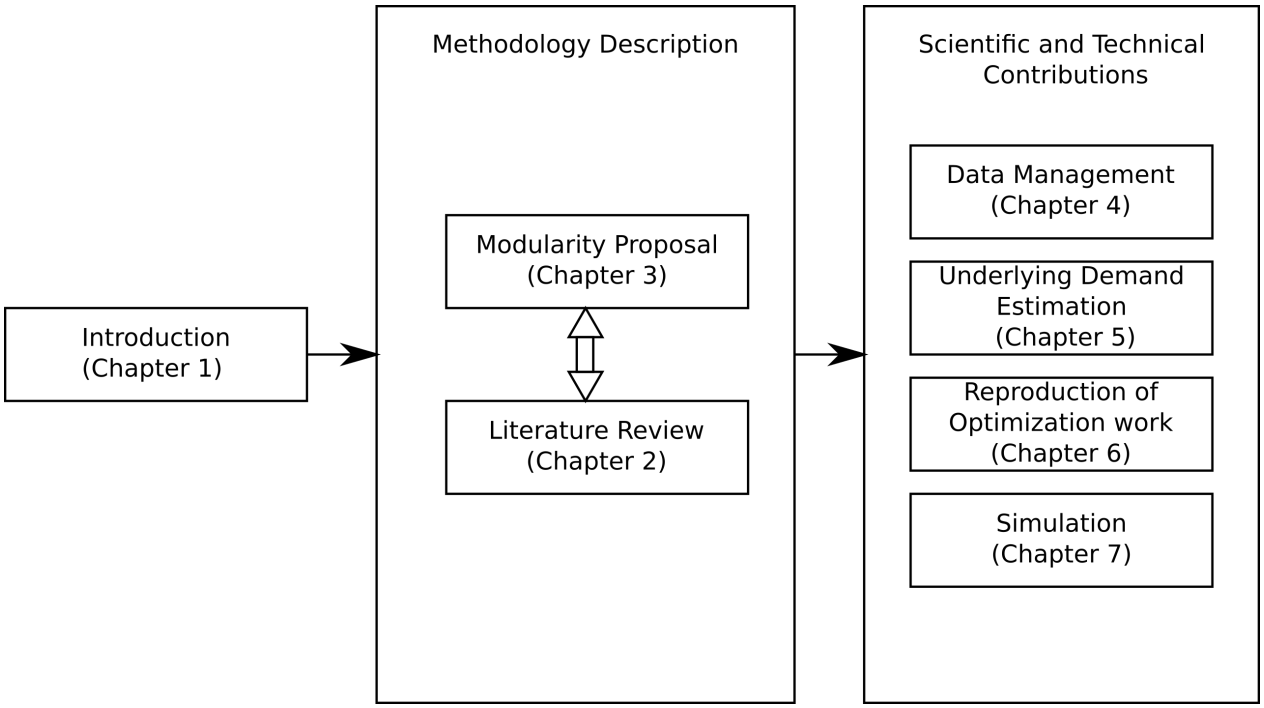


Figure 1.4: Organization of our contributions in this document

In Figure 1.4 we present the organization of this document. In Chapter 2, we present a literature review on optimization in bike sharing systems, we also present a literature review of the different modules introduced in Chapter 3, for the modules that have few literature we detail the literature on closely related topics. In Chapter 3, we describe the need for a modular approach and describe the interfaces between each modules. Chapter 2 and Chapter 3 must be read in regard to each other, Chapter 3 is constructed on the literature review presented in Chapter 2, while Chapter 2 is organized according to the method proposed in 3. The reading order has little importance.
Similarly, Chapters 4 to 7 present our technical and scientific contributions to the different modules presented before. All those chapters are independent, they are presented in the order they appear in Figure 3.1 but this order should not impact the understanding.

Chapter 2

Critical literature review

In this chapter, we present a literature review on the optimization of BSS management strategies, along with a new classification of the corresponding articles. We categorize the publications depending on the method used to relocate the bike and the performance objective chosen. From the review on optimization literature, it appears that contributions in the corresponding publications are not always ''reused" within other publications. We present different reasons that could explain this lack of comparison. As a result, within the limit of our knowledge, it is difficult to find and maintain a unique solution that would fulfill all the requirements of our decomposition of the approach into independent modules as they will be specified in Chapter 3. We thus present a quick literature review on the different contributions proposed to the different modules presented in Chapter 3.

In the first period of the life of Bike Sharing Systems, the systems were relatively small and were not (or scarcely) monitored. With the third generation of bike-sharing systems (DeMaio, 2009), the systems grew in size and in number of users. With this new generation, new research needs arise in different fields. This new generation of systems also made available larger datasets, allowing researchers to propose new insights into the systems. More recently, another type of bike sharing system appeared, without stations and with connected bikes. Those systems allowed for even bigger sized systems, inducing new needs for managing such systems. In this chapter, we review the works done in three fields, being: the optimization of the BSS management strategies, the analyses of the data provided by such systems and the prediction of demand at bike sharing stations.

2.1 Optimization of BSS Management Strategies

We base our state-of-the-art mainly on two extensive literature review on Bike Sharing Systems optimization, (Laporte et al., 2018) and (Shui & Szeto, 2020). Both authors keep the classic decomposition of problems into three decision levels: Strategic, Tactical and Operational. However, the problems studied are not always classified in the same category, they may integrate several decision levels. Here, to categorize problems, we keep the strategic level as a

single category, but mix the tactical and operational levels to come up with new categories built, on the management strategies and the proposed solving methods. We completed our literature review with 80 other articles in this area, some of which are referenced by the two extensive studies.

2.1.1 Strategic Decision Level Problems

2.1.1.1 Bikeway Design

The bikeway design problem consists in determining the type and placement of bikeways. The bikeways differ in their types, each type with its own characteristics (width, cycling risks, cycling comfort, maintenance costs, impact on vehicular traffic etc.). The publications on this problem can have two main objectives: increasing the modal share of bicycle (Mesbah et al., 2012; Sohn, 2011) or reducing the costs (Duthie & Unnikrishnan, 2014; Klobucar & Fricker, 2007). Those objectives are usually combined with different constraints to form a service network design problem.

2.1.1.2 Station Location

The station location problem (or station network design problem) focuses on determining the placement of stations throughout the city. The key indicator often taken into account is demand coverage (Çelebi et al., 2018; Garcia-Gutierrez et al., 2014). The maximization of this indicator as objective is traditionally balanced by economical objectives or limited by budgetary constraints (Correia & Antunes, 2012; Garcia-Gutierrez et al., 2014). Solving this problem, authors often use other data sources to guide their decision. For instance, geographic information or demographic data can be used to assess the quality of cycling infrastructure or the distribution of the transport demand between different transport modes (Bourgeois, 2017; Rybarczyk & Wu, 2010). This problem, similar to a hub location problem, is often associated with a dimensioning problem which consists in determining at the same time the total number of docks in each station (Nair & Miller-Hooks, 2014; Wang et al., 2016). With the new types of bike sharing system, new constraints have been added to this problem. As examples, the constraints of sizing and placing charging points for electric bikes (Li et al., 2016) and the new problem of deciding which station is fixed and which is for free-floating bikes in hybrid free-floating/station-based systems (Albiński et al., 2018).

2.1.2 Management Strategies

Once the strategic level problems are fixed, the availability of bikes (and docks) is quickly identified as a main issue for operating bike sharing systems. Indeed, two demand patterns are usually described in bike sharing system. The tide is a pendular pattern of journey during the day (home-to-work and reversely for example) and the gravity (riders will rather ride downhill than uphill), a given station can have more user departures than arrivals, or the

opposite. All of those movements leave areas with high demand without bikes, while other areas have an overload of bikes making bike returns impossible. Many practical methods have been proposed to solve this real-life problem, which can be summarized by the following question:

> ''How many and how to place the bikes in the stations in order to optimize the system's performance?"

We choose to gather all the solutions to this problem by the term of *Management Strategies*. In this problem, we can distinguish two sub-problems: The *Inventory Level Problem* and the *Bike Relocation Problem*. Those two sub-problems will be the first criteria to classify the different solutions.

2.1.2.1 Inventory Level Problem

The *Inventory Level Problem* can be stated as follows:

> Given a set of stations, determine the number of bikes to place in stations in order to ensure a good performance over a given time-horizon.

Some publications work at solving this problem separately. In (Datner et al., 2017), the authors try to minimize the total time lost by the users looking for bikes or docks and, in (Schuijbroek et al., 2017), the authors compute the number of bikes to guarantee a defined service level. But, the vast majority of the works in this review aggregate the objective of both the Inventory Level Problem and the Bike Relocation Problem, to solve them simultaneously. Since the performance of a system is commonly measured by its ability to serve trips, most of the objectives are service oriented. We can find, for example:

- The number of served users (Liu et al., 2016; Shu et al., 2013; Yan et al., 2018),
- The number of users leaving the system due to the lack of bikes (Alvarez-Valdes et al., 2016; Caggiani & Ottomanelli, 2012; Contardo et al., 2012),
- The service levels, being the probability that a user finds a bike at a given station (Clemente et al., 2013; Schuijbroek et al., 2017),
- The number of unused bikes per time step (Angeloudis et al., 2014).

Similarly, user-oriented objectives are often proposed as well. (Datner et al., 2017; Kaspi et al., 2014, 2016) use the excess time, a classical performance measure for transportation systems. More complex objectives using one or more of the following items have also been put forward:

- The number of failed attempts to find a bike,
- The number of failed attempts to return a bike,
- The time to find a bike for users waiting at a station,
- The time to return a bike for users waiting at a station,
- The price paid by the users who rent a bike at a station,
- The holding costs of keeping a bike in a station.

Among those objectives, (Raviv et al., 2013) proposed a *user dissatisfaction function* which has been reused in many other works (Chung et al., 2018; Repoux et al., 2019). The main benefit in considering an aggregated cost is, that it will easily sum up with the Bike Relocation objective. Nonetheless, considering such an objective has a massive disadvantage, the aggregation makes the costs interchangeable, asking to find the correct trade off between the objectives. For example, we can exchange waiting time for a user if it reduces the holding time of a bike.

Finally, some station-based objectives are also considered. For example, (Barth et al., 2004; Kek et al., 2006) propose to measure the *Zero-Vehicle-Time* and the *Full-Port-Time*, the sum of the time during which a station is empty (resp. Full). (Chiariotti et al., 2018) measure the survival time of a station, the time before the station runs out of stock or becomes full. (Fricker & Gast, 2012) only counts when an empty or full station occurs. The advantage of those objectives is that they are easily measurable in real-life systems.

Nonetheless, it must be noted that some authors propose original objectives trying to simplify the Bike Relocation Problems. In their work, (Haider et al., 2018) propose to simplify the bike relocation problem by incentivizing the users to place the bikes in a way that would maximize the imbalance in a set of stations. Similarly, (Schlote et al., 2015) measure the probability of an empty station having a non-empty neighbor station.

2.1.2.2 Bike Relocation Problem

The *Bike Relocation Problem* could be stated as follows:

> Given a target level (number of bikes) at each station, how to move the bikes between stations to reach the target levels while reducing the operations cost?

On this problem, we classify the Management Strategies based on the working force employed for moving the bikes. On one hand, we have *operator-based* management strategies, where bikes are moved by employees of the system provider. These solutions provide more controllable solutions but come with a greater economic cost. On the other hand, we find *user-based* strategies that try to incentive the users of the system to make some decisions to move the bikes across the station network.

Operator-based Strategies
We can again subdivide operator-based strategies by the considered planning horizon. In Static Bike Relocation Problem (SBRP), the workers move the bikes during a time period when the movement of bikes by the users can be safely ignored (usually during the night). (Raviv et al., 2013) is often considered as the first publication on this problem, the authors propose a complete Integer Linear Program to model the problem and solve it by CPLEX. This problem has then been studied by different authors, introducing new constraints and new objectives. Many of the authors minimize the economic cost of operations (Ho & Szeto, 2014; Szeto et al., 2016; Tang & Dai, 2018), or the time needed for operations (Bulhões et al., 2018; Luca Di Gaspero et al., 2016; Rainer-Harbach et al., 2013), while other authors add new possibilities for operators, such as (Wang & Kim, 2018), who propose to remove broken bikes from the stations. When describing solutions to the SBRP, it must be stated that (Dell'Amico

et al., 2018) is the only publication we found to deal with stochastic demands and doing the optimization directly from the demand laws.

In the opposite, in Dynamic Bike Relocation Problem (DBRP), the operators move the bikes while the system is in use. When studying this problem, the authors deal with more detailed objectives than in the SBRP. Obviously, economic costs are often considered (Arabzad et al., 2018; Kek et al., 2009; Zhang et al., 2017), but many authors only try to find feasible routes without minimizing an objective (Boldrini & Bruno, 2017; Contardo et al., 2012; Repoux et al., 2019). Indeed, to consider dynamic bike relocation, the target inventory levels must be defined (almost) continuously, thus making the problem really hard to solve. Nonetheless, some authors propose new objectives such as (Bourgeois, 2019), who tries to minimize the total number of relocation trucks used.

User-based Strategies

On user-based strategies, users are incentivized to change their initial routes to place the bikes in the system. Different strategies have been tried in order to incentivize the users. Some authors define a price to be paid by the user depending on the stations used (at the start or at the end) (Chemla et al., 2013; Pfrommer et al., 2013) or on the trip planned by the user (Waserhole & Jost, 2016). As far as we know, only the work of Waserhole and Jost (Waserhole et al., 2013; Waserhole & Jost, 2012, 2016) gives an exact mathematical formulation of the problem they consider. In some publications, the authors not only propose to divert users from their original trips, but also to incentivize new trips to rebalance the bikes (Chung et al., 2018). To our knowledge, the ''bike angels" (users who relocate bikes to be part of a leaderboard) as implemented in New-York and studied in (Chung et al., 2018) give surprisingly good rebalancing performances.

2.1.2.3 Aggregating the Inventory Level Problem and the Bike Relocation Problem

As said previously, many authors do not make the distinction between the Inventory Level Problem and the Bike Relocation Problem, they solve both as one single problem. This explains why so many authors consider aggregated economic costs (Boyacı et al., 2015; Fan et al., 2008; Raviv et al., 2013; Zhang et al., 2017). This allows to easily compromise between the satisfaction of users and the relocation costs. However, it is hard to find a good trade-off between these two objectives in a real-life context. As example, (Szeto et al., 2016) try to compromise between the time spent on bike relocation and the number of users that did not find a bike. Other authors choose to model this double problem as a bi-level problem (Caggiani & Ottomanelli, 2012; Chiariotti et al., 2018; Di Febbraro et al., 2012; Haider et al., 2018), this presents the advantage of having a more applicable result but comes with the harder difficulty to solve to optimum.

2.1.2.4 Solving Methods

Until now, we did not discuss any solving methods. Operations Research publications are often focused on solving methods for complex problems. The literature on bike sharing systems is no exception to this trend, with many publications focused on defining a new complex problem and solving it with a new method. A significant part of the literature presents BSS management strategies as variations of the Vehicle Routing Problem and, new metaheuristics are then designed to solve them. In this category, we can for example cite (Teodorović & Dell'Orco, 2005) presenting bee-colony-optimization to solve a static relocation problem, or (Mukai & Watanabe, 2005) using a string-equilibrium to find a near-optimal solution to a dynamic relocation problem. On static or dynamic operator-based strategies, the problems considered can be mathematically formulated (MILP, Stochastic Programming, ...) and solved. We can find in this category, works such as (Raviv et al., 2013; Sayarshad et al., 2012) where the authors express the problem of maximum gain as an MILP and solve it with a commercial solver, or (Contardo et al., 2012) who uses column generation and Benders decomposition to provide solutions to their MILP. Some authors use known metaheuristics to give solutions to the problem they propose. For example, (Repoux et al., 2019) uses a Markov decision process to solve a dynamic relocation problem, or in (Datner et al., 2017) where the authors use a guided local search to solve an inventory level problem.

A few authors solve their problems with ad-hoc optimization methods. It is often made to take into account a particularity of the system and the resulting problem they consider. We can cite (Barth et al., 2004; Gómez Márquez et al., 2021), who propose decision-aid making methods in order to build solutions that fit well to the simulations they make of their systems.

On the user-based strategies, mathematical formulations (MILP, Stochastic Programming, etc.) are not suitable to capture the dynamic behavior of those systems. The majority of the authors considering these strategies present an algorithmic description of the behavior of the users and use simulations to assess and optimize the efficiency of the management strategies.

Once the optimization is made, the remaining question is: how to confirm the results found? Indeed, in order to have a solvable problem, many assumptions are made on the systems are not always verified in the real systems. There is thus a need for a verification method. Some authors use the same mathematical models they used for optimization as a reference to the real systems (Raviv et al., 2013; Yan et al., 2018). Others develop simulators able to represent the real systems. These simulators can be used for optimization (Datner et al., 2017) or can be used only for evaluation (Waserhole & Jost, 2016). To our knowledge, no author used a third-party simulator to check their solutions. It must be noted that some publications present an evaluation through real-life systems, such as (Barth et al., 2004) who observed a reduction of the number of relocations in the University of California-Riverside campus. Or (Singla et al., 2015) and (Repoux et al., 2019) who respectively implemented their management strategy in Mainz, Germany and Grenoble, France.

2.1.3 Existing Works are Hard to Compare and Assess

The analyzed literature is broad and propose a great variety of management strategies to optimize the functioning of a bike sharing system. But, the published results do not seem to be reused from one study to another. Each research team presents a new study with little or no relation to the previous studies. This leads to a hard comparison of the existing works. We explain this with six issues presented below.

Variety in the types of problem: As presented in Section 2.1, to solve the same practical problem, the technical solutions proposed to implement management strategies differ strongly from one publication to another. As example, the impact of a user-based relocation strategy cannot be easily compared to the impact of an operator-based one made with autonomous vehicles. Indeed, even if some common criteria can be chosen, the impacts of user-based strategy (user-dissatisfaction, predictability) is a different kind from the impacts of operator-based ones (costs, environmental, traffic).

Variety in the objective: We showed in the previous Section 2.1, that the objective of a management strategy can always be decomposed into two sub-objectives, one for the Inventory Level Problem, one for the Bike Relocation Problem. For both of those sub-objectives, many indicators can be chosen (failed user attempt, excess-time, routing cost, etc.) and this complicates the comparison of the works. Indicators cannot be converted one into another, and the conjunctions of the two sub-objectives is almost unique to every publication. One could argue that economical costs could lift this problem but, as said previously, using economical objectives comes with the charge of finding the fair prices to aggregate the costs.

Variety of the work contexts: The behavior (and thus the performance) of a bike sharing system depends strongly on the city they are implemented into but, in the literature review we made, either the data used comes from one of the various systems across the world at different time periods, or are completely theoretical. Over the 82 publications selected, as studying BSS management strategies, the system the most studied is the Wien system with 5 publications. This variety of data origins adds up to the variety of data preprocessing methods used in the literature. This variety makes it hard to re-obtain the same usable data from historical data. Of course, some authors have proposed data as references (Dell'Amico et al., 2014; Vogel et al., 2011) but, because the data proposed were too specific to a certain problem, and could not be adapted to new management strategies. Their attempt to provide classical instances for comparison did not have the impact they hoped.

Variety of the demand models: It has quickly been identified by many authors that the historical data are not representative of the real demands the systems have to face (O'Mahony & Shmoys, 2015; Raviv et al., 2013; Waserhole & Jost, 2016). The historic demand for bikes (resp. docks) is observed if and only if it is satisfied. To estimate the real demand, different solutions are used across the publications but no common method is used, adding differences between the studies.

Lack of a common assessment process: To optimize a BSS management strategy, some assumptions are made on the behavior of the system, the performance depends on these hypotheses. This implies that the solution itself is not enough to assess the performance of

the optimization process. To do this, a checker must be used and, again, when reviewing the literature, no checker appears to be common between publications.

Lack of details in global approaches: In all the works presented in Section 2.1, the authors, to produce their work must solve diverse sub-problems. But, since it is not the main topic of their work and the publication needs to be concise, they often do not detail the methods used to solve those sub-problems. This hardens for researchers to propose a work with the same demand model, work context and assessment process, and thus generate again and reuse the solutions.

Reproducibility issue: Even if we assume that we overcame the previously presented difficulties, a more practical problem remains. Apart from comparability, a reproducibility concern appears, we use the guidelines proposed by (Desuilbet et al., 2019) as references for reproducible research. Of all the publications presented here, 5 publications have easily accessible data (Bulhões et al., 2018; Datner et al., 2017; Dell'Amico et al., 2014, 2018; Forma et al., 2015; Pal & Zhang, 2017), and only one has both data and code easily accessible (Pal & Zhang, 2017). Nonetheless, it must be noted that other authors tried to provide their data. But, either they directly used the data given by the operator of the system, giving the responsibility to the operator to make the data available. Or they made the data available on a personal website which by the time we write this thesis is no longer available. In this literature review, (Pal & Zhang, 2017) are the only authors who make their data available in a long-term storage (github). This last issue is, from our understanding, the major obstacle to a good comparison of the works. Indeed, without easy reproducibility, none of the existing works can be compared to another. Readers can refer to Chapter 6 for a deeper example of different problems that one can face when trying to reproduce an existing work.

2.1.4 Concluding Remarks on the Optimization of BSS Management Strategies

As we saw through this literature review on various bike sharing systems, the optimization of BSS management strategies has been widely studied since the booming of such systems. The authors proposed a large variety of solutions in terms of management strategies and, in terms of optimization methods. In spite of this diversity of studies, none of those optimized management strategies seems to emerge as a strategy with a state-of-the-art performance. Apart from the complexity of the problems, this could be mainly due to a lack of a fair comparison between the studies done with a common methodology. This lack of comparison has a variety of explanations: in the first place, many objectives are considered throughout the literature. The purpose of those objectives is relatively similar but, the remaining differences make it hard to compare performance without an appropriate study of the behavior of those objectives. Secondly, the origins of the data used are different from one study to another. If real systems are used as examples, the systems chosen are not the same among the authors and, the provided data instances cannot be easily reused because they are too specific to their own problems. These add to the fact that, often, the demand data extracted from the systems are processed to make more reliable and predictable data. But, this process is not standardized between the studies. Besides, the methods for assessing the results of a

BSS management strategy are not common either. Some authors use simulation in order to gain insights into the system, while others use only the objective function as a performance indicator. The conjunction of these difficulties added to the fact that neither code nor data are reusable, makes it hard for a new researcher to draw a global conclusion from the literature and compare her work with others.

To help overcome these difficulties, we propose to decompose the publications on the optimizing of BSS management strategies in four aspects. Each aspect can be studied independently, but all four are needed to (re-)produce a publication (see Figure 3.1 in Chapter 3). In the current literature, all these aspects exist and are regularly treated but, often, the solutions used for each aspect are not detailed or even omitted.

2.2 Data Handling in BSS

2.2.1 Importance of Real-life Data

In our study of bike sharing systems, many authors confirm our thought: working on data from a real system is fundamental. The most common use of real-life data is having information on the state of the system at each time period over a time horizon. In order to provide a better quality of service (Clemens, 2018), every system provides for its users a way to visualize the number of bikes available at each station. There is also a high number of apps that use such data to provide new experiences for the users: a number of classical apps provide availability of bikes in stations either in real time or in near future (Javier, 2022). Some other apps are more diverse, such as (Keser, 2017) who proposes to ''Help Bike Enthusiasts to Find their Mates''. Finally, the largest use of those data is made by scientists or data enthusiasts who propose research or small projects on those systems. For instance, the New-York system has a group of citizens dedicated to small projects on those data:''the Citibike Hacker Group" (Hebbert, 2022).

2.2.2 Few Publications on Data Cleaning and Preparation

In this thesis, we mainly focus on scientific publications. It must be noted that, even if the number of scientific publications is important, and they use numerous data, few publications deal with data-cleaning and data preparation. This task is time-consuming but nonetheless a fundamental part of the work. Many of the works made to prepare data were mainly meant for prediction work (Borgnat et al., 2011; Come et al., 2014), and most of the preparation is outlier removal.
Another field where data preparation is often detailed is Big Data related research domain. In (Bordagaray et al., 2014), the authors define four steps in raw database preparation and, propose a method to prepare data for further analysis. They focus specifically on data cleaning to remove bike trials (pickup-dropoff, to test if the bike is functional) and bike substitution

(changing bike to remain below the time limit). Nonetheless, some works on data cleaning and preparation can be found in recreational projects. We can present the work of ("Bike Prediction - RStudio," 2022) or (Keser, 2021), who propose excellent work on preliminary data cleaning, but do not work further on quantifying the quality of the data. The only official group doing similar work we found is (Bike share research contributors, 2022).

2.2.3 Variety of Data

Various publications studied the factors impacting user behavior in bike sharing systems (Eren & Uz, 2020; Guo et al., 2017). We use those results as the data to consider in order to have a good insight into a system. The first kind of data used is system-related. It consists of trips made by the users along with the state of the system at every time period (Citibike, 2022) (number of bikes in stations, open/close stations, etc.). The North American Bike-share association (NABSA, 2022), made an important effort in standardizing the format of the data and listing the uses of those data.

The second type of data often used is meteorological data. Indeed, this strongly impacts the usage of a bike sharing system (Gallop et al., 2012). The meteorological data is often available, scrapped from an open API or coming from a private weather company. Similarly, the city topography has a huge impact on the trips and stations the users choose (we all would rather ride downhill than uphill). Nonetheless, we can estimate that the observed trips are a good representation of the topology of the city (in particular in New-York, which is a rather flat city).

The next type of data that impacts the behaviour of users is the interaction between the bike sharing system and the other means of transportation (Martin & Shaheen, 2014). These data are more complex to obtain and to use. Even though these data allow for a better understanding of the city dynamic, we chose not to investigate these data sources.

With all that being said, we choose to work with only two data sources:

- User trip data,
- Bike stock data.

2.2.4 Data Sources

As said previously, with the Open-Data movement, various system operators give access to the trip history with all the trips longer than 60 seconds (Citibike, 2022). These data being first-hand from the operator of the system, we always consider this data source as trustworthy. Secondly, the operators provide the state of each station every 5 minutes. These data can be exploited to develop user applications. These data are scraped by some researchers (Come, 2022; Justin Tyndall, 2020), and are often processed for storage and/or analysis. These resulting data are less trustworthy and more subject to errors.

2.2.5 Concluding Remarks on Data Handling

Concerning data preparation, while being a fundamental topic, the scientific literature is not very extensive. Most of the work is from unofficial organizations ("Bike Prediction - RStudio," 2022; Keser, 2021). While those authors propose interesting methods to clean the data, they do not present a scientific methodology to assess and reuse their method. It is interesting to note that the works proposed by those authors from unofficial organizations are often more reproducible than scientific works. In majority of those works, both the code and the results are easily available. **In Chapter 4 we expand the concept presented in (Kranish, 2021) to propose an entire processing of the NY BSS data.**

2.3 Literature Review on Demand Estimation

In the studies of bike sharing systems, estimating and predicting user interactions with the system has been quickly identified as a major challenge for the development of the system. Indeed, it is useful for the operator to have a predictive view of the needs of each station. This helps to design the initial bike stocks and the repositioning operations. For users, having a prediction of the availability of bikes (or docks) at their departure (or arrival) stations represents a great improvement in the service quality. This scientific challenge, associated with the fact that BSS data is perfect for illustrating machine learning and statistical tools, makes available a large variety of works on estimation and prediction. We classify the works hereafter depending on the type of data they try to estimate or predict.

2.3.1 Departure/Arrival Estimation

Historically, the first type of estimation made on those systems was the number of bike pickups/ dropoffs (Borgnat et al., 2009). As said previously, this is a major indicator for bike sharing operators to organize operations and the system. This forecast problem has thus often been studied either to present new prediction methods such as (Pan et al., 2019). They present a deep Long Short-Term Memory (LSTM) model and test it on this prediction problem to explain factors impacting departures and arrivals. In the same way, (Gallop et al., 2012) uses a SARIMA along with weather variables to predict departures of users in New-York.

2.3.2 Bike Stock Prediction

Another prediction problem often studied is the prediction of bike stocks. This problem has quickly been identified as a key problem to be solved (Gast et al., 2015; Kaltenbrunner et al., 2010) to improve the understanding of the system. As in the departure/arrival prediction, this problem is studied with new prediction techniques. For example, we can cite (Ashqar et al., 2017), who predicts stocks with a non-linear machine learning model (LSBoost algorithm), or

(Wang & Kim, 2018), who uses an LSTM model. The problem is also studied to have precise prediction by introducing new regression variables. In this category, (Rudloff & Lackner, 2014) determine what factor impact the bike stock of the stations using Hurdle and Negative Binomial regressions. This problem can be derived into an availability prediction problem such as in (Yoon et al., 2012), where the authors try to predict if a given station will be non-empty and non-full in near future. Or as in (Chen et al., 2016) who uses a monte-carlo simulation to predict if the station will be empty or full.

2.3.3 User Trip Prediction

This easily leads to trip prediction, where the objective is to predict where the users go and when. Except from (Loaiza-Monsalve & Riascos, 2019), all the publications on trip prediction associate this prediction problem with a clustering problem (Briand et al., 2016; Come et al., 2014; Randriamanamihaga et al., 2013). The authors cluster each station into a category, depending on its demand patterns. This allows to have at the same time, a description of the station demand pattern and of the user's behavior.

2.3.4 Demand Estimation

In the previous paragraphs, we emphasized prediction types that allow a better understanding of the system. In reality, the optimization community shows a slightly different need. Indeed, as said previously, there is already a relocation strategy at work in all the working systems. Thus, the number of trips (or departures, or stock) registered in the system corresponds to the demand conditioned by the fact that this particular strategy has been used. In particular, all the users that did not find a bike at their initial departure station or did not find an available dock at their arrival station, are not correctly registered in the system. If this demand is used to forecast and schedule relocation operations, it will lead to possible wrong decisions. This highlights the need for methods able to predict the "underlying" demand. If we suppose that the underlying demand is divided between the *observed demand* and the *censored demand*, it should be possible to estimate the underlying demand from the observed demand. We give a more precise definition of the censored, underlying and observed demand in Chapter 5. To our surprise, we found few publications dealing especially with this problem. Many authors (Raviv et al., 2013; Repoux et al., 2019), before doing their optimization, make such an estimation. But, since the main topic of their work is optimization, they do not make a real study of this estimation problem. In (O'Mahony & Shmoys, 2015), the authors present briefly a method to give a lower bound on demand by using the demand observed when the station is not empty as estimator. We give more detail on this method in Chapter 5. To our knowledge, only two publications present a real study of this problem of underlying demand: (1) (Negahban, 2019) input different types of underlying demands in a simulator and compares the results of their simulation to the same measurements in the real system. But, due to the lack of "modularity" (the code is not available easily and the study is complex), their work can hardly be re-used for another system or other indicators. (2) (Liu & Pelechrinis, 2021) propose an ad-hoc estimation method and test it over real-life scenarios and one simulated

scenario. They then compare this estimation method with an "observed" demand predictor but not to other "censored" demand estimators of the literature.

2.3.5 Concluding Remarks on Demand Estimation

When studying the demand estimation literature, we see that there are many publications on different estimation problems, such as the estimation of user trips or bike stocks. All those topics and publications share the same need for data already cleaned and prepared. The question of estimating the underlying demand from historical data has been identified in different optimization publications but, only two works focus specifically on this problem. Those two publications propose original methods to solve this estimation problem but, none of those two give a precise comparison of the estimation methods proposed by the different authors. **This comparison of methods should be a crucial scientific and technical work to conduct. In Chapter 5 we propose a first work on this subject.**

2.4 Literature Review on BSS Simulators

BSS are often considered as complex systems in reason of the number of interactions and the difficulty of the choices involved. The simulation of BSS has been quickly identified in the literature as an important scientific challenge. In Chapter 7 we will give more detail on the need for simulation in the optimization of BSS management strategies. In the rest of this section, we present a small review on the existing simulators. We classify those simulators depending on their use in literature.

2.4.1 Simulators in a Simulation-Optimization Process

Modeling precisely the interaction between the objects (bikes, stations, bikers, bike relocation operators, etc.) in the system is not always possible with exact mathematical formulation. This limitation is often lifted by the use of a simulation-optimization loop. For those reasons, simulators have often been used in optimization works, especially in dynamic relocation (Chemla et al., 2013; Kek et al., 2006; Repoux et al., 2019). Those simulators are most likely multi-agent simulators where each agent is described by its interactions with the other agents. Simulators can nonetheless have multiple forms. As an example, the authors of (Caggiani et al., 2018, 2019; Caggiani & Ottomanelli, 2012) used a stochastic model where each user is not described as in multi-agent approaches, but global tendencies are simulated to optimize BSS management strategies both static and dynamic. Such a solution allows to simulate the adaptation of the demand to the performance of the system. Finally, some authors used classical simulation-optimization methods. In this category we can present the work of (Clemente et al., 2013) who uses a Petri Net model, to maximize the service level through user relocation.

Nonetheless, in the vast majority of the optimization publications, when a simulator is used, the simulator is not considered by the authors as a stand-alone result of their work. Indeed, few of the authors do a publication especially on the simulator (discussing the simulation assumptions, performance, precision. . . .).

2.4.2 Publications on Simulation

The first publications on the simulation of BSS had to deal with the need to run in restricted time. This explains the size of the simulated systems (5 stations for (Caggiani & Ottomanelli, 2012), 9 for (Kek et al., 2006) and 25 vehicles maximum in (Barth & Todd, 1999)) that used to be limited. For those simulators, the validation was easier due to the small size of the systems. It is possible to compare the results of the simulation to the behavior of the system observed in real-life. This validation was made in (Barth & Todd, 1999; Kek et al., 2006). More recently, different authors presented simulators able to simulate larger systems in reasonable time. As example, in (Romero et al., 2015), the authors present a simulator developed in Python able to simulate a system with 15 stations, in a multi-modal transportation network. The model is validated by comparing the modal share simulated to the modal share censored in the Spanish city of Santander. The limitation of the system size was also lifted thanks to the usage of existing simulation frameworks. Those frameworks are optimized to do simulations of large systems. In (Lin & Liang, 2017; Saltzman & Bradford, 2016) the authors use Arena to define their simulation process, the simulator is validated through a comparison to indicators obtained on a real-life system. To give an idea of the size of the system, in (Saltzman & Bradford, 2016) the authors are able to simulate and visualize a system of 35 stations. It is lately that the simulators were able to simulate much larger systems with good accuracy. In (Fernández et al., 2020), the authors simulate a system with 174 stations and 1800 vehicles, while in (Coretti Sanchez et al., 2022), the authors are able to simulate more than 70 000 trips with excellent accuracy. With this third generation of simulators also came solutions to easily distribute, maintain and update the simulator for new needs. We can present the works of (Leblond et al., 2020) and (Coretti Sanchez et al., 2022) whom the authors work actively on the distribution and maintenance of their simulator.

2.4.3 Concluding Remarks on BSS Simulation

We saw, through this part of the literature review, that there is a need in the optimization community for BSS simulators. This need has been answered by various authors in optimization publications but, has rarely been studied as a work independent of optimization. The first period of simulators were not able to simulate precisely a large system, making it hard to use the result on current, real-life systems. In the second period, new simulators appeared. Those new simulators were able to simulate larger systems but, are also easily available to re-use them in different works. In Chapter 7 we present a description of the different parts a simulator must implement, and we use a simulator to validate the results of Chapter 6.

Chapter 3

A Modular Approach to Bike Sharing Systems Optimization

A literature review on the different modules necessary to design the optimization of BSS management strategy has been presented in Chapter 2. In this chapter, we present in a more specific way: the skeleton of this modular approach, the purposes of those modules and the data linking them. We illustrate the use of the modules with examples taken from publications in the literature. One will also find in this chapter links to all the codes and data we used in the chapters corresponding to the developed modules.

3.1 Design and Specification of the Modular Approach

As introduced in Chapter 1, the optimization work in BSS management strategies has interactions with estimation, prediction and verification works. From the literature review it appears that the majority of the authors propose a global approach to manage this interaction. The major limitation from those global approaches is, as presented in Chapter 2, the lack of comparability between the obtained results. Using this information, it seems obvious that it is hard to treat and maintain solutions for all of those problems in a global approach.

In our opinion, a modular approach could help to solve this comparison problem. Indeed, using a divide-and-conquer approach on complex problems have many advantages. It allows to study, solve, and manage each module independently from the others. This simplifies the re-use of the modules by making them easily available and understandable. This adds to the fact that the results of each module can be shared for future use without understanding the methods used. In this chapter, we describe the different modules along with the different kind of data used to link the modules.

In Figure 3.1, we represent the different modules and data used to design the modular approach. Each box represents a module either technological (like code or software) or scientific (a mathematical method, an algorithm etc.). The other objects represent the data used to link

the modules, they should be readily available, so that future researchers can use them without having a precise understanding of the work that leads to this result.

In this thesis, we present solutions to the modules represented with a wide square, and show an example of the links represented by the wide arrows. The wide objects are made available for re-use without need to understand the methods used. The colors of the arrows represent the type of data concerned by the link, see below for a detailed description. In the following section, we describe the main modules and their interactions with the other modules. In section 3.3, we highlight how the methodology is usually used in the publications by taking example from two publications.

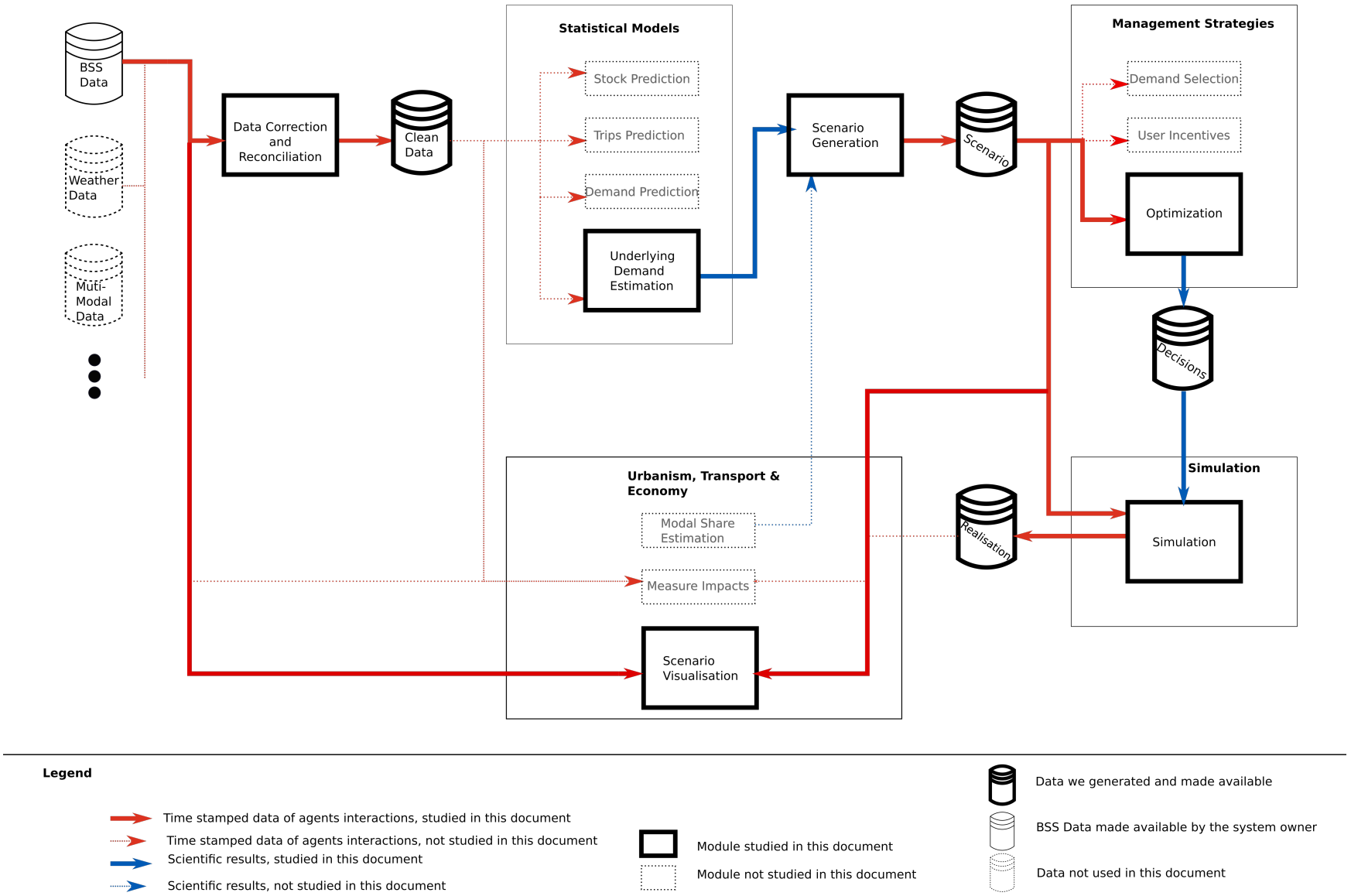
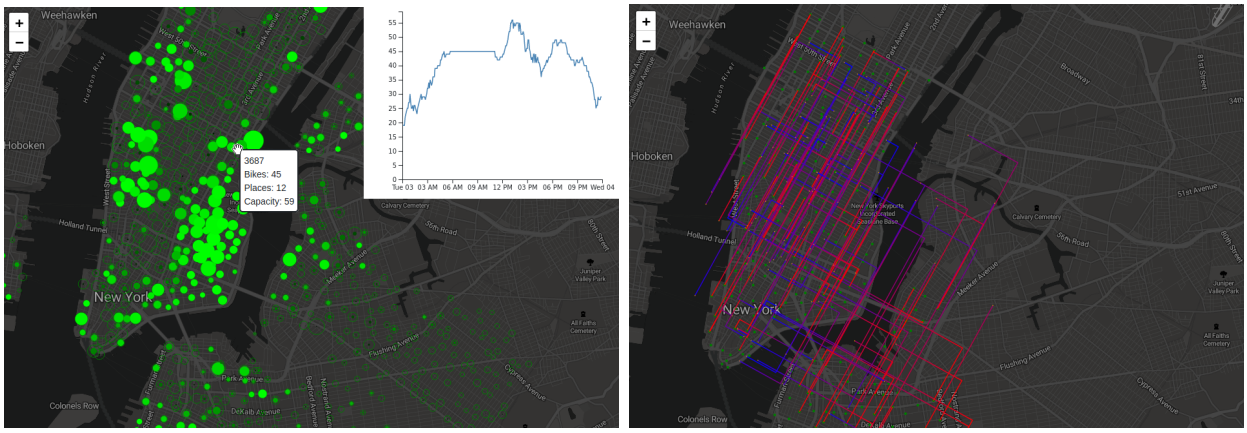
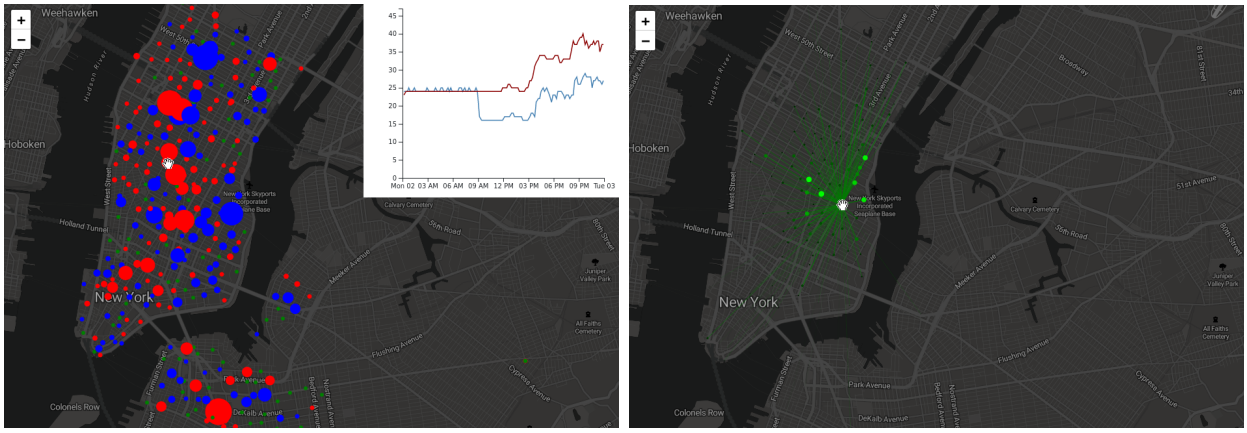


Figure 3.1: Visual representation of the interactions implied in the optimization of management strategies in BSS, the closely related topics are also represented.



(a) Stock data per station (b) User trip data

Figure 3.2: Examples of raw data visualisations



(a) Comparison of stock data and number of user trip (b) Trips repartition per station

Figure 3.3: Examples of aggregated data visualisations

3.2 Modules Description

Here, we present each of the modules represented by wide boxes in Figure 3.1.

3.2.1 BSS Data

These data represent the real-life BSS in the scientific studies. Data are gathered by the BSS and can be of various forms. Many BSS operators have already started to make those data available. It can be used, as is, to measure different indicators on the behavior of the BSS, but, in the majority of the cases, a cleaning process must be applied first.
We focus here on station-based BSS, where the users can rent a bike at a station and leave it in another station. In such systems, these data are often time stamped data that describe the interactions between the different objects in the system. For example, the bike stock data are the time stamped interaction between bikes and stations describing how many bikes are present in each station while, the user trip data represent the bike/user/station interactions by describing which user used which bike and at which station the bike was taken and left. In Figure 3.1, all the red arrows will represent time stamped data describing interactions between the objects in a BSS. A quick presentation of the data available has been presented in Chapter 2 in Section 2.2 while, a more specific description of the data considered in this thesis is presented in Chapter 4.

3.2.2 Data Cleaning Module and Cleaned Data

In many works, cleaned data are required since the raw BSS data present various type of errors and inconsistencies. It makes impossible to use them in a correct work. It is interesting to note that, while being fundamental to many studies, no community seems to take responsibility for this cleaning part of the process.
Once this cleaning process is done, the cleaned data should be directly available. To ease the re-using of those data, it seems natural to use the same format for the cleaned data and the raw data. Some authors from different scientific fields have nonetheless worked on historical data and, to this extend, proposed some methods of data handling, but the cleaned data are rarely available. A review on the solutions proposed for the management of historical data from bike sharing systems is presented in Section 2.2. In Chapter 4, we will detail the inconsistencies that can appear when working with historical data: in particular the bike stock data and the user trip data do not match, and the fact that the bike relocation operations are not available makes hard to reconciliate those two data sources. We propose a solution to increase the consistency of the data, by first inferring bike relocation operations from the historical data, and then conciliate the newly obtained data with the bike stock data. All the code and results data are available here.[1]

[1] https://gitlab.com/BarzoThom/datacorrection.git

3.2.3 Modal Share Estimation Module

As said in Section 2.3 of Chapter 2, having an estimation of the modal share is fundamental to the optimization process. This modal share can be estimated in various way, mostly it will be done by researchers in transportation economy using sociological & economical data. Another way is to estimate the modal share from the cleaned historical data. The historical data contains the observed demand, but also the periods in which the demand has not been observed. By combining those two data it is possible to give an estimation of the number of demand censored by the lack of bikes (or docks).
In Chapter 2 Section 2.3, we have presented a literature review of different estimation and prediction works that are made from historical data and detail more specifically the ones which estimate the underlying demand. In Chapter 5, we detail the solution we propose on this question. The main technical and scientific challenge, apart from the design of estimation method, comes with the evaluation of the different methods, since the data to estimate is not present in the historical data. To overcome this problem, we generate synthetic data and evaluate the different methods (two from the literature, one we designed) on those data.

3.2.4 Scenarios Generation Module and Scenario Data

The previously presented module on Modal Share Estimation can be scientifically and technically complex to set up. To ease the communication between the modal share estimation module and the optimization module, we propose to transit through a scenario generation module. This module is used to produce and make available the scenario data that will used in the optimization module.
The scenarios describe for each user in the modal share, the origin and the destination of the user, along with the time of departure. Since the modal share is a random process, a set of scenarios can be needed to describe correctly the modal share, but the concept remains the same. Those scenarios should be easily available for further researchers to re-use, in order to compare new optimization methods on the same "instances" of modal share realization. In some cases, the scenario generation module can be really simple, as generating data from a Poisson process, or more complex, as making a choice model for all populations described in the studied city. In Chapter 6, we use a simple scenario generation module in order to provide scenarios for the optimization module. The scenarios can be found on Zenodo[2] and the module on Gitlab[3]

3.2.5 Optimization Module

This module is the main module when doing optimization in BSS management strategies. A literature review on the methods has been presented in Chapter 2. In many cases the optimization process is a complex process. To offer a modular solution that can be linked to the other modules, we must define a way to share the results. We did not manage to find a

[2]https://doi.org/10.5281/zenodo.7195152
[3]https://gitlab.com/BarzoThom/redatner.git

conceptual format that would be able to describe all the solutions proposed in the literature. In some cases, the solution can be easily described, for example by the initial bike stocks at the beginning of the day. But in other cases, this process can be much harder to describe: we can think of the bike-angels operating in New-York, a group of users that relocate the bikes in order to win points and appear in the leader board. From a more general point of view, before defining a conceptual format, it would be required to make available the code and intermediary results used in the optimization process. In Chapter 6, we try to reproduce an existing work from the optimization literature to illustrate this module. The code used in the reproduction the analysis, and all the intermediary results can be found on Zenodo[4] and Gitlab[5] .

3.2.6 Simulation Module and Realization Data

Simulators are fundamental in order to assess the performance of an optimisation method. A more detailed explanation of this need is presented in 7. A simulator is a complex software, this requires an important software quality to produce a reusable and accurate simulator. This module must thus be treated separately from the others.

Also, since the simulator is one important way to check the performance of the optimization solutions, it should be independent of the optimization process to ensure a fairness on the evaluation. To finalize the consistency of the modules, the simulation should output the same type of results as the raw or cleaned data. This allow to easily interface the results of the full process with other studies such as the impact evaluation (that can be made directly from historical data or from simulation results). The consistent format between the simulation result, and the raw data can also be used to check the validity of the previously used modules. As example, we could verify the underlying demand estimation method by estimating from observed demand obtained through simulation. Or, we could optimize on simulated demand to (re-)improve the performances of the BSS. In Chapter 7, we present the different aspects such a simulator should implement. We also use a simulator to assess the performance of the solutions found in Chapter 6. The complexity of this task comes from the transmission of the results from the optimization results to the simulator. In Chapter 7, we overcomed this limitations by adapting the simulator to our needs in term of city geography. The simulator can be found in Github[6] and the code used for the analysis of the results on Gitlab[7].

3.2.7 Scenario Visualization Module

The visualization module aims at representing the data obtained from the different modules presented previously. Visualizing data is a very efficient way to:

[4]https://doi.org/10.5281/zenodo.7195152
[5]https://gitlab.com/BarzoThom/redatner.git
[6]https://github.com/tellae/starling.git
[7]https://gitlab.com/BarzoThom/starlinginventorylevel.git

- Grasp the dynamic of time-dependent data, as in Figure 3.2(a) where we present a snapshot of a dynamic visualization of the user trips.
- Perceive structure, as in Figure 3.3(b) where we represent for the selected station, the number of trips to the other stations, allowing to see the preferred destinations of users from a given station.
- Detect anomalies, as in Figure 3.3(a) where we represent the difference accross stations between the evolution of the bike stock and the number of trips. This shows that those data do not match.
- Observe patterns, as in Figure 3.2(a) where we see the number of bikes present in each station accross time. We can distinguish from this the usage patterns in function of the geographical areas.

The need for visualization at the different steps of the methodology, argues for a common format of data all along the methodology. Such a format will allow to develop few visualisations and reuse them in various contexts. This last part is important, designing efficient visualisation tools is a scientific and technical challenge. Therefore, the visualization module must be developed independently of the other modules and be easily available to the community. This availability will ease discussion and comparison. We did not propose here a review nor a comparison of the existing visualisations. Nonetheless, at the beginning of our research, we did not find easily available visualisation module. So, we developped our own (see Figure 3.2 and Figure 3.3 for examples). The code of those visualization is available on Gitlab[8].

3.3 Examples of Module Descriptions

From our analysis of the literature, we argue that, all the modules we propose are in fact present in a form or another in the existing literature. Neverthelesse, for the majority of the works in the literature, we only have access to a formal description of the global results and not to the different modules, neither as a software nor as a formal description. However, since the modules are used in the works of the literature, we think that proposing a modular approach constructed from those modules would allow to cross the different publications and enable comparison. Doing a proper demonstration of how the different modules are presented accross the literature would require too much time and is of little interest. In this section, we will rather illustrate this idea by taking sake citations from two publications: (Raviv et al., 2013) and (O'Mahony & Shmoys, 2015). In our thaught, those publications are among the publications of higher quality and are representative of the treatment of the modules accross the literature.

Data Cleaning Module
Concerning the use of real data in the two publications, we can read:

> *Our experiments are based on data collected from ten of the busiest stations in the "Capital Bike-share" program ((Raviv et al., 2013) p.17)*

[8]https://gitlab.inria.fr/tbarzola/stage_manuel.git

We tested the [...] approaches on real world instances gathered form actual system data ("Capital Bike-share") ((O'Mahony & Shmoys, 2015) p.693)

Since the data comes from real-world systems, from our experiences, those data present some errors and they must be cleaned before being used in the optimization process. We can therefore assume that, in both publications, the authors have developed algorithms to do this cleaning process. However, we do not have a precise description of those algorithms. It must be said that, in both publications, the authors propose to give access to the cleaned data upon request, we have not asked if the data is currently available and can not certify there are not. But, the authors of (Desuilbet et al., 2019) precise that data available upon request are rarely actually available.

Modal Share Estimation Module
In the chosen publication, we can read:

We note that the data is censored [...] we employed standard statistical method to overcome this difficulty ((Raviv et al., 2013) p.18)
To compute more accurate lower bounds on demand we need to take censoring of demand into account; to do this we mask the observed trip matrices, removing elements where the corresponding level element is at zero ((O'Mahony & Shmoys, 2015) p.690)

Both publications, by choosing to work on real-life data, must have used a model share estimation module. Nonetheless, in neither case we have access to a software implementing this module.

Simulation Module
Both publications verify their optimization results. In both cases, the authors chose to use the optimizer to verify the solution found (a more detailed discussion of this question can be found in 7).

In total 120 problem instances were tested [...] the calculation and the simulation was programmed in MathWorks Matlab ((Raviv et al., 2013) p.19)
We implemented the Integer Program in Gurobi [...] to generate the instance we took a series of system snapshots((O'Mahony & Shmoys, 2015) p.693)

In both publications, since the optimizer is used as verification model, we have an extensive description of the model. Nonetheless, in neither case, we could use the proposed verification model on another work without re-implementing the module.

3.4 Conclusion

In this chapter, we detailed the different scientific and technical challenges identified in Chapter 2. A modular approach, using a divide and conquer paradigm, could overcome the complexity of comparison of BSS management strategies, induced by the global approaches proposed in the literature. Such an approach is also efficient to propose solutions that are easily maintained and shared accross the scientific communities. From the literature review

presented in Chapter 2, we identified 5 modules used in every publication. Here, we explicited those different modules along with their interactions, and integrated them in a modular approach. For each one of those modules, we are able to describe specifically the workflow needed. We also specify the behavior of each module independently and the data that should be used as input and generated as ouput. In the remaining work, from Chapter 4 to Chapter 7, we propose an implementation of each module and, as specified in this chapter, make available the codes and data needed for future use.

Chapter 4

Data Handling and Reconciliation

In this chapter we explore the user trips data and bike stocks data. These two main data sources are often available for bike sharing systems, in particular for New-York city BSS (citibike). However, two problems occure when working with these data: First, the data sources are not consistent. Second the bike stock data are not accurate (there are too long intervals between bike stocks records in peak hours). The reconciliation of these two data sources is not an easy task, mostly because the user trips data do not contain the bike transfers due to relocation operations made in the system. We present a method to detect those relocations and to reconciliate the user trips data and the inferred relocations with the bike stocks data. This method adds the information from the user trips, into the new inferred bike stocks data obtained, this allows to have more accurate bike stocks data than the original.

4.1 Introduction

When studying bike sharing systems, it is fundamental to work on real-life data. Indeed, many factors impacting bike sharing systems are hard to model and it is impossible to model them all at the same time. Data from current bike sharing systems allow researchers to catch a glimpse of the complexity of the different factors affecting a bike sharing system. With the open-data trend, many operators propose to access to historical user trip data.Those are anonymized data corresponding to trips made in the system. The user trips are highly impacted by the number of bikes present in the stations, and the user trips data do not give access to the number of bikes in station. In 2014 the North American Bikeshare Association (NABSA[1]) defined a specification for real time data feeds of bike sharing systems (Sean, 2020). It was made to allow citizens, application developers or technology vendors to easily use the data. One of those feeds is the bike stocks data, where the number of bikes in stations are registered regularly. However, those records have many errors that cannot be easily detected nor corrected. Besides, the large granularity (time duration between two records) of the records makes that some variations in the stocks are missed because they occur in between two

[1] https://nabsa.net/

records. As detailed in chapter 3, the user trip data and the bike stock data, are the minimal data needed to do the comparison of BSS management strategies optimization methods. As presented in chapter 5 those data allow for a minimal working version of the underlying demand estimation version of the module.

Those two kinds of data, user trips and bike stocks, are two visions of a same reality, and can thus be used jointly. In this chapter, we try to reconcile these two sources of data in order to compensate for the lack of each data source individually. Adding together the bike stocks and the user trips data allows to access the number of bikes available in each station while reducing errors and decrease the time granularity of the bike stock records. These data improvements increase the fidelity of the data regarding the real events of the bike sharing systems. To present our methods, we use the data from the New-York city BSS (citibike), the system is a station-based system serving New-York city, Jersey City and Hoboken. The system is owned by Lyft who propose to it's user to rent bikes accross the city.
The rest of the chapter is organized as follows: In Section 4.2, we do a descriptive analyses of the two kinds of data independently. In Section 4.3, we describe how to compare the vision proposed by the two types of data and detail the errors obtained when comparing those two visions. Then, in section 4.4, we present the method to detect relocation and quantify it's performance. Finally, in Section 4.5 we recreate the new bike stock data by matching the user trip data and the bike stock data.

4.2 Data Description

In the following section, we describe the data from citibike that we collected from two different data-sources. The user trips data from the historical data made available by the operator, and the bike stock data collected by (Come, 2022). The data starts at June 2013 and is still available today (June 2022). To describe our algorithms we only focus on three months of 2016 as example.

4.2.1 User Trips Data

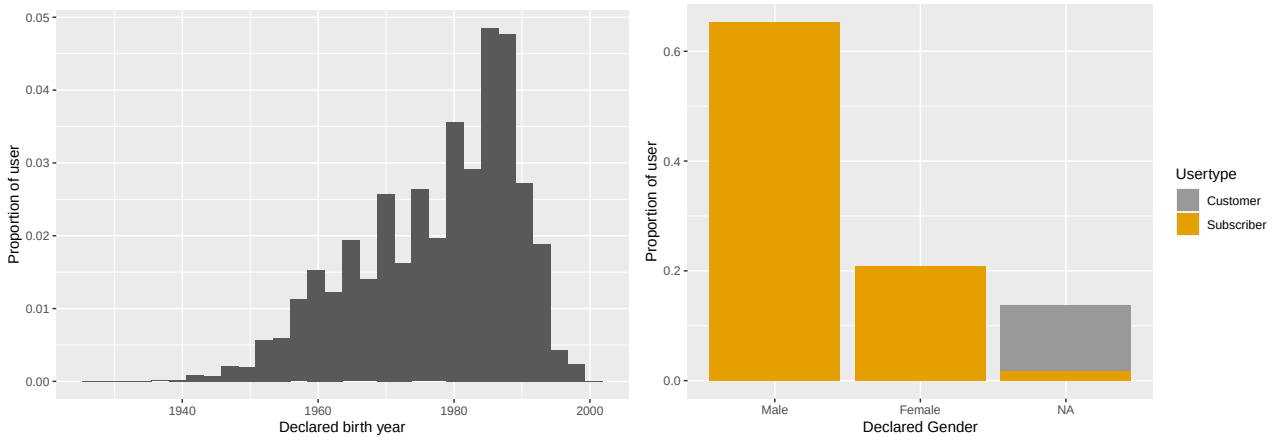
The user trips data correspond to all the trips recorded in the system. Nonetheless, the system operator made some processing on these data: all the bike movements corresponding to maintenance operations have been removed from the records. The abnormally shorts trips (less than a minute), have also been removed since the operator consider those trips as false starts or bike re-dockings.

The user trips data consist of 15 features:

- 5 trip related features
  - `tripduration` the time between the start time and the arrival time, in seconds;
  - `starttime` the time when the bike has been taken;
  - `stoptime` the time when the bike has been left;
  - `start station id` the identifier for the departure station;

  - `end station id` the identifier for the arrival station.
- 6 station related features
  - `start station name` the name of the departure station;
  - `end station name` the name of the arrival station;
  - `start station longitude` the longitude of the departure station;
  - `end station longitude` the longitude of the arrival station;
  - `start station latitude` the latitude of the departure station;
  - `end station latitude` the latitude of the arrival station.
- 3 user related features
  - `usertype` detail of the contract of the user, `subscriber` for user with a subscription and `customer` for the other users;
  - `birth year` birth year of the `subscriber` if available;
  - `gender` gender of the `subscriber` if available;
- 1 bike related feature
  - `bikeid` the identifier of the bike used for the trip.

We can observe some descriptive statistics on the data we have. On the three months of data we observe 3685747 user trips, accross 503 stations. All accross the data, the required fields are always filled. This is expected since the data are cleaned by the system operator. The declared year of birth and the gender are the two only fields not mandatory for customers. Without surprise the majority of the trips are made by subscribers (87.86%). As found by (Winters et al., 2019) for the considered months (and we can expect a similar behavior for the others months), subscribers are in mostly young males as we can see in Figure 4.1.



(a) Distribution of users by declared birth year, Distribution of users by declared gender

(b) Distribution of users by declared birth year, Distribution of users by declared gender

Figure 4.1: Description of the users

When looking at the trip duration (Figure 4.2), we already observe a preference of the users regarding this criterion. Due to the fees applied to users for keeping bikes more than a certain time (45 minutes for subscribers and 30 mins for customers), we clearly see the drop in the distribution of subscriber trip duration. To construct Figure 4.2, we removed the trips lasting more than two hours. In the trip duration we see some outliers (maximum at 42.28 days for

the subscriber and 39.66 days for the customers), we choose not to remove those outliers since we cannot decide which of the start time or stop time is incorrect and which is correct.

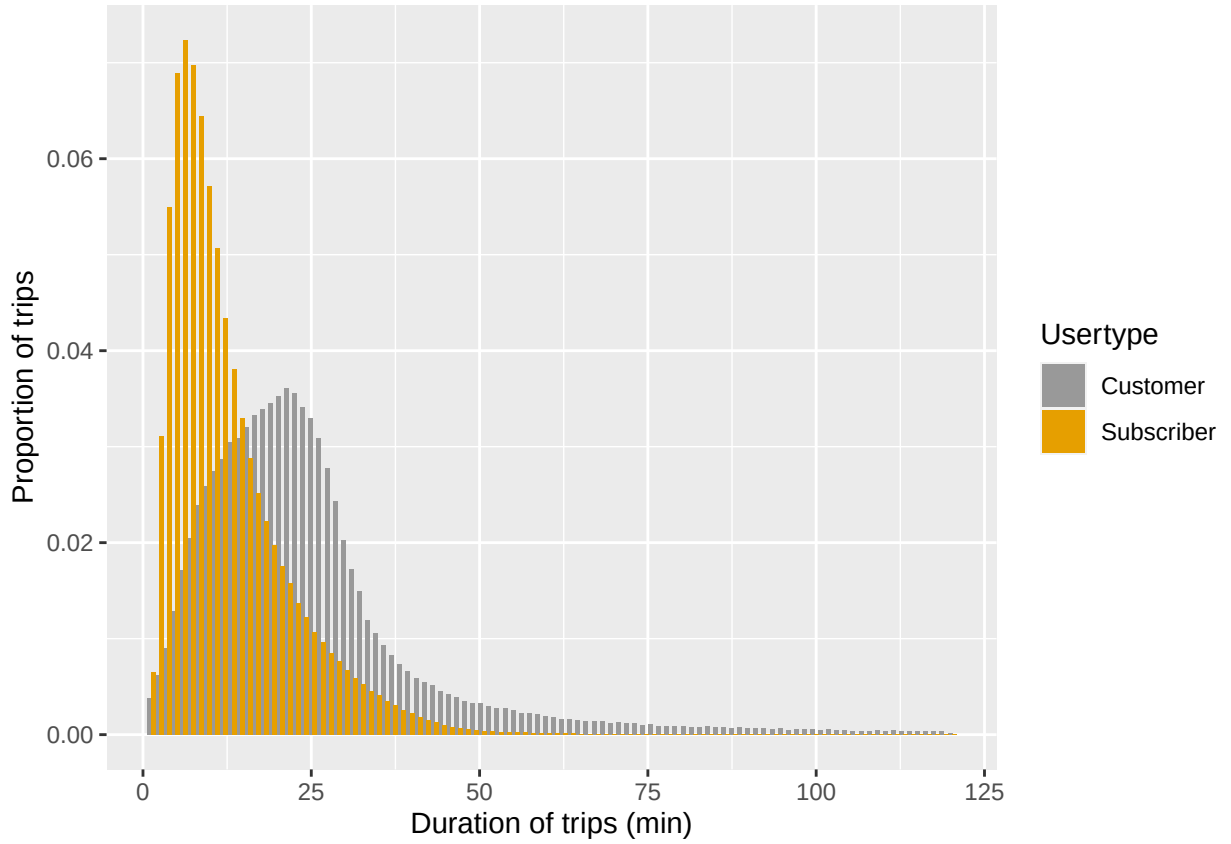


Figure 4.2: Distribution of the tripduration for three months of data, duration is ploted only if it is below two hours

It can be concluded than there is few missing data, but for most of the features of the files, we do not know how to assess the quality of the given data. Since the data have been pre-processed by the BSS operator, we can think of this data source as trustworthy. In the rest of the work we will assume that the user trips data are complete and without recording errors.

4.2.2 Bike Stocks Data

According to the GBFS (Sean, 2020), Citibike operator (Lyft) provides the bike stocks data at least every five minutes. These data have been scraped from their website and stored by various researchers, we use the data from (Come, 2022) who store the records every 20 min. However, when opening the files for further analysis, we have multiple records that do not follow the standard of the GBFS (Sean, 2020). This is the reason why we started by automating the correction of errors on the files. The errors we found were categorized in four categories that were corrected separately:

Table 4.1: Number of error by category for the different files

Error categoy	Month of Data		
	April 2016	May 2016	June 2016
1	3409	3550	0
2	2	1	0
3	5	0	0
4	1	0	0

Table 4.2: Number of missing data for each feature of the stock data

Feature Name	Nb Missing Data
available_bike_stands	0
available_bikes	1
bike_stands	0
contract_name	0
download_date	8
number	0
status	0

- Category 1: File format error, those errors make impossible to read the concerned line in the file;
- Category 2: Column name error, some columns are not registered with a valid name;
- Category 3: Value name error, some values are not correctly registered with a valid name;
- Category 4: Manually corrected error, all the other errors that were fixed by hand.

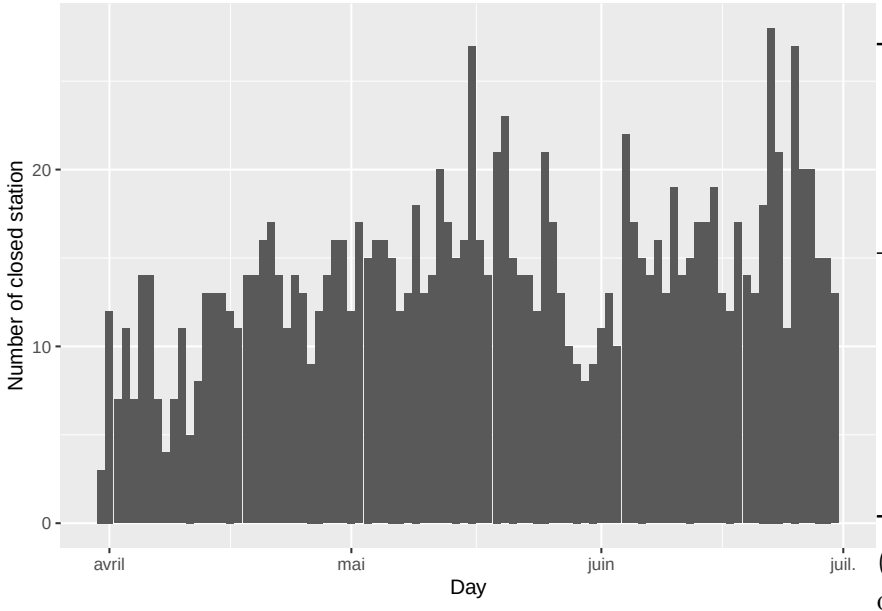
Once this phase has be processed, we obtain the bike stocks data with 7 features:

- `download_date` the time when the recording was made;
- `number` the identifier of the station where the recording was made;
- `available_bike_stands` the number of empty bikes stands in station at the time of recording;
- `available_bikes` the number of bikes present in the station at the time of recording;
- `bike_stands` the total number of bikes parked (empty or not) at the station at the time of recording;
- `status` the operating status of the station (in operation or out-of-order);
- `contract_name` the city where the stations operate.

We start by counting the number of missing data, in table 4.2 we see that most of the records are present.

The quality of the data can also be seen on the number of valid observations. On the features obtained we know whether a given station is operating (Open) or not (Closed), this gives an

indication of whether the observations are valid or not. We observe in Figure 4.3(a) that the number of closed stations is relatively constant during the time period. This is quite positive because it means that we do not have period without any data on the system, but it also means that there are no perfect days when all the stations are operating properly.



(a) Number of closed station per day

Station Id	number of closed inter-vals	number of open inter-vals
403	6615	0
3248	4336	817
3230	4387	2227
375	428	40
3253	164	26
3252	116	22

(b) Stations that are more often closed than open

Figure 4.3: Evolution of the number of closed station

On the stations that are more often closed than open, in Figure 4.3(b) we see that only one of the station is always closed, and only 6 are more often closed than open on a total of 520 stations.

The data quality can also be seen through the time between records, in our data the records should occur every 20 minutes. When observing the repartition of the times between records we observe that less than 1% of the records are spaced with more than 22 minutes. This could indicate a good data quality.

Finally we can try to quantify the consistency of the data. In the system, a given dock is either paired to a bike or empty. Thus, the number of bikes added to the number of empty docks should be equal to the total capacity of the station. In Figure 4.4, for each station we represent the average difference between, the sum of the number of available bikes plus the number of available docks, and the total capacity of the station as a cumulative distribution. We see that all the average difference are positives: the capacity is always greater than the number of bikes and free docks. We also observe that this difference in scarcely equal to zero, meaning that for the majority of stations there is at least one record that is not correct.

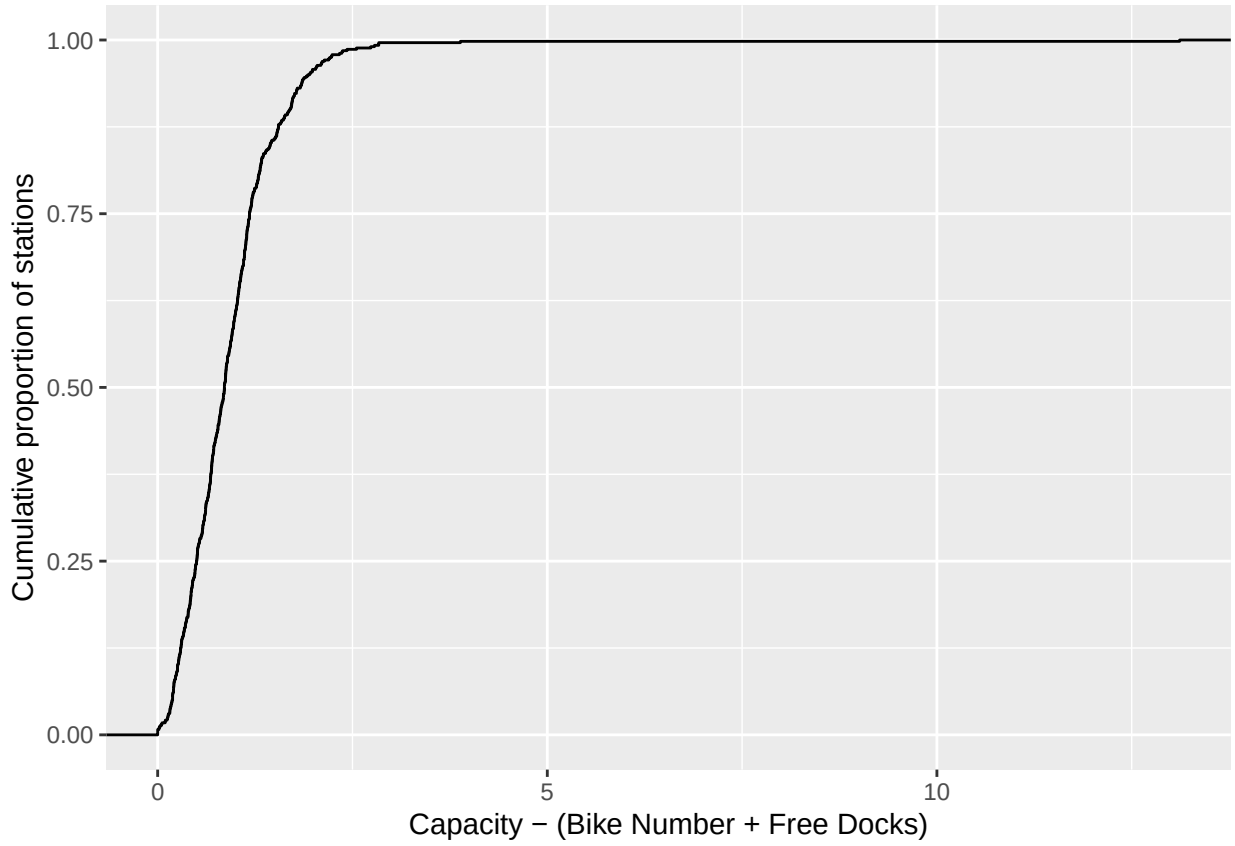


Figure 4.4: Distribution of the average difference between the capacity and the number of bikes and empty docks per station

4.3 Inconsistencies Between Bike Stocks and User Trips Data

4.3.1 Number of Stations

We can compare the number of stations present in the user trips data (as departure station and as arrival station) to the number of those in the bike stocks.

In the user trips data, we have 503 stations while in the bike stocks data we observe 520 stations. It means that there are stations for which we have bike stocks data but that do not register any trip. Furthermore, we have 25 stations that are in the bike stocks data but not in the user trips data and reversely, 8 stations are in the user trips data but not in the bike stocks data. Even if it would be surprising, we can not exclude that some stations did not receive any trips during the three selected months. Nonetheless, the fact that stations recording user trips with no available bike stocks data is definitely an error in the recording. To have an idea in the number of wrong records we can say that there are 665 user trips from (or to) stations where the bike stocks data is not available. Only one trip is made between two stations where the stock data is not available.

4.3.2 Relative Bike Stocks and Relative Stock Balance

By observing the user trips incoming and outgoing at a given station we can compute the *relative stock* of this station. We construct the relative stock by taking the current stock as initial stock and iterating on the user trips, for each incoming trip we add one to the relative stock, and we subtract one for each outgoing trip. In Figure 4.5, we present the evolution of the relative stock for station 72 over the whole month of May.

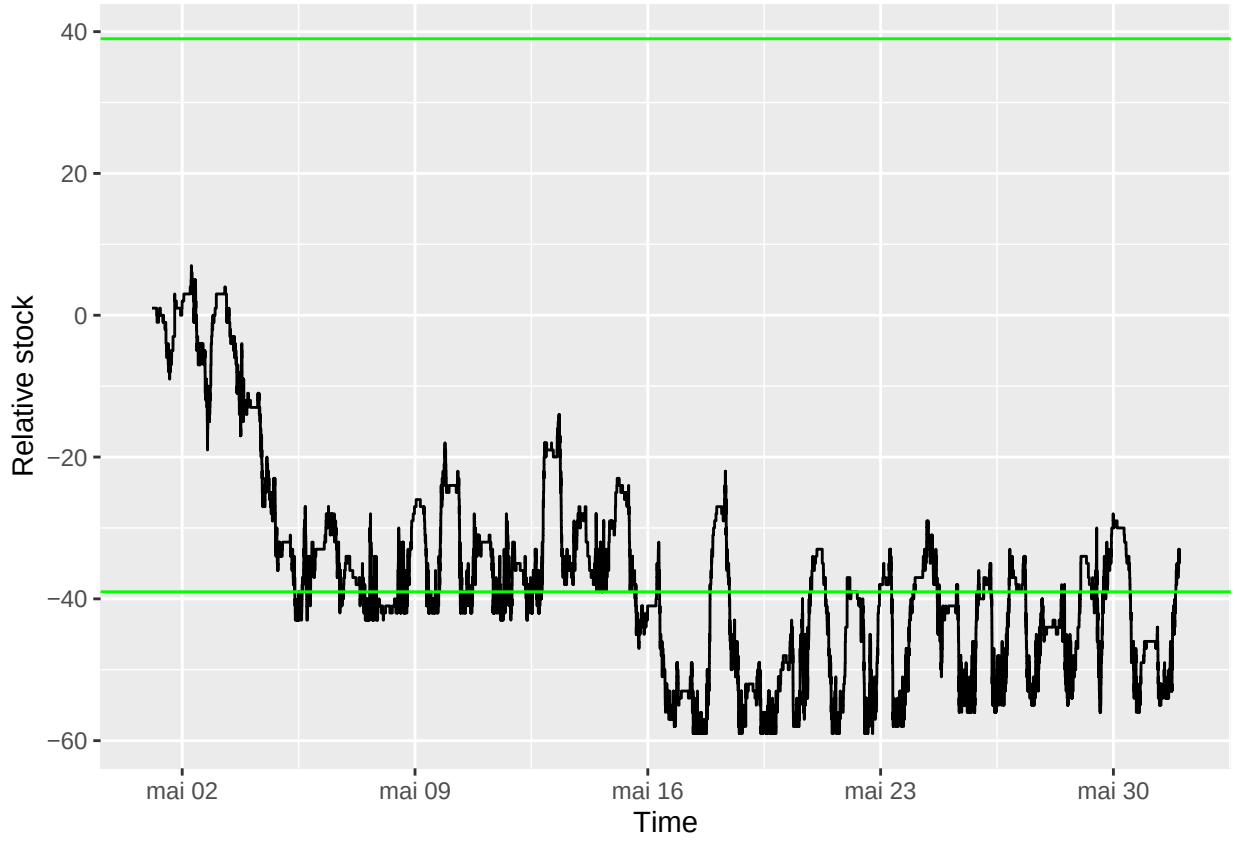


Figure 4.5: Evolution of the relative stock of the station 72 for month of may along with the capacity of the stations

For a given time interval and a given station, we can compute the *relative stock balance* of the station over the time interval. We make the difference between the relative stock at the beginning of the time interval and at the end of the time interval. For instance, for station 72 in May, the month balance is -34. If the relative stock balance is negative, it means that more bikes left the station than bikes arrived. Thus the initial number of bikes placed (which is below the capacity of the station) in the station should be greater than the absolute value of the balance. For station 72, it means that we should have at least 34 bikes at the beginning of May. Symmetrically, if the relative stock balance is positive there are more incoming bikes than outgoing bikes, the station should be able to dock those bikes. Furthemore the station can dock a number of bikes until its capacity is reached. Therefore, we have the following property regarding the relative stock balance.

Property: For all time intervals the absolute value of the relative stock balance should always be less than the station capacity.

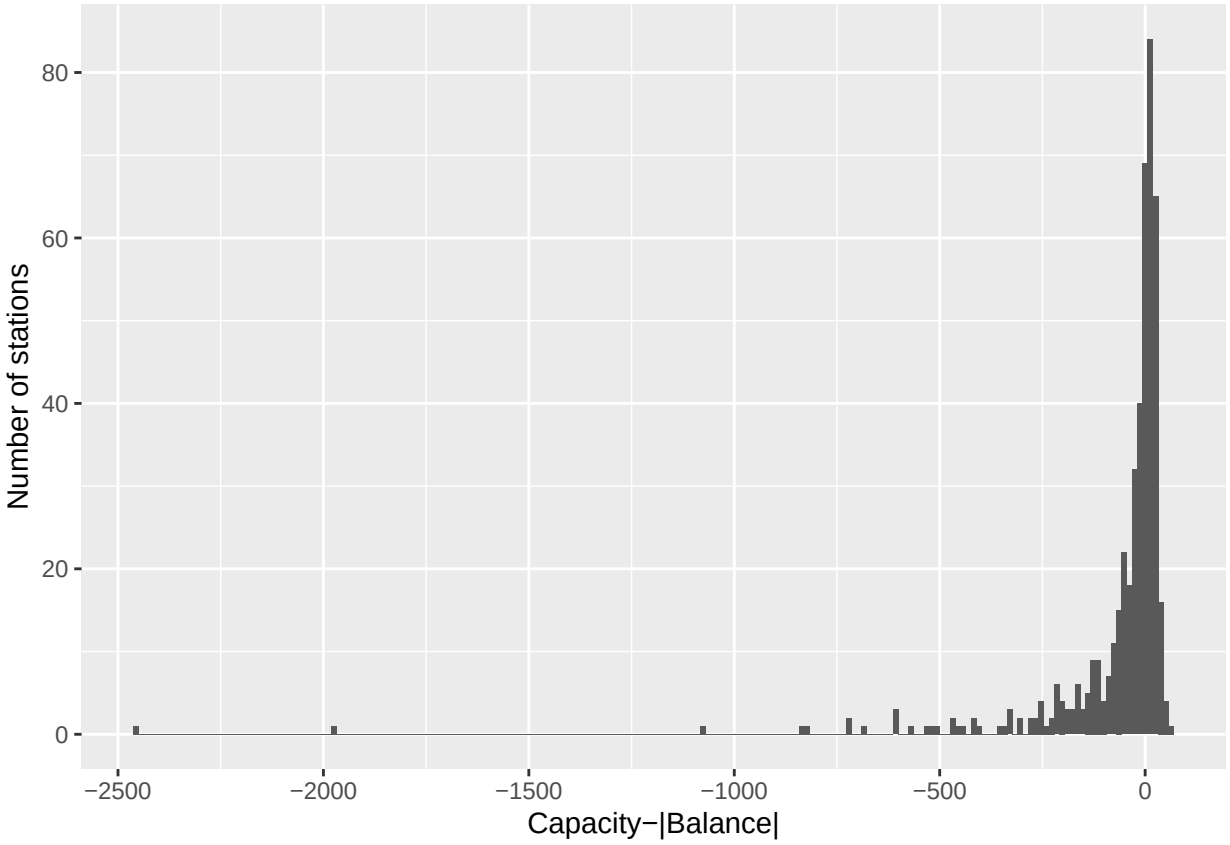


Figure 4.6: Difference between the capacity and the balance for each station

In Figure 4.6, for each station we compare its average capacity to its absolute relative stock balance. Note that we use average capacity instead of capacity, since in the data the station capacity is not constant (small variations, probably errors or broken docks). To be consistent with the property given previously, we should observe positive differences for all the stations. However, the majority of the stations are in the negative part of the figure. It means that the absolute values of the balances exceed the capacities. This indicates that we are missing some bike tranfers that would correct the relative stock balances to keep them below their respective capacity.

In the rest of the chapter, we will complete the user trips data with all the bike relocations and then reconciliate it with the bike stocks data in order to have fewer errors and more accuracy in the bike stocks data.

4.4 Enriching Data With Bike Relocations

As presented in Section 4.3, the bike transfers obtained from the user trips and are not consistent with the given bike stocks in stations. This is expected since bike relocations are

made to reduce imbalances between stations. We destinguish two kinds of imbalances. The *tide* is when users take (resp. leave) bikes in the morning and leave (resp. take) them in the evening, this leaves the stations with a lack (resp. surplus) of bikes in the morning while the opposite occurs in the afternoon. The second type of imbalance is the *gravity*; in some stations the users take (resp. return) bikes without returning (resp. taking) them (users will rather bike downhill than uphill). Those bike relocations, for disclosure reasons, are not available. To reconciliate the trips data and the bike stocks data we expand the user trips data with those *inferred relocations*. To infer those relocations, we start by identifying the relocated bikes with inconsistent paths, then we try to schedule the relocation operations in a feasible way.

4.4.1 Bike Path Reconstruction Method

We use the method detailed in (Kranish, 2021) to find "ghost rides" (inferred relocations in our terms). The method can be descibed as follows: Since we have access to the `bikeid` of the bikes used for each trips. For each bike we can build a path between stations. If we detect some inconsistencies in this path, we can assume that this bike has been relocated between stations. In Figure 4.7 we present an example where for bike 1 the path between stations is not consistent. We have a user trip from station 3 to station 1 between t_1 and t_2, and another from station 2 to station 3 between t_3 and t_4. Adding the inferred bike relocations allows to rebuild a consistent path for bike 1.

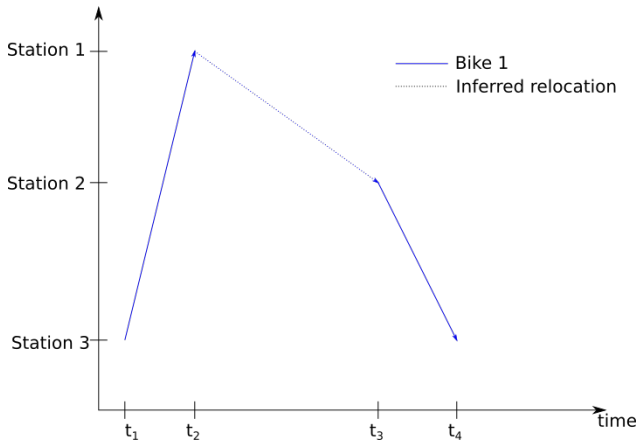


Figure 4.7: Space-time graph with 2 trips and 1 inferred relocations

We will describe the inferred relocations through 5 features:

- `start station id` the identifier for the departure station,
- `end station id` the identifier for the arrival station,
- `release time` the last time the bike was seen at the departure station,
- `deadline` : the first time the bike was seen at the arrival station,
- `bikeid` the identifier of the bike concerned by the inferred relocation.

In the example presented in figure 4.7, the relocation of `bikeid` 1 found has as `start station id` the station 1, as `end station id` the station 2, as `release time` t_2 and as `deadline` t_3.

The time interval between the release time and the deadline will be called the *Relocation Time Window (RTW)* of the inferred relocation. In practice, the bike must be relocated in this time window.

However, we cannot easily compute the exact time when this relocation occurs. In section 4.6.3 we propose an formalized formulation of the problem of finding the exact relocation times. In the rest of this work, we often use the term "inferred relocation" for both inferred relocations and vanishings. We will not use much of the vanishing information and vanishing is just a particular case of relocation without known arrival station.

Because the user trip data is provided on a monthly basis, we decided to detect the relocations for the whole month of May 2016. This raises the issue of the time horizon boundaries in our path reconstriction method, an issue that has not been addressed in (Kranish, 2021). The inferred relocations can only be found in between two user trips. Thus, at the end of the time horizon, we may miss some inferred relocations since the second user trip needed could not be present in the time horizon. It is thus necessary to take as time horizon a longer period than the period studied (1 month). After experimenting on different periods, we estimated than, using one month before and after the studied period, allows to compromise between the number of relocations found and the computation time. This period of one month is explained by the fact that a bike idle for more than one month in a station can be assumed *vanished*, i.e. relocated without precise destination. Following this assumption, in order to find inferred relocations on a single month (in our case May 2016), we work over a time horizon of three months (April to June 2016). We keep track of all inferred relocations with at least one ending in the month in between.

4.4.2 General Characteritics of the Inferred Relocations (periodicity, tide, balance)

In this section we propose to describe some characteristics of the inferred relocations. We try to give a general idea on those data in particular how the relocations are distributed in time and space accross the month of May 2016.

With this method applied to May 2016 of the New York City data instance, we found a total of 45,080 inferred relocations for 1,212,280 trips. They can be split into 44,904 inferred relocations and 176 vanishings. This result is not surprising since, as we said previously, a bike idle for more than a month is an exceptional event. We start by observing the number of potential relocations that can happen at time t (i.e. with a RTW including t). In Figure 4.8 we represent this number as a function of time.

We have an average of 1461 inferred relocations during the month. A daily periodicity can then be observed, but this is not surprising since we can assume that many of the inferred relocations are made periodically. Indeed, bikings are observed during the day, leading to RTW over one or more nights. We note also that we start the month with 1911 inferred relocations. This is explained by the fact that some inferred relocations start out of the concerned month but will end inside.

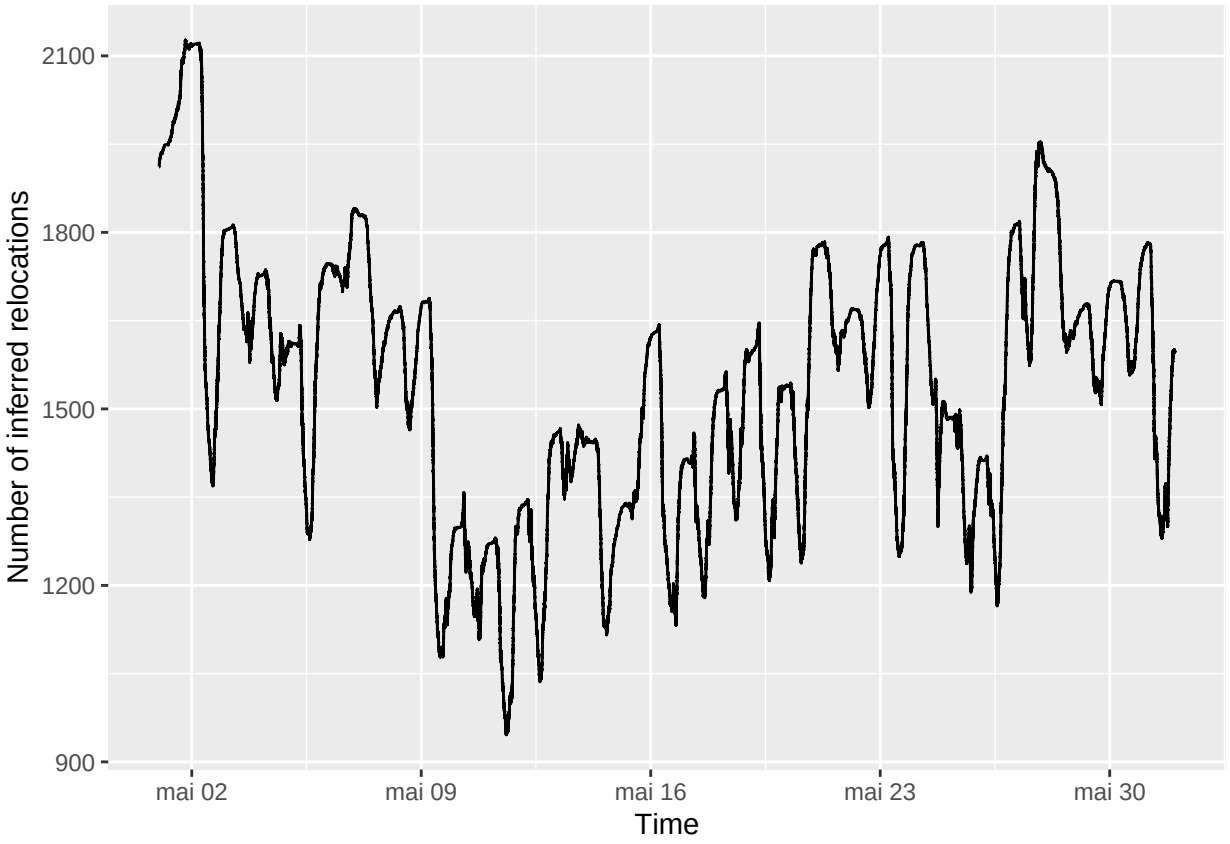


Figure 4.8: Number of potential relocation in function of time for May 2016

As mentioned previously, it is useful to observe the distribution of the size of the relocation time windows (see Figure 4.9). We observe that, a great proportion of the inferred relocations have a time window of less than two days, but some inferred relocations have a greater time window. This can be explained by various factors. As said previously, the RTW found for the inferred relocation is highly impacted by the usage of the bike. Indeed, if a bike is left in a station with few trips, then it will wait for a long time before being taken again leading to a longer RTW. Furthermore, the relocation are often made during the night thus, by working on windows of 12 hours, we could group the inferred relocations in coherent group that might happen at the same time.

Another interesting information to observe is how the inferred relocations are distributed across the stations. In Figure 4.10(a), the stations are ordered according to the number of incoming inferred relocations and outgoing inferred relocations.

From Figure 4.10(a) we can separate the station by the relocation pattern. The first group consists of the few stations with many inferred relocations both outgoing and incoming. This pattern is explained by the *tide* phenomenon: To reduce this phenomenon, bikes are relocated multiple times in the day and with both incoming and outgoing inferred relocations. The second group is the stations with a huge majority of one type of inferred relocations, either incoming or outgoing. This inferred relocation pattern can be view as a way to reduce the *gravity* phenomenon. All the other stations with few inferred relocations constitute the third

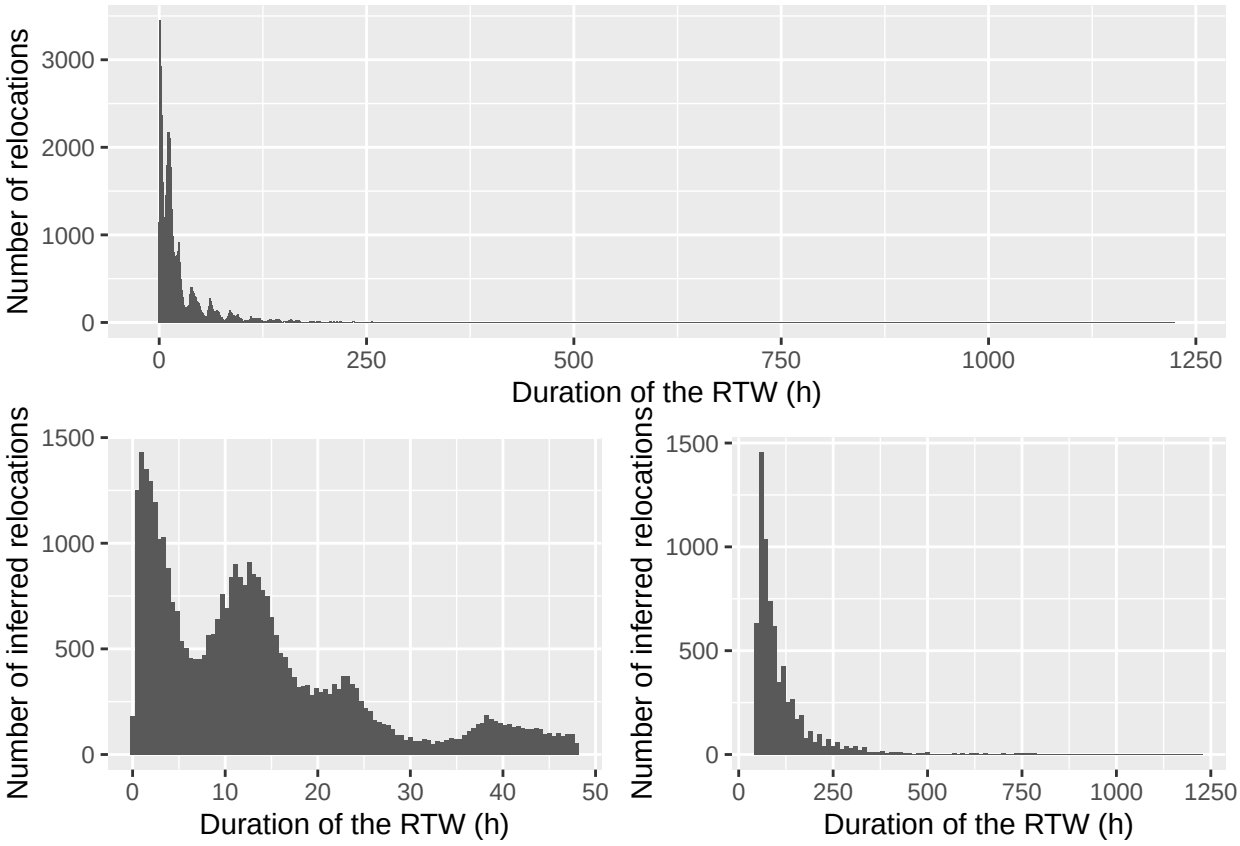


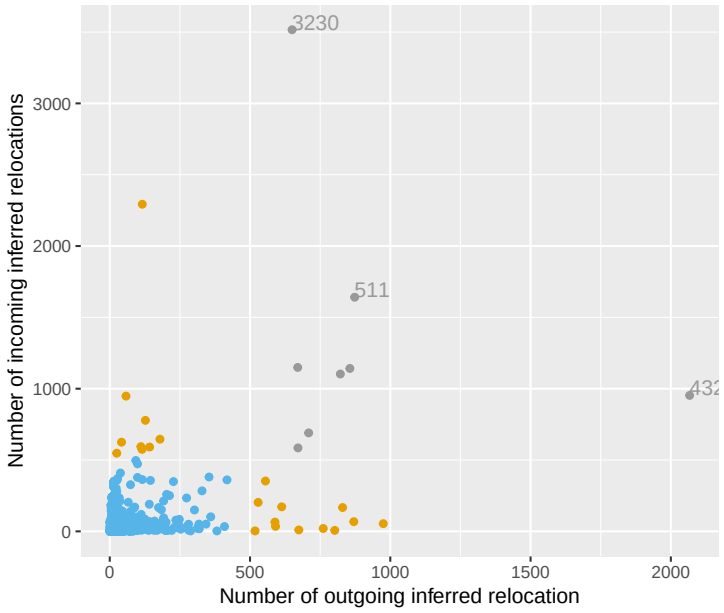
Figure 4.9: Distribution of duration of the RTW, above are all the inferred relocations, bottom left with tripduration less than 2 days, bottom right with tripduation more than two days

group.

In Figure 4.10(b) we zoom in on the 7 stations with the highest number of inferred relocations. We see easily the concentration of inferred relocations in a few stations: the first station has twice as many inferred relocations as the fifth.

We can observe in Figure 4.11 the cumulative number of bikes relocated at each time instance t for the top three stations. The curves show how the pattern of inferred relocations evolves for each type of station:

- Station 511 has a small gravity and a small tide. We observe a clear wavelet pattern of inferred relocations with a daily periodicity in order to compensate the tide during weekdays, but also a clear global inceasing pattern over the whole month with more incoming inferred relocations than outgoing ones to compensate the gravity. We also note a periodicity due to weekdays and weekends (five wavelets followed by a 2-day plateau), where the user needs are different.
- For stations 3230 and 432, the gravity is much bigger, but the periodicity in the inferred relocations shows the stations tide. Again, the usage of those stations is different between weekdays and weekends.



(a) Number of incoming relocations as function of the number of outgoing inferred, each dot represent a station

station id	number of outgoing inferred relocations	number of incoming inferred relocations
3230	650	3516
432	2067	953
511	873	1641
3236	116	2293
304	856	1142
487	822	1103
520	670	1149

(b) Station with the most inferred relocations in total

Figure 4.10: Number of inferred relocations per station

4.4.3 Assessing the Number of Inferred Relocations Found

We need to assess whether the number of inferred relocation found is accurate or not, assuming that the given station capacities and user trips are correctly recorded. To do this, for each station we compare its capacity to its new absolute relative stock balance obtained while taking into account both user trips and inferred relocations. Figure 4.12(a) represents the repartition of this comparison over the whole month of May 2016.

We note in this figure, almost all the stations have a consistent relative stock balance. Only four stations have an absolute balance over their capacity (negative difference). We can consider that, for most of the stations, we have been able to detect most of the inferred relocations.

We know that sometimes, Citibike used ‘’valets stations” (“Valet Stations,” 2022) in order to temporarily extend the capacity of a station. We can thus expect that these four stations are stations where valets are operating, leading to a miscount of the bikes available in station and a wrong station capacity. But for the moment, we do not have a clear explanation of which kind of inferred relocations are missing for these four stations.

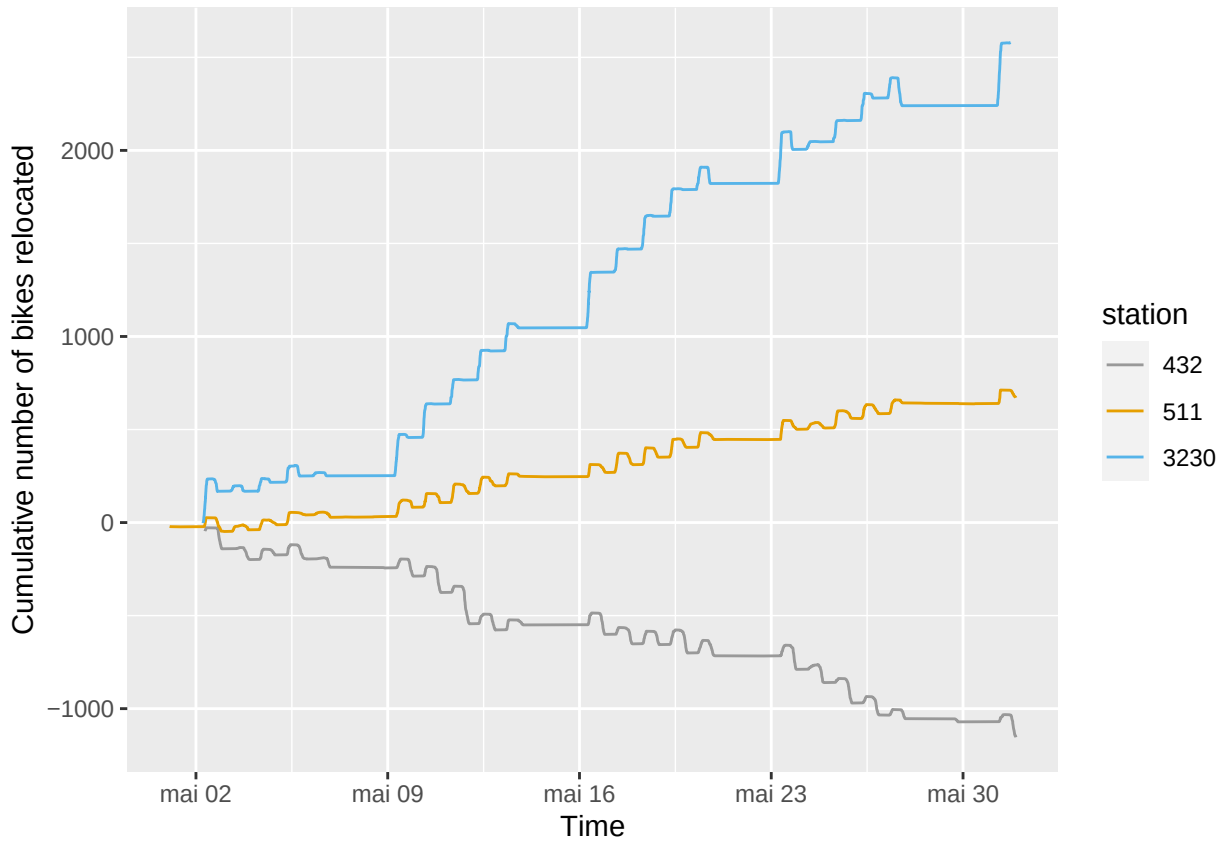


Figure 4.11: Evolution of the number of bike relocated as function of time for three stations

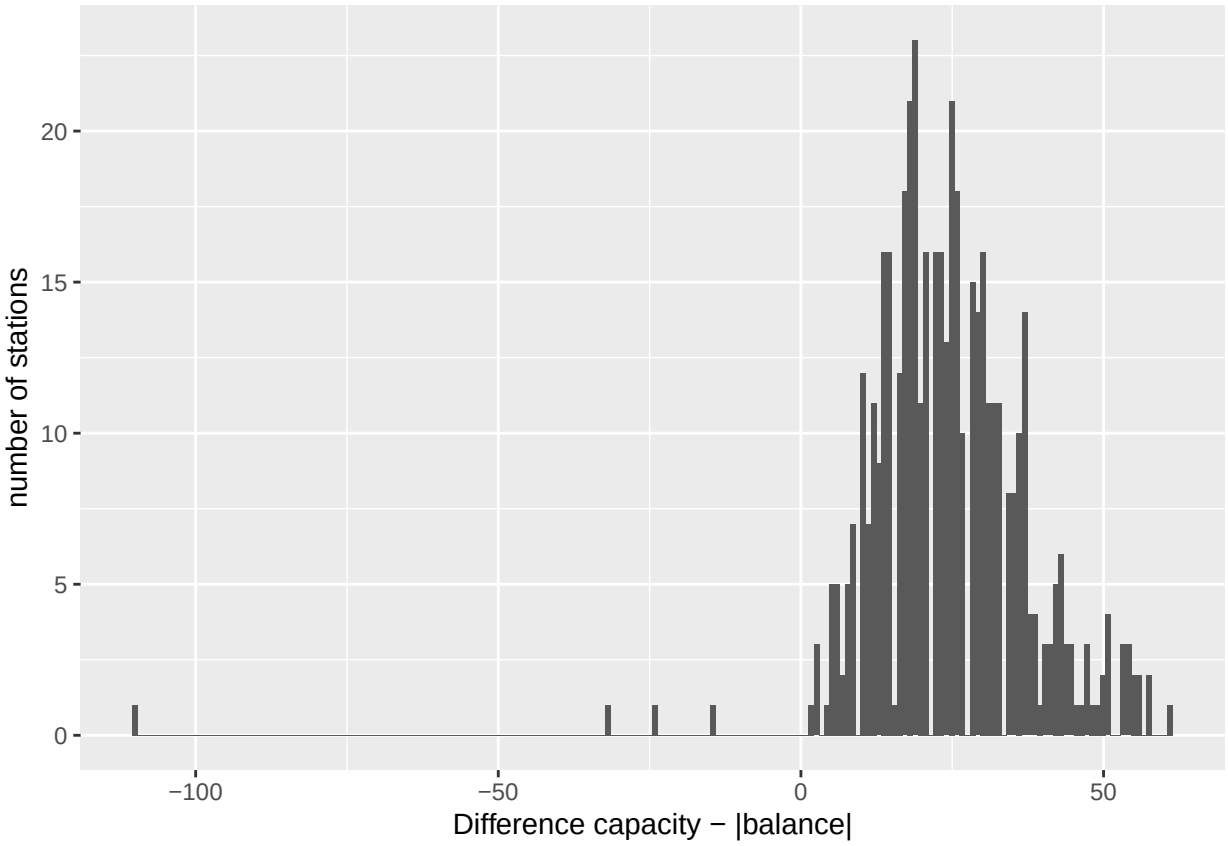


Figure 4.12: Repartition of difference capacity-|balance| accross stations

4.5 Matching Inferred Relocations and Bike Stocks Data

4.5.1 Constructive Alogrithm

The final stage of our work on data processing is to *reconciliate* the bike stocks with the users trips by inferred relocations to have better quality data. To do this reconciliation, we use the user trip data and the inferred relocations to build a new bike stock data. We call this new stock the *Inferred bike stock*. We assume that, except for the missing inferred relocations, the user trips data are correct. We will use the user trips data and the inferred relocations found to build new bike stocks as close as possible with the observed bike stocks data. For each station, to build a consistent bike stock from the user trips and inferred relocations, we start by computing the relative stock. Matching the relative stock and the given bike stock data can be decomposed in two parts:

- Scheduling the inferred relocations over time to have a variation of the relative stock consistent the variation of the bike stock data.
- Determining the initial stock level in order to have the relative stock consistent with the bike stock data values.

Those two parts are related, but, since it would be easier to determine the initial stock level once the relocations are scheduled, we choose to start by scheduling the inferred relocation. Since the bikes are mostly relocated during the night and run by users during the day, we decide to work on group of 24h to schedule the inferred relocations from noon the current day to noon the next day. We will use a subset of data from station 72 as example to describe the idea of the algorithm we designed.

In Figure 4.13, we show the data from May 2nd at 12am to May 3rd at 12am. Over this time period, we have three types of inferred relocations:

- The 15 incoming inferred relocations whose relocations time windows overlap each other (group 1).
- The 4 outgoing inferred relocations whose relocations time windows overlap each other (group 2).
- The last outgoing inferred relocation crossing the end of the time period (group 3).

For the group of incoming (resp. outgoing) inferred relocations, we try to look for the largest gap of incoming (resp. outgoing) bikes in the given bike stock without there being any related user trips. Around May 2nd at 18pm, we can easily distinguish a gap of 14 incoming bikes. Furthermore, the time interval between the two records of the gap intersects all the relocation time windows of group 1. We consider 14 of the 15 inferred relocations of group 1 with the tightest time windows. We schedule them all between the two records of the gap. The unused inferred relocation, which has the largest relocation time window, is awaiting scheduling. In the same way, around May 2nd at 21pm, we can see a drop in the stock level corresponding to 5 outgoing bikes. The time interval between the two records of the drop, intersects all the relocation time windows of group 2. We apply the same process as for incoming bikes

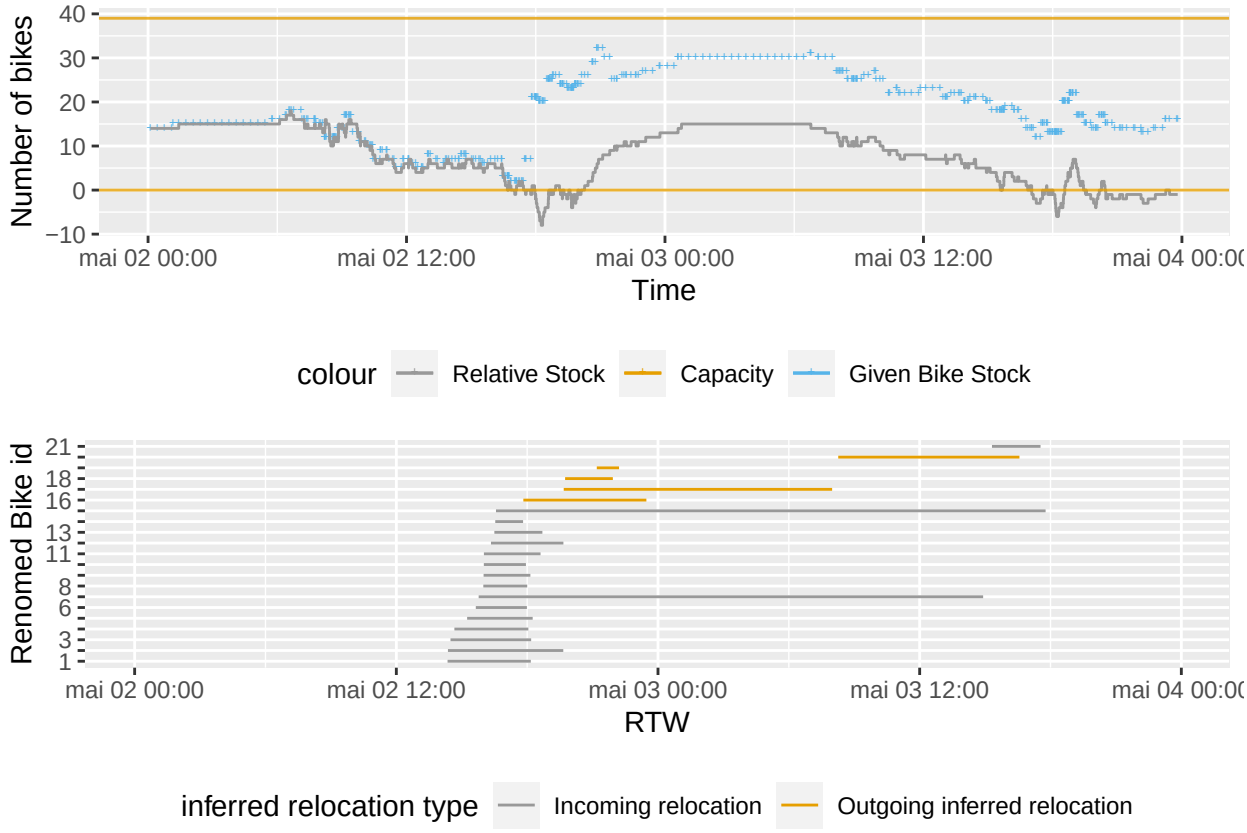


Figure 4.13: Relatives and given bike stocks of station 72 on two days, below are represented the RTW of the inferred relocations, the bikeid are modified to fit the figure

by sechduling the 4 outgoing inferred relocations between the two records of the drop. For the group 3, we are looking for a drop of one bike in the given bike stock level without there being any related user trip. Such a drop cannot be distinguished from a measurement error. However, since its relocation time window continues after the end of the time period, it will be considered with other outgoing inferred relocations if there is any in the next time period. For all relocations that has not already been scheduled (groups 1 and 3) we keep track of this inferred relocation as unused and they are available for other stations.

A more specific description of the algorithm is given in algorithm 1:

Now that the inferred relocations are scheduled, we can compute the relative bike stock for each station over the time horizon. But, to compute the inferred bike stock, we need to calculate the value of the stock at the beginning of the time horizon.
To do this, we compute the median of the differences between the levels of the given bike stock and the ones of the relative stock (starting at 0 at the beginning of the time interval). We then use this median as the initial stock level. An inferred bike stock over a time horizon is then obtained by starting the relative stock at the initial stock level.
In figure 4.14 we represent the result of the algorithm on the data presented as example (station 72 from May 2nd at 12am to May 3rd at 12am). The scheduled relocation operations are represented by arrows placed at the operation date, the upwards yellow arrows represent

Data: Given Bike Stock Data, User Trip Data, Inferred Relocations with RTW
Result: Scheduled Relocations
forall *Stations in the given bike stock Data* **do**
  forall *24 hours periods from 12am to 12am the next day* **do**
    Group the inferred relocations by type and overlapping RTW;
    Compute the gaps in the given bike stock data without related user trips.;
    forall *t time of the gap (by descending gap value)* **do**
      if *gap* > 3 **then**
        Find the group of relocations with an RTW intersecting t;
        Schedule a maximum number of relocations starting with the tightest RTW, to explain the drop;
        Keep the unscheduled relocations awaiting;
      end
    end
  end
  Delete all unscheduled relocations;
end

Algorithm 1: Inferred Relocation Scheduling Algorithm

the incomming relocations while the downwards green arrows represent outgoing relocations. We see, that three relocation operations are scheduled, two incomming around 6pm and one outgoing around 10pm. This scheduling solution give a good match between the inferred stock and the given stock.

4.6 Comparison of the Inferred and Given Bike Stocks

So far, we have various indicators to quantify the comparison between the inferred and given bike stocks, such as Mean Square Error or Mean Absolute Error. Those indicators allows to compare the given stock to the relative stock before scheduling the relocations and to the inferred stock. For instance, the example presented in 4.13 (before the matching) has 11.72 as MAE and 213.49 as MSE. After the matching, the result presented in 4.14 has 2.07 as MAE and 15.15 as MSE. However, these indicators provide an extremely limited insight on performance of the method. We already shown that the relocations were missing in the user trip data, thus adding those relocation (even scheduling them poorly) will improve the error measures. Those indicators can be used for comparison to other methods or, as an objective in a formalized problem such as presented in Section 4.6.3. In order to assess the reconciliation method, we observe the result over eight different stations chosen randomly among the stations, and we compare visually the inferred bike stock and given bike stock.

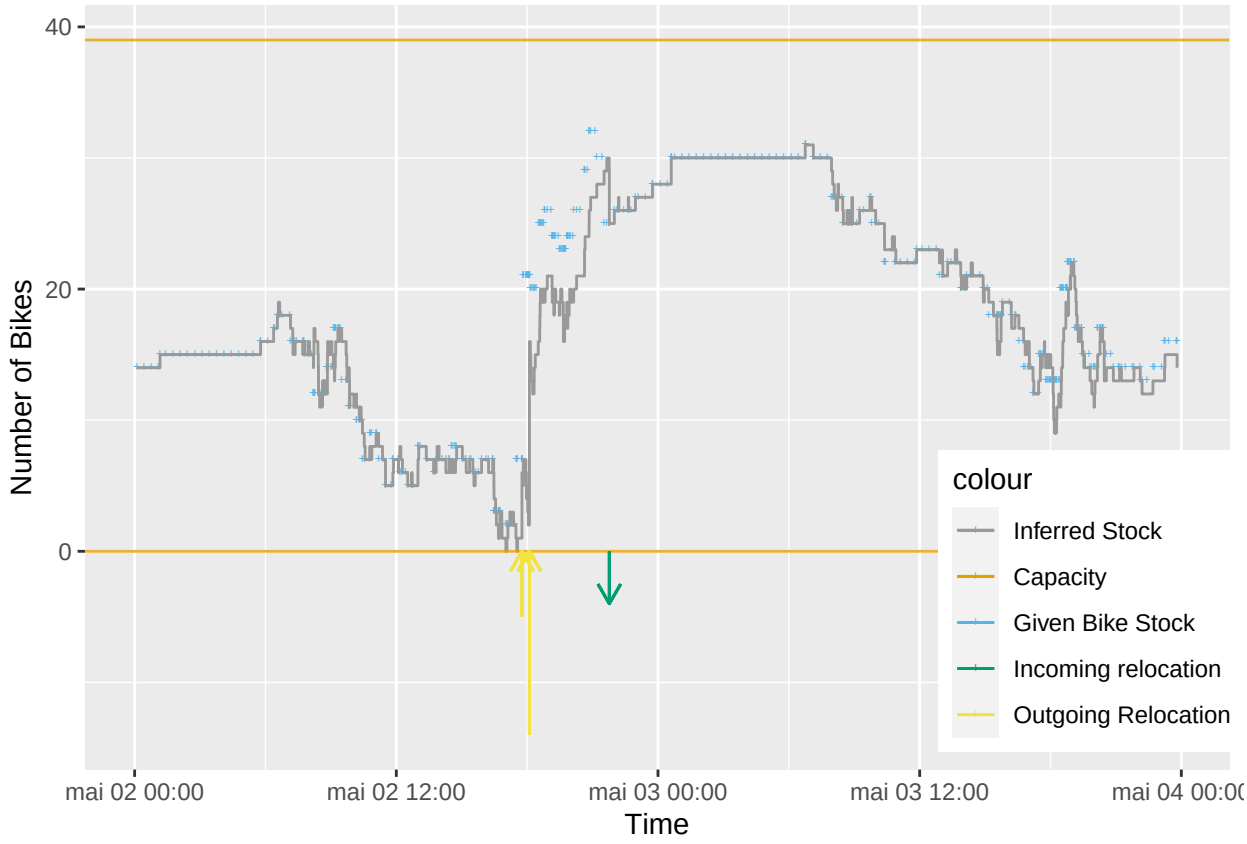


Figure 4.14: Placed relative stock compared to the given bike stock

4.6.1 Visualization of the Rebuilt Stocks for Different Stations

In Figure 4.15 we present the result of the inferred bike stocks compared to the recorded ones for height stations. The first aspect to conclude from those observations is that for all the stations presented the inferred bike stocks are relatively close to the recorded ones. This could indicate that the reconciliation method does work correctly. Nonetheless, it appears from various stations (242, 303, 459) that we infer too many bike relocations. In particular, we infer many relocations of a single bike. In the algorithm we tried to schedule group of at least 3 relocations, but those small relocations are needed in order to keep the number of bike constant in the system. Another flaw of the method is that we did not ensure that the inferred bike stock is below the capacity nor that the inferred bike stock is always positive. In the presented result we can see that the stations 303 and 459 present this kind of errors.

In many stations of the Figure 4.15 (e.g. stations 3254 and 3089), we can observe that the numbers of bike relocations detected are not enough to match the levels of the observed stocks. Finally, it is important to observe that the difficulty of matching the trips with the observed stock is not the same on the eight stations. For stations 72,79,174,3089 the amplitude and the number of relocations to schedule is relatively small leading to good results. On the other hand on stations 242, 303 and 459, since the number of bikes being relocated is important, more errors are possible. It thus reduces the precision of the construction of the inferred stock.

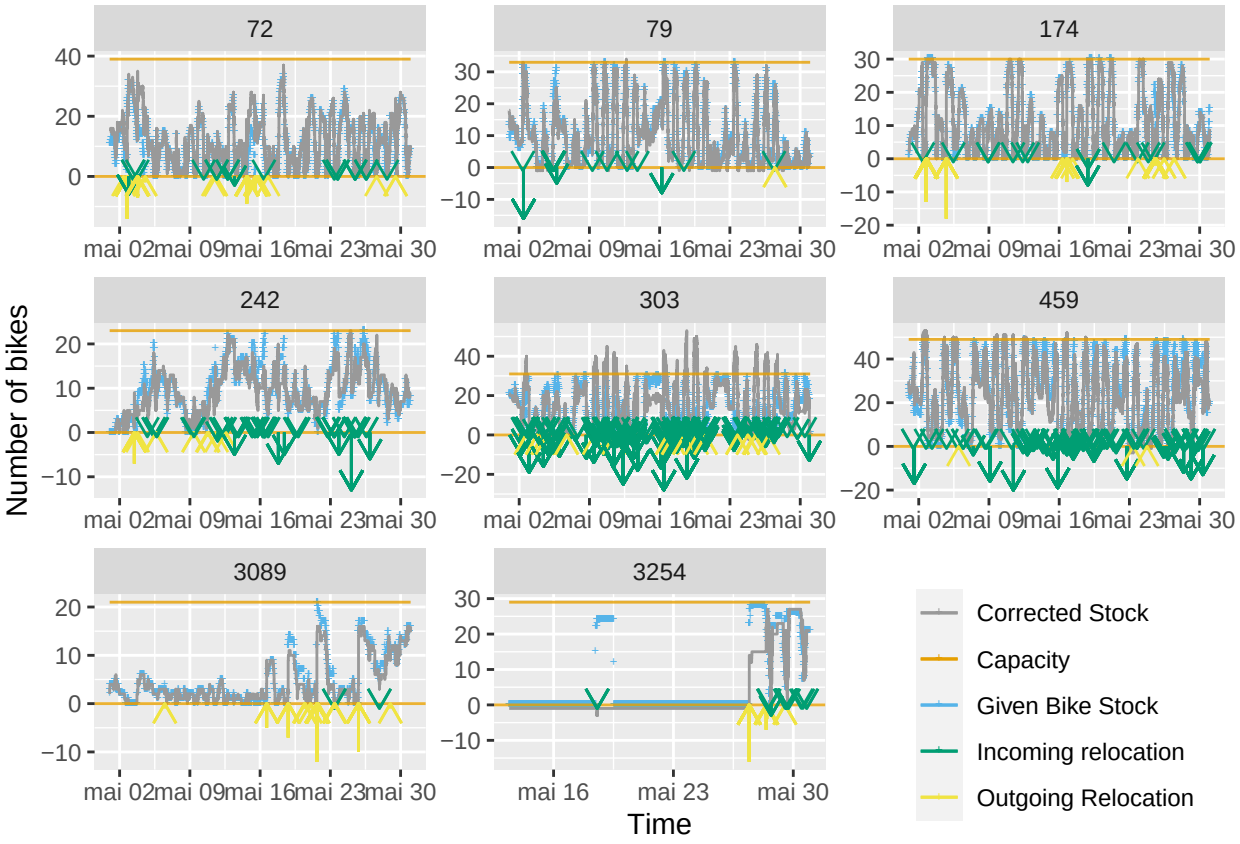


Figure 4.15: Comparison of the corrected stock and the given bike stock for different stations

The case of station 3254 is particular, indeed, the observed stock has important variation and no inferred relocations to explain those variation (allegedly recording errors). In this case, the method is not able to rebuild a bike stock close enough to the observed one.

4.6.2 Time Precision of New Inferred Stock

In Table 4.3 the average time between records of the given stock and the inferred stock. As expected, the time between records is smaller than the 20 mins that we had previously, since the records of the relative stock are more dense than the existing records. We see that in the original data 1% of the records were spaced with less than ten minutes, while 75% of the records were spaced by twenty minutes or more. In the inferred data, 75% of the records are spaced with less than 10min. Furthermore, because the inferred is constructed from the user trip data we can assume that the data we have are continuous, and we have access to the number of bike in stations at each time.

Furthermore, since our method uses the user trips data to calculate the relative bike stocks and build the inferred bike stocks, it yields a more accurate estimation when there are lots of user trips. This accuracy is mainly due to a more accurate calculation of the initial stock level. This is particularly interesting to the estimation process presented in Chapter 5.

Table 4.3: Comparion of time between records

Quantile	Time between records (min)	
	Original Data	Inferred Data
0%	0	0.00
1%	10	0.00
25%	20	0.73
50%	20	2.75
75%	20	9.52
99%	21	20.82
100%	111154	95329.72

4.6.3 Towards a Model of the Stock Reconciliation Problem

The algorithm we propose is not completely satisfactory. We have an inferred stock that does not validate the constraint of capacity of the stations. In fact, if we assume that the bike relocation are correctly detected and that the user trip data is correct, we can define how to construct a solution and the different constraints the solution must verify. This leads to define the *Stock Reconciliation Problem*. We give here a definition of the problem in plain english.

Input:

- A set of stations
- A time horizon
- For each station its total capacity
- For each station a time-serie of given bike stock over the time horizon
- For each station a time-serie of bike events (bike arrivals and departures) over the time horizon
- A distance measure between two time-series
- The set of inferred relocations (start station, end station, release date, deadline)

Decision Variables:

- Relocation operations as (start station, end station, start time, stop time, number of bikes)
- Initial stock level for each station
- Assignement of relocated bikes to the relocation operations

Objective: Minimize the distance between the inferred and given bike stocks while having a minimal number of relocation operations scheduled.

Constraints:

- The inferred stock is computed from the bike events at each stations, the relocation operations and the initial stock level.
- Each bike taken in a relocation operation, must be taken after its release date and

dropped before its deadline.
- Each bike taken in a relocation operation must be taken at its start station.
- Each bike taken in a relocation operation must be droped at its end station.
- At each time, for each station the inferred stock must be below the capacity of the station.
- At each time, for each station the inferred stock must positive.
- All the relocated bikes must be placed in a relocation operation.

With this definition, we can see that our algorithm gives feasible solution of a simplified version of this problem. In such a version, the capacity constraint is relaxed and all the relocated bikes are not affected to a relocation operation.

4.7 Conclusion

In this chapter, we have explored the available data (user trips and bike stocks) on Bike Sharing Systems and, in particular the ones of New York City. In order to understand correctly the behavior of a Bike Sharing System, it is useful to have accurate data on the number of bikes in each station, especially on the bike availability. The user trips can be compared to the given bike stock trough the relative stock. For various reasons, the bike relocations are not recorded in the user trips data which only record the trips made by the users. Thus, the first part of our work was to infer those relocations from the user trip data. From our preliminary results, we can conclude that the detection method we use detects the right amount of relocations for each stations. Nonetheless, the inferred relocations are detected without a precise scheduling. The second part of our work was to schedule those inferred relocations. Once those relocations scheduled, we are able to construct a piece-wise constant and time continuous inferred stock. This new inferred stock is perfectly consistent with the user trips. This is an important improvement. Nonetheless, the inferred stock does not match perfectly the given bike stock data, but it yields close results on all the tested stations. We can conclude that, with this method, we can have access to a more precise stock data than the previously given stock data and, that, this new data is consistent with the trip data. This consistent and precise data is fundamental to the demand estimation module, presented in chapter 3. However the evaluation of the method is partial and must be strengthened with more quantitatives results.

Chapter 5

Demand Estimation

In this chapter, we attempt to estimate the true demand from observed data. For this, we compare several methods to compute a demand estimation as close as possible to the true one. Two methods (Ignore, EDR) originate from BSS publications and another one (NZVT) comes from M/M/1 queuing theory. Since the observed data do not contain the censored demand, we test the methods on synthetic data from the simulation of a bike sharing station with Poisson demand and return rates. The synthetic data allow to simulate observed data and give access to a synthetic true demand. We want to investigate estimation with a time dependent demand parameter and to this end, we split the time horizon in steps, and perform a time independent demand estimation on each time step, i.e. the demand estimation is constant in each time step. To select the appropriate methods which perform a good time independent demand estimation, we compare the three methods over a first set of experiments with a time independent demand parameter. These experiments show that Ignore and EDR are strongly biased in their estimation, and therefore they are discarded for the estimation with a time dependent demand parameter. With the remaining NZVT method, we observe that the quality of the overall estimation depends on the number of observations (days), as well as the number of time steps and the method chosen to create them. We report numerical analyses on all these results.

5.1 Introduction

To optimize the management strategies, a description of the users demand is needed. The observed demand cannot be user, since it does not contain the censored demand. We suppose that there is a true demand for bikes (resp. docks) and that, through a censoring process lack of bikes (resp. docks), the demand is splitted into the censored demand and the observed. In this chapter, we focus on estimating the demand on a single station in presence of direct censoring on synthetic data. This chapter is organized as follows. In Section 5.2, we start by formalizing the underlying estimation problem. Then we present our comparison methodology and compare the different estimation methods in Section 5.3. In Section 5.4, we describe the estimation methods for a time-independent parameter and evaluate the methods with two

numerical experiments. We then extend the method in Section 5.5, to estimate time dependent parameters and quantify the performance on numerical experiments. After concluding on the results in 5.6, we discuss in Section 5.7 of the hypothesis made in this chapter and the scientfic challenges that must be faced before doing the estimation of the underlying demand on a real-life BSS.

5.2 Formal Definition of the Problem

Given historical records of the bike departures, bike arrivals and bike stock at a station, the problem consists in estimating the rates of attempts of bike pick up.

Let us define:

- $\lambda(t)$ the hidden parameter of the Poisson process modeling bike demands.
- $\mu(t)$ the hidden parameter of the Poisson process modeling bike returns.
- $\lambda_o(t)$ the number of observed bike departures at the station.
- $\mu_o(t)$ the number of observed bike arrivals at the station.
- $[0, T]$ the time interval considered, where the Poisson processes are defined, also called the horizon.
- N the number of observations that will be tested.
- S_0 the number of available bikes at the beginning of the time interval.
- $\hat{\lambda}(t)$ the estimated parameter of the Poisson process modeling bike demands.

Bike Demand Estimation Problem: Given N independent and identically distributed realizations of an M/M/1 queue over time horizon $[0, T]$ and S_0 the initial number of jobs in the queue waiting area, the bike demand estimation problem consists in calculating $\hat{\lambda}(t)$ an estimator of $\lambda(t)$ the hidden parameter of the arrival rate of the queue.

Note that in this chapter and in the BSS literature, the bike demand is denoted λ and the bike returns is denoted μ. It is the opposite of the standard queuing theory notations.

When the station is not full, we can observe all the arrivals. But, as soon as it is full, we cannot observe any new arrivals. Since the M/M/1 queue model assumes infinite capacity, it cannot model correctly the station full situation. However, by choosing an appropriate time interval, we can avoid those station full situations.

For all the methods presented hereafter, we focus on the estimation of bike demands and not for bike returns. Both problems are symmetrical, bike returns can be estimated by the same methods if they are seen as bike demands and the lack of available bikes is seen as the lack of available docks.

5.3 Comparing Estimation Methods

5.3.1 General Idea

The main limitation when trying to estimate the true demand, is that censored demand leaves no trace in the observed data. This implies that we cannot simply use the observed system data to test our estimation methods. To overcome this limitation, we propose a comparison approach in three steps: (1) The first step is to construct a synthetic population of potential users. It is common to model users' bike demands as a time dependent Poisson process. We choose to also model the users' bike returns as a time dependent Poisson process. (2) Once these demands constructed, we can simulate the functioning of a station. This simulation will divide the previously defined bike demands in an observed part and a censored part. Since the stations have infinite capacity, there is no censored part on the bike returns. (3) Finally, we can use the observed demand part in each of the estimation methods. Each method should give a demand estimation as a time dependent Poisson Process. This result will be compared to the synthetic true demand constructed in step (1) to qualify the performance of the methods. This comparison approach is also used in (Liu & Pelechrinis, 2021).

5.3.2 Description of the Simulator

For each user, we can simulate the interaction between the user and the station. Because the interaction is symmetrical for bike return and bike demand, we only describe the bike demand interaction. The modeling of bike demands by a Poisson process allows to easily generate synthetic data. Indeed, let's define $\lambda(t)$ the parameter of the Poisson process and $[0, T]$ the time interval. We start by generating users' bike demands according to a Poisson process of parameter $\lambda_{max} = \max_{[0,T]}(\lambda(t))$ (to do this we generate the time between each user as an exponential law of parameter $\frac{1}{\lambda_{max}}$), then for each demand at time t we keep the demand according to a Bernoulli draw of parameter $\frac{\lambda(t)}{\lambda_{max}}$. When a user has a bike demand, if a bike is available then the user takes the bike the number of bike available is updated and the demand is considered successful, if no bike is available then the demand is considered censored.

5.3.3 Error Measure

To measure the difference between the estimated parameter and the hidden parameter, we propose a relative $\mathcal{L}_2$ norm (see Equation (5.1)). This distance is expressed as a ratio of the hidden parameter. Such a measure is only useful for comparing methods on underlying parameters when the comparison can not be easily made. In the time-independent experiments, comparing directly the estimation to the hidden parameter is simpler and more precise.

$$Err(\lambda, \hat{\lambda}) = \frac{1}{T \int_0^T \lambda(t)dt} \sqrt{\frac{1}{T} \int_0^T (\lambda(t) - \hat{\lambda}(t))^2 dt} \tag{5.1}$$

5.4 Estimation of Time-Independent Parameters

In this section, all the considered hidden parameters are time independent, meaning that at all time $\lambda(t) = \lambda$. When there is no ambiguity we will use λ as $\lambda(t)$

5.4.1 Non-Zero-Vehicle-Time Method (NZVT)

The general concept of this method is to make an estimation when the demand can be observed and expand the estimation to the rest of the horizon, thus making a time independent estimation. This method is also presented quickly in (Jian et al., 2016). For the whole interval, we measure the Non-Zero-Vehicle-Time: the proportion of the time when the station is non-empty. We then estimate, for the horizon:

$$\hat{\lambda} = \frac{\lambda_o}{NZVT}$$

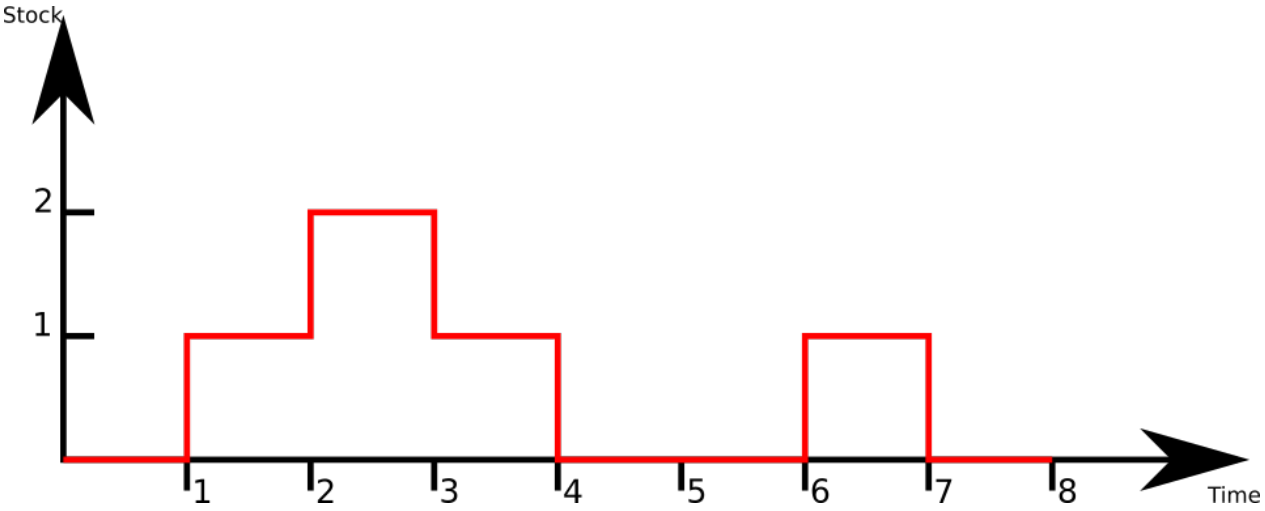


Figure 5.1: Example of bike availability curve to illustrate the estimation methods. (Adapted from (Liu & Pelechrinis, 2021)

Numerical Example: In Figure 5.1, considering a horizon of 8, we count three bike departures for a total Non-Zero-Vehicle-Time of 4. The resulting estimator for this period would be $\lambda = \frac{3}{4} = 0.75$

Particular Case: If during the full horizon the station is empty, then $NZVT = 0$ and thus $\lambda_o = 0$. In such a case the method will not yield any value.

5.4.2 Excess Demand Rate Method (EDR)

This method is introduced by (Liu & Pelechrinis, 2021), again the parameter estimated will be time-independent over the horizon. We look for Excess Demand Pulse pattern: No available

bikes followed by a bike taken quickly after its availability. In the idea of the authors, such a pattern should indicate that some demand that have not been registered. Once the pattern is detected, we measure the average duration of the pattern that we note τ_e. It follows at exponential law of parameter $\frac{1}{\lambda+\mu}$, since it is a sample of the inter-event time. Furthermore, since the station is of infinite capacity, there is no censor at the arrival. We can thus estimate the time between arrival by measuring the average time between arrivals τ_a. τ_a follow an exponential law of parameter $\frac{1}{\mu}$. We estimate the demand parameter by:

$$\hat{\lambda} = \hat{\lambda} + \hat{\mu} - \hat{\mu} = \frac{1}{\tau_e} - \frac{1}{\tau_a}$$

Numerical Example: In Figure 5.1, we count only one excess demand pattern thus making $\tau_e = 1$, and count three arrivals thus $\tau_a = \frac{5}{2}$. The methods yield, an estimation for a horizon of 8, $\lambda(t) = 1 - \frac{2}{5} = 0.6$

Particular Case: If the Excess Demand Pulse pattern is not present in any observations, the methods returns no estimations.

5.4.3 Ignore Method

This method is proposed in (O'Mahony & Shmoys, 2015). As for the others methods, estimated parameter will be time-independent on the horizon. We look for the observations where there is always at least one bike available in the station (the other observations will be ignored). We then estimate λ by the average number of departures on those observations.

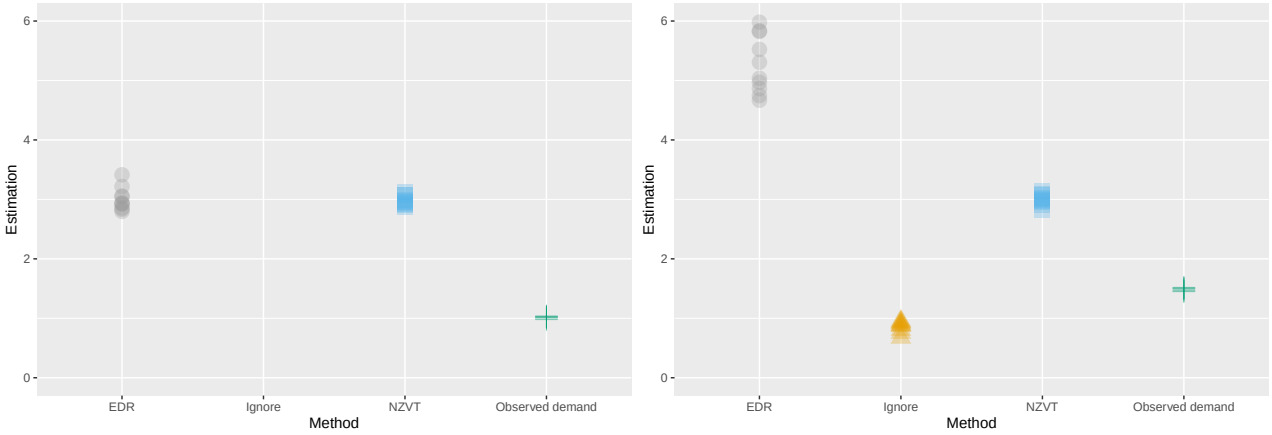
Numerical Example: In Figure 5.1, the only observation is not usable since it is censored. The method will not return any value.

Particular Case: If in all the observations, the number of bike in stations reaches zero. The method can not return any estimation.

5.4.4 Numerical Experiments

Experiments of Liu et Al.
In (Liu & Pelechrinis, 2021), the authors propose an experiment with $\lambda = 3$, $\mu = 1$ and $T = 1000$. Because the number of generated departures is important (and because the first authors did so), we choose to make only one observation. We repeat the experiment 10 times to assess the variability of the methods.

Because we used only one observation with initial stock of 1, the ignore method do not find a valid interval to make a prediction. The method is thus removed from the result for this experiment. We see in Figure 5.2(a) the average number of demand observed and the average prediction made by each method, both of the methods predict a correct total number of demand (around 3000). It is still interesting to note that the method by Excess-demand rate, have a bigger variability in the result than the Non-Zero-Vehicle-Time method.



(a) For a horizon on 1000, in this case the Ignore methods returns no estimation

(b) For a horizon of 1

Figure 5.2: Result of the different method over the 10 experiments for an underlying parameter of 3, each dot is represent one experiment

Variation of this experiment for a low demand period

The parameter tested previously are not realistic, usually when doing demand estimation the size of the horizon is around 30 minutes, the number of observed departures and arrivals is below 20. We thus redo the experiment with more realistic parameters, $\lambda = 3$ and $\mu = 1$. Those parameters correspond to a station with a "low" traffic over the considered time interval. To keep the "number of events" comparable to the previous experiment we increase the number of experiments to 1000. We see in Figure 5.2(b), that the ignore method underestimates by far the demand. It is even less than the observed demand, this is no surprise by construction of the method. We see also that the excess-demand rate method over-estimate the demand. It predicts a demand around 5 with a high variability. In the other hand, the result of the Non-Zero-Vehicle-Time method seems to give a rather precise estimation without bias.

Evolution of the performance of the methods in function of λ

The EDT method seems to over-estimate the demand for smaller demand. To confirm the hypothesis, we design various experiments with variation of λ and μ to observe the predicted demand in function of λ. To ease the interpretation of the result we choose to keep the ratio $\frac{\lambda}{\mu}$ constant. We will use: $\lambda = 3000, 30, 3, 1, 0.3$ along with $\mu = 1000, 10, 1, \frac{1}{3}, 0.1$, and $NumberObservations = 1, 100, 1000, 1000$

We see that, for smaller values of λ, the EDR method consistently over estimate the hidden parameter, by far. This indicates that **the method by EDR is biased** and this bias increases when the parameter reduces. At the opposite, **The NZVT method is unbiased**. For the future experiments, we only use the Non-Zero-Vehicle-Time method.

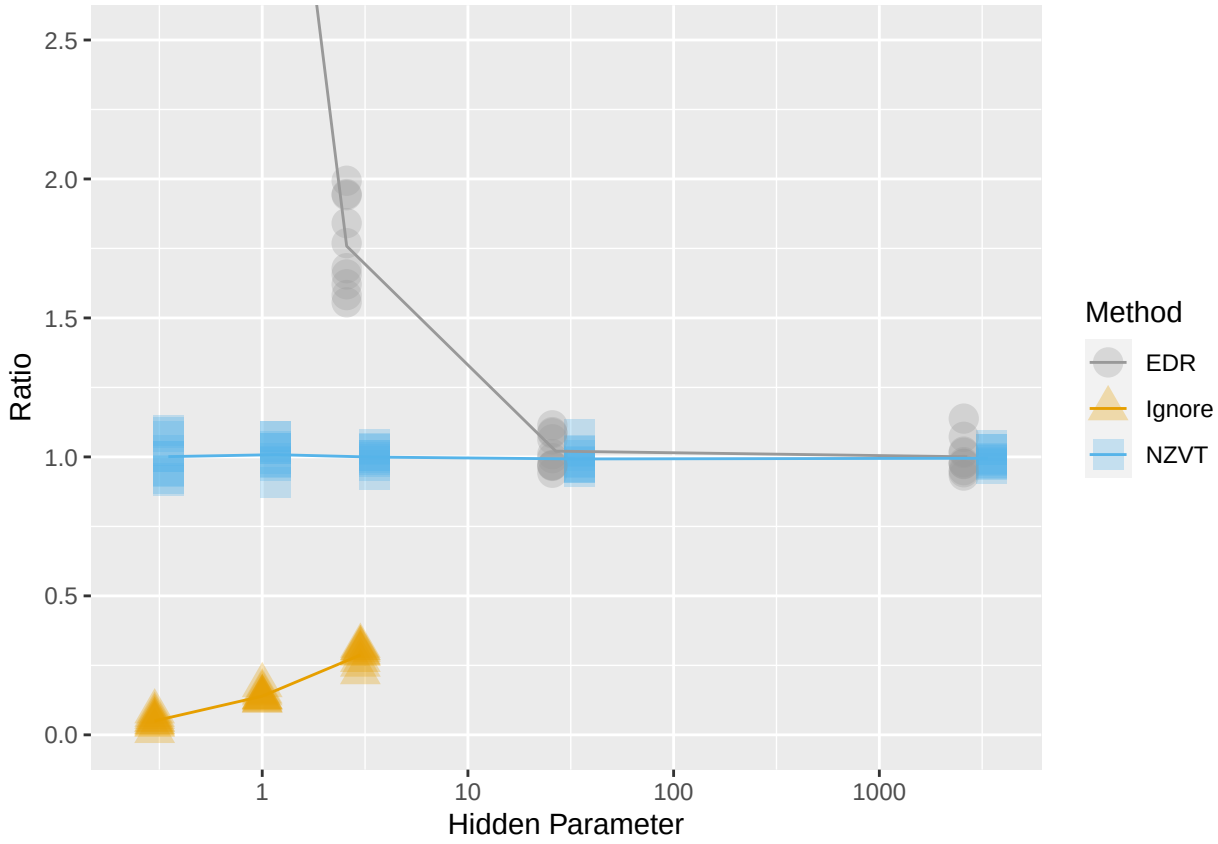


Figure 5.3: Ratio of the predicted demand $\hat{\lambda}$ and the true demand λ for each of the three methods (EDR, Ignore and NZVT). Being closer to one is better. Each dot represents a single experiment, the line joins the mean over the 10 experiments. For $\lambda = 0.3$, the ratio for the EDR method do not fit on the figure

5.5 Estimation of Time-Dependent Parameters

We now test the estimation on time varying parameters.

5.5.1 Time Splitting Methods

To estimate time dependent parameters, we decompose the problem in two steps. First, we split the full horizon in time-steps. Then, for each time-step we do a constant estimation with the NZVT method. To do the splits, we propose to compare three methods.

Equal time division We start by choosing a time length of interest (usually chosen by finding the characteristic time of demand pattern). In a majority of works the time length chosen is 30 minutes or 1 hour. Then the whole horizon is divided in time-steps of the time length chosen. The main limit of this method consists in station with high variability in the demands amplitudes. Indeed, when the demand is low, the time-step chosen could lead to periods with few events, which makes harder to make a precise estimation. On the other

hand, periods with high demand are poorly estimated because the time step is too large to describe the changes in the demand parameter.

Equal departure division In this method, we choose a number of departures and divide the horizon into time-steps with equal number of departure. The main advantage of this method, is that it solve the problem of the equal time division. We can hope the estimation will have the same precision all the sub-intervals. The main limit is that it is harder to decide the good number of events to use.

Equal Event division This method divides the horizon into time-steps with equal number of departures and arrivals. However, because the censor process correlates the bike demands with the bike returns, choosing to make sub-intervals with equal number of departures and arrivals could try to have approximately the same precision in the estimator.

Particular Case of missing values It can happen than, for a given time-step, the estimation method can not yield any result. In such a case we choose to use the previously available value. If there is no previously available value, we will use the next one available.

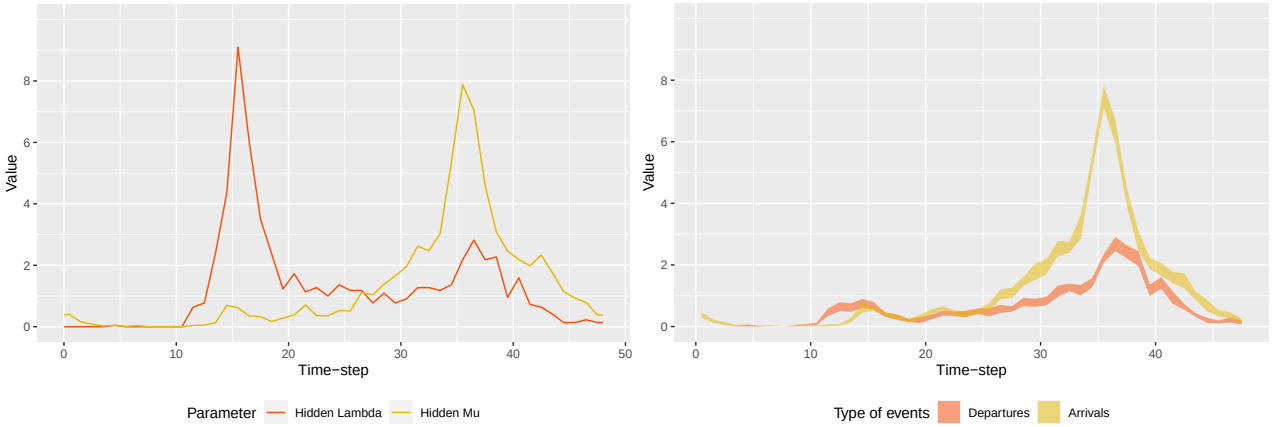
5.5.2 Parameters of the Experiments

To test the decomposition methods, we use the estimation made by Raviv et al. in (Raviv et al., 2013). We use the parameter as piece-wise linear. We chose a station that will face at censor and that will have a variation in the parameters. We will use the real capacity of the station as capacity for our station. We start the simulation with an initial stock of 1.

In Figure 5.4a, we represent the parameters of bike demand and bike returns for the 48 half hours of the day. We see that, at the beginning of the day, the arrival rate and the return rate are low and relatively close, we should not observe much censoring. In the morning peak we observe more departures than returns, one can imagine the station placed in a residential area where the users drop-off bikes at the beginning of the day.In the rest of the horizon we see a period of lower activity followed by an evening peak. During this period the bike demand rate is higher than the bike return rate. If we look at the observed bike demands and returns observed in the experiment (see Figure 5.4b, we see that the observed demands and returns does not have the same pattern as the parameters input. This confirms that there is censor at the departure, especially in the morning peak.

5.5.3 Comparison of Time Division Methods

In Figure 5.5, we compare the errors of the three time-splitting methods in function of the number of time step chosen. In the figure, we distinguish between four different performance behaviors. (1) When the number of time-steps is minimal (3 divisions), we see that all the time-splitting methods have a high but similar error. This indicate that there is too few time-steps.
(2) When the number of time-steps is small (6-24 divisions), the equal time splitting methods have a better performance than the two other methods (the worse one being the equal event



(a) Hidden $\lambda(t)$ and $\mu(t)$.

(b) Representation $\lambda_o(t)$ and $\mu_o(t)$ the areas correspond to the maximum and minimum along experimentation.

Figure 5.4: Hidden parameters and realizations along experimentations

splitting method). We will explain this result below when describing the behavior of the methods for 12 divisions.
(3) For an intermediary number of time-steps (48-192 divisions), none of the three methods seems to have a better performance than the others. Nonetheless, it is important to note that, it is in those number of time-steps that the minimal error can be found. This tends to indicate that, if the number of time-steps is correctly chosen, the method used to determine the steps seems of little importance.
(4) Finally, when the number of time-steps is important (384-768 divisions), the equal event splitting method seems to have a better performance than the two other methods (the equal event method being the worse). It must also be noted that the variability of the error measure increases with the number of time-steps chosen. We explain this behavior by the fact that smaller time-steps leads to an increased variability of the estimator (cf. time independent experiments with small parameters) which translate into over-fitting.

In Figure 5.6, we represent the estimated parameter with the three time division method for 12 time-steps along with the underlying parameter. In this figure, we can observe clearly the difference of behavior in the three methods:

- The equal event method, makes a single time-step for the whole [0,18] intervals. At the beginning of the day, because μ is relatively small and because there is much censor, the time-step must be bigger to equalize the number of events of the other time steps (in particular at the end of the day, there is no censor and high μ which generate many events). This bigger time-steps transfer into a poor estimation of the morning peak.
- The equal departure methods suffers from the same limitation but, with better results. Indeed, the difference in term of number of departure between the beginning of the day and the end is smaller than the difference in terms of number of events.
- Finally, the equal time method is more precise because it is able to give a better estimation of the morning peak.

From this first set of experiments, it is not possible to conclude on a method that would

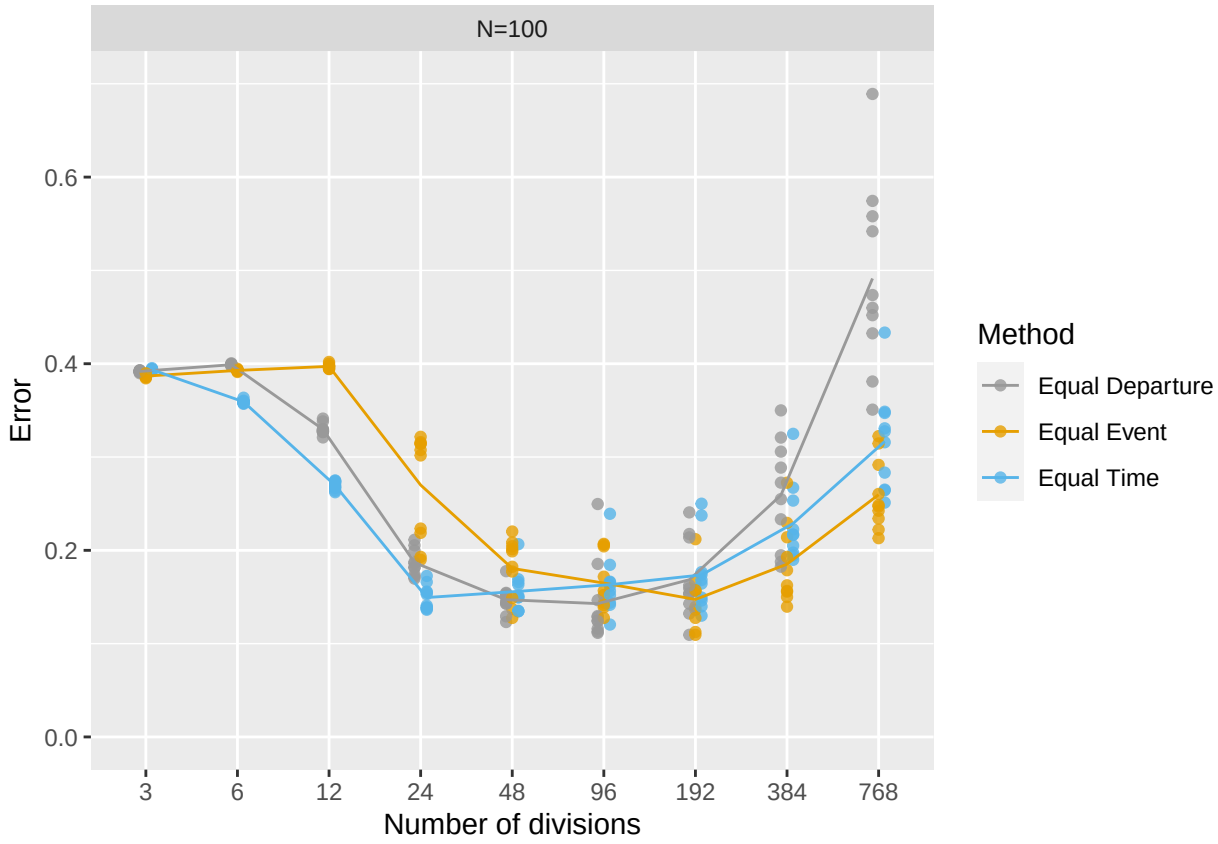


Figure 5.5: Evolution of the error as function of the number of division chosen, depending on the time-splitting method each dot represent a single experiments the colored line represents the mean over the 10 experiments

be better than the others. For the three methods, the number of division chosen have an important impact on the performance. Nonetheless, for all the methods, when the number of division is correctly chosen, the performance is similar even if the behavior is different.

5.5.4 Impact of the Number of Observations

As said previously, the size of the time-step must be chosen to depend on the number of observation available and the intensity of departures and arrivals. In the previous section, we choose to have 100 observations per experiment, this number of observations is unrealistic. Indeed, it is impossible to gather data from a hundred of homogeneous days. And, even if some factorization methods are used in order to group similar station and/or similar days, it is impossible to ensure the days are homogeneous. To finalize the assessment of our estimation methods, we observe the behavior of the estimation methods when changing the number of observations per experiment.

In Figure 5.7, we compare the error measure depending on the number of time steps chosen and the number of observation available. As expected, if the number of observation reduces

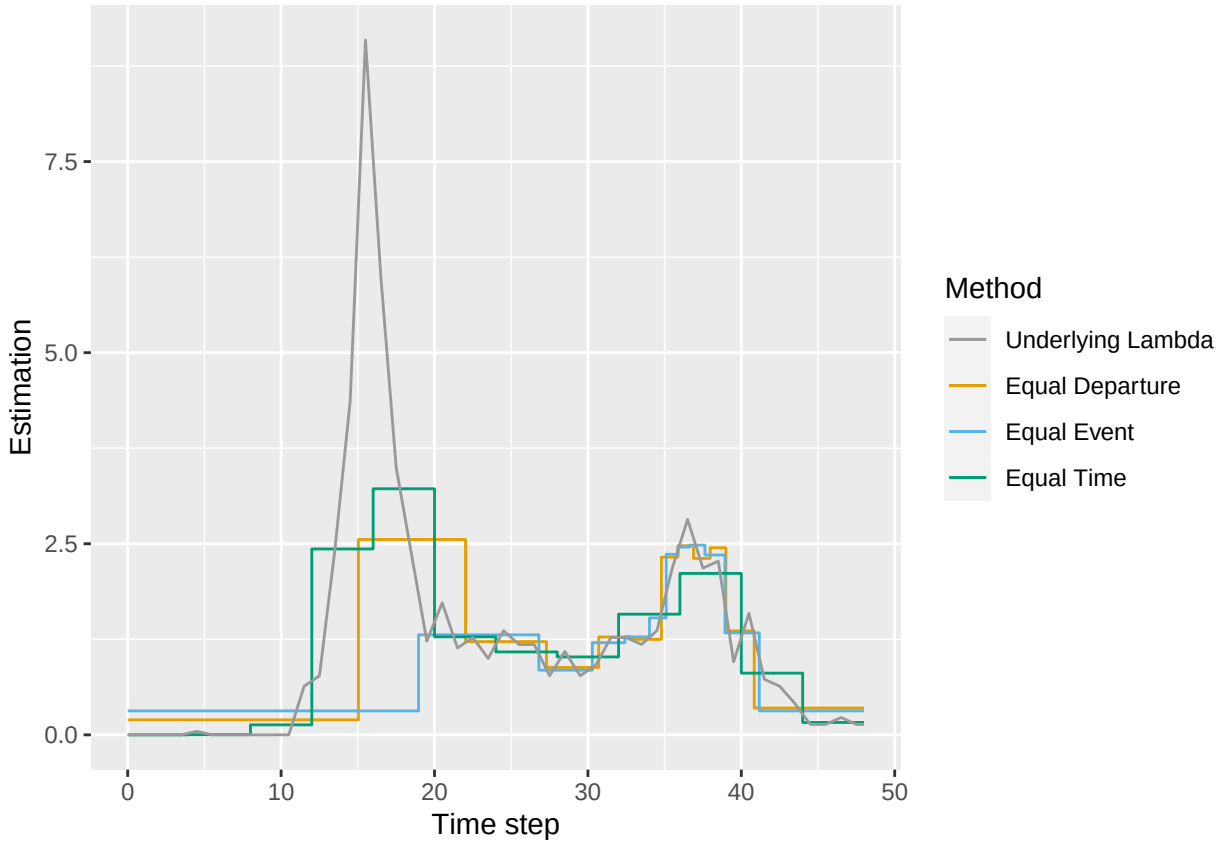


Figure 5.6: Comparion of the different estimation depending on the method chosen for 12 time-steps

the estimation, using a big number of time steps will have a bigger error. For 100 observations the 192 time-steps still have a small error while with 10 observations 96 observations is already too much for two methods. As said previously, for a given number of observations, for a given method, when the number of time steps is too important, the error increases drastically. For a given number of observations, we can order the different methods depending on the minimal number of time steps that makes the errors explode. We see that, for a 100 observations, the three methods have a relatively close error for 384 time steps. But, the errors tends to separate when doing 768 time-steps. On the other hand for 10 observations, at 48 time-steps the equal time method have already an error much higher than the other methods (for which the separation will be made at 96 time-steps). This result can indicate that, the method by equal event seem to be more robust to the reduction of the number of observation. This is confirmed by the fact that, when the number of observations is divided by 10, the error for 96 time-steps with the equal event method goes from 0.16 to 0.31. It is interesting to note that, for all the number of observations tested here, choosing to do 24 time-steps is always a good number of division to be made (for 100 and 30 observations, 48 time-steps yields better but similar result). This, added to the fact that for such a number of time steps the equal events methods is always among the methods with the best result, can lead to a satisfactory estimation method for time-varying parameter.

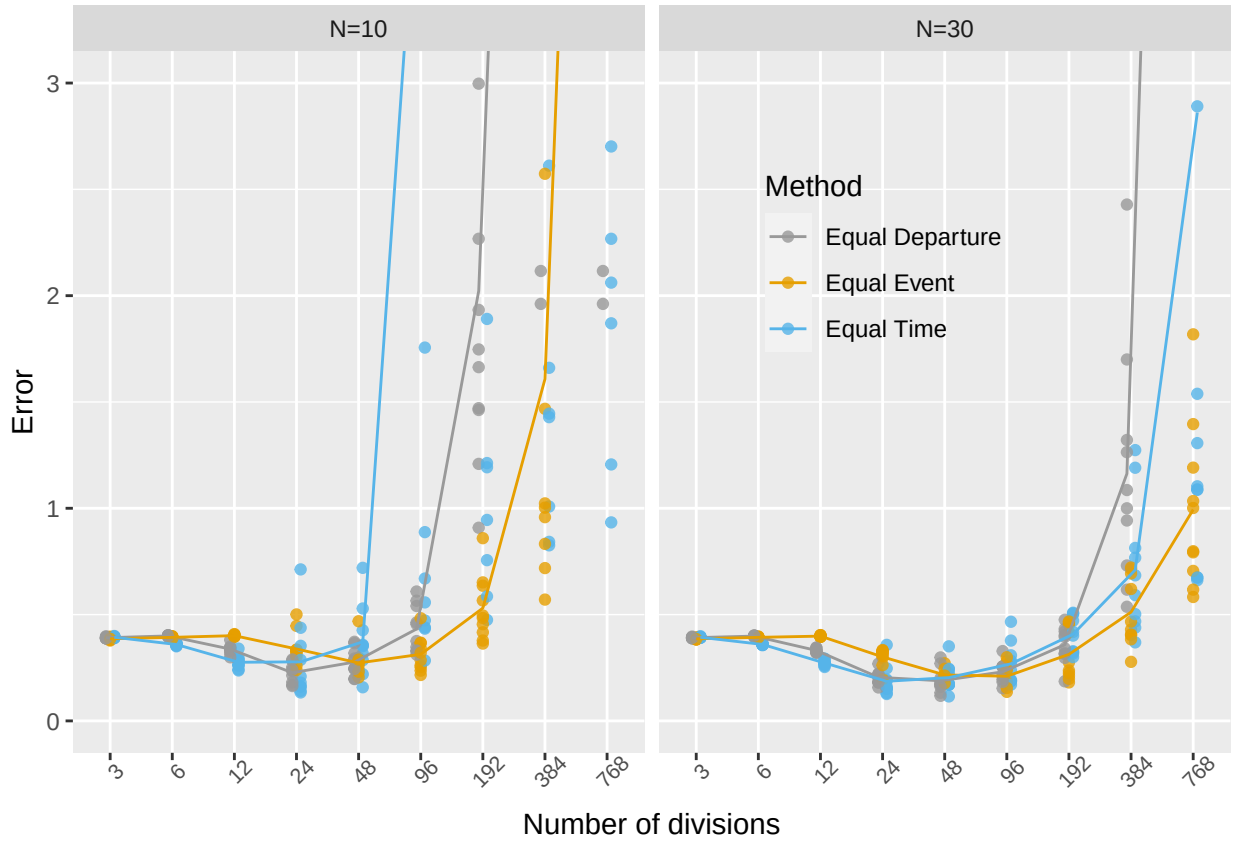


Figure 5.7: Evolution of the error in function of the number of time steps chosen and the number of observations, each dot represent a single experiment the line represent the mean over the 10 experiments, errors above 500 are not represented on the figure

5.6 Conclusion

In this chapter, we compared different demand estimation methods on synthetic data we generated with a time independent Poisson parameter. From those experiments, we showed that two of the three methods present a bias in their estimation. The last method seems to make an unbiased estimation independently of the values of the parameters. We then proposed different time-splitting methods to adapt this estimation method to time-varying parameters. To evaluate those methods, we propose an error measure to simplify the comparison of the different time-splitting methods. When comparing those methods in the experiment we proposed through the error measure, no method appears better than the others. Nonetheless, the methods have different behaviors. This difference in the behaviors can help to choose between the different methods. The performance of the methods depends, on the number of divisions chosen and, on the number of observations used to do the estimation. To understand the behavior of the methods when the number of observations varies, we repeated the previous experiment with different number of observations. When the number of observations reduces, the methods loose in performance. Nonetheless, one of the method seems to be more resilient to this reduction. To summarize our work in this chapter, we can say that the method

we presented as the Non-Zero-Vehicle-Time method is the only unbiased method in our comparison. On the time-varying experiment we can not give a reliable conclusion on the performance of the methods and other experiments should be conducted.

To extend the work presented in this chapter it could be interesting to improve the time-splitting methods, the idea behind the equal-departure and equal-event time splitters, is to keep the performance of the estimator constant. To respect this idea, it would be much more precise to design a splitting method keeping $\frac{\lambda_o}{NZVT}$ constant. Also, the conclusion drawn in this work are not strongly affirmed. The performance of the NZVT method on time-independent methods, could be affirmed through a formal demonstration. It would also be fundamental to reproduce the conclusion we obtained on the time-splitting methods for other values of λ and μ.

5.7 Limits of the Approach

In this chapter, we made various assumptions to obtain conclusions. The conclusion we draw from this work depend strongly on those assumptions, before doing the estimation on a real-life bike sharing system those assumptions must (at least) be questionned. The objective of this paragraph is to present the lacks of the hyptotheses we made and discuss scientific challenges the alternatives raises.

5.7.1 Exogenous Parameters

Bike demand is impacted by exogeneous parameters such as season, rain, temperature etc. In this chapter we do not consider such variables. While being often taken into account in different works on estimating observed demands or stock, such variables have never been taken into account in underlying demand estimation work. A minimal working solution could be to group the observation by the values of those exogeneous parameters. However, such a solution could lead to an insufficient number of observations to do the estimation. It would be much more precise to use the exogeneous parameters inside the estimation process.

5.7.2 Direct Censor

In this chapter we considered only direct censor. We call the censor direct, because we consider that the users have made their choice of using (or not) the system independently of the availability of the system. In real-life systems, the modal choice is impacted by various factor (a user that had a bad experience with the system is less likely to re-use the system (Kowald et al., 2022)). This could lead to indirect censor, being that a user is not observed in the historical data even if bike a present in station. Estimating this indirect censor can not be done from the historical data, and is a complex process involving various scientific fields (Eren & Uz, 2020). Nontheless, this hypothesis of estimating only the direct censor is shared in multiple publications on demand estimation. It allows to easily construct realistic

demand from the observed demand, that would be used to validate management strategies optimization algorithms.

5.7.3 Single Station

We focused here in estimating the underlying demand on a single station. The conclusions we draw will be different if we take into account a network of stations. In such a case, appart from censor, the redirection shift observed demand away from the real demand. The redirection is when a user does not find a bike (or a dock) at the desired station, she might go to another station in order to take another bike (or leave the bike) and continue her journey.
The historical data do not keep track of such events. Worse, a rebound event will be registered in the historical data as a successful departure in the wrong station, and estimated as a censored event at the origin station. To our knowledge, no publication propose a method to estimate those redirections, and we have no clue on how to do such an estimation.

5.7.4 Poisson Hypothesis

On a real life system, the independance of the user arrival is not always verified. As example, for stations placed near train stations (or other multimodal-hubs) the arrival of the user depends on the arrival of the trains. This could lead to a station emptied quickly by an arrival of a group of users. This would leave the station empty for the rest of the horizon. In such a case, the estimator we designed will strongly over-estimate the demand at this station. These behaviors question the Poisson hypothesis made on the demand (in (Rudloff & Lackner, 2014) the authors conclude that the observed departures and returns would be better modeled by a Negative Binomial Process).
In the estimation problem, we could replace the M/M/1 queue by a G/G/1 queue. In such a case, the global methodology of comparison would remain the same. Nonetheless, the error measure should be modified in order to allow measuring the distance between a single parameter function and a multiple parameter function.

5.7.5 Variation on the Comparison Methodology

At the beginning of our work, we tried to estimate demand through regression methods such as Hurdle Model (Rudloff & Lackner, 2014) and Zero-Inflated Poisson models we adapted from other problems (Atkins & Gallop, 2007). However, these trials were unsuccessful and we changed our approach.

In (Negahban, 2019), the authors propose a completely different comparison methodology along with another estimation method. The order of the operations differs from the one we propose in this chapter, they start by making a prior estimation from the observed departure data, and simulate the censor process to output simulated observed departures. Using a non-parametric time-series distance measure, the authors can compare the observed departure

data and the simulated observed data. With this distance measure, they can adapt their estimation to reduces this distance and improve the performance. They justify the usage of this methodology by arguing that, the estimation methods we propose can not yield efficient estimation when the number of observations is small. Nevertheless, they do not verify this assumption, and our experiments do not confirm their hypothesis.

This methodology present advantages compared to ours:

- The non-parametric comparison allows for an easier transfer to real-life historical data.
- The methodology would remain the same if the parameters and estimation were not Poisson.
- The methodology would remain the same to adapt to different simulation hypothesis.

Nonetheless, this methodology present some major disadvantages:

- The non-parametric comparison of the time-series is a complex process and hardens the understanding and the evaluation of the methodology.
- While being easily adaptable to real-life data, this methodology does not seem to be easily adaptable to including exogenous parameters such as neighboring stations or weather data.

Other comparison methodology could also be used (BIC, AIC, etc.). However, when estimating underlying demand, such methods are not easily applicable since there is no record of the censored demand in the data.

Chapter 6

Optimization Algorithms and Reproducibility

In this chapter we reproduce the work of (Datner et al., 2017), we detail the reproduction process and the results we obtained. This reproduction as a reproduction work failed since, while being able to re-code all the algorithms and re-generate data from the authors' parameters, we have quantitative results at every phase of the work. Nonetheless in an optimization point of view the reproduction strengthens the original results since we observe a higher improvement of all the indicators than the original authors expected. In Section 6.1 we present the reasons for such a reproduction and detail how we chose the work to reproduce. The rest of chapter is an extended version of a submission to ReScienceC in which we detail the reproduction process and the results.

6.1 Objective and Method

The initial objective of this thesis was to study bike-sharing systems and propose ways to optimize those systems. With the new direction of the thesis, in order to compare different works between them having functional optimization solutions. It is then natural, in the spectrum of this work, to propose an optimization method.
As we saw in the literature review, the diversity in technologies as in relocation methods, propose to the operational research community a large variety to problem to consider. We proposed a decomposition of the problems in two sub-problems an inventory level problem followed by a routing problem. This first problem, is presented as a basis almost independent of the bike technology and the relocation method studied. Furthermore, once this problem studied it should be easy to add a routing problem in order to make this work comparable with other works that can be found in the literature. This makes the inventory level problem the perfect first problem to consider when studying optimization of bike sharing systems.
As a reminder, the inventory level problem consists in determining the number of bikes to place in each station in order to optimize the user satisfaction over a given time-horizon.

For this optimization part we decided not to produce a new work but rather to reproduce an already existing work. This firstly comes from the objective of the thesis being to compare works and to provide modular solutions, before proposing new optimization methods, having a baseline work to compare future results to is primordial. This comes with the interest for reproducible science, of which we discussed earlier, the second objective of this work was to assess the simplicity of reproducing the results of an existing work.

Once the decision of reproducing an existing work was made and the studied problem was chosen. The question of the choice the publication appeared, multiple criteria were used in this choice:

- The publication had to be recent in order to increase the chances of successful reproduction.
- The publication had to be cited by other to measure the impact of the publication on recent studies.
- The notoriety of the authors gave us good confidence on the quality of the work.

We chose "Setting an Inventory Level in a Bike Sharing Network" by Sharon Datner, Tal Raviv, Michal Tzur and Daniel Chemla.

It must be noted that verifying the reproducibility of a result by reading only the published article is as complex as verifying the validity of the results. Nevertheless, some hints can be used, if the publication is clear without ambiguities it is the first step to a reproducible paper. Often on easily reproducible papers some supplementary material is available, either an extended version of the paper or Laboratory notes made during the time the research was conducted. Last but not least a good insight on the reproducibility potential of a publication is the software available, by software we mean all the resources needed to re-run the experiments from the data to experiment conditions (third part libraries, hardware description) with obviously the code itself. In the paper of (Datner et al., 2017) most of those conditions are fulfilled, the document is clear and precise, the internship report of Sharon Datner can be found which gives more precision on the published version, the data used for running the experiments can be found on Tal Raviv's website. But no code nor software environment is available. In order to reproduce the results all must be re-coded.

When doing this reproduction we followed a simple method, once we read the paper listing all the different steps involved and looked for algorithm description in the supplementary material, we contacted the authors to ask for code or Laboratory Notes. Pr. Raviv was very helpful but could not provide us any document in addition than the ones available on the web. We then reproduced the algorithms based on our understanding, while asking Pr. Raviv for clarifications if needed.

6.2 Description of the System

6.2.1 System Considered

The authors choose to work on a station based system. In such a system, there is a finite number of stations. A given station k can host up C_k bikes, and C_k is called the capacity of station k. The number of bikes at time t in station k is denoted by $B_k(t)$. They also consider the geography of the city as bi-modal and time-independent. This means, that to go from a station i to a station j, a user can ride a bike, which takes T_{ij} or walk, which takes W_{ij}. Not other transportation mode is proposed, and the travel time does not depend on the hour of the day when the trip is made. They also make the assumption that the user will not abandon the journey and will always reach his destination.

Those assumptions represent the fact that the demand that will be used is the extracted modal share of the bike-sharing system. And that the users use this transportation mode as their principal transportation mode. With those assumptions made we can, without loss of generality, assign to each user a departures station, being the station from which he would in first intent take a bike, and an arrival station, being the station from which he would in first intent drop a bike. This allows to consider user that can be described as a triplet with origin station, departure time and destination station.

6.2.2 Journey Dissatisfaction Function

The objective of (Datner et al., 2017) is to solve the *BSIP-SI : Bike Sharing Inventory Problem with Station Interactions*. This problem consists in determining the optimal number of bikes to place in each station in order to optimize a *journey dissatisfaction function (JDF)*. As the name suggests it, the JDF is a function that maps the journey of each user to a real value. It is natural to restrict the set of possible values to positive values, making then 0 the value for a journey with perfect unfolding.

In the publication, as in (Kaspi et al., 2014, 2016) the JDF considered is the *excess time*. The Excess time measure the time lost by the user trying to reach his destination. Ideally, a user who wants to go from a station i to a destination j should take T_{ij}. In practice, however, depending on the availability of resources (bike or empty slot to return a bike), the actual duration of the trip might be longer. For a given trip we define the excess time of this trip as:

$$\text{Excess time} = \text{Actual duration of the trip} - T_{ij}$$

By similarity, for a given initial distribution of bikes in the system, we call excess time of the system (or excess time) the sum of excess time of all the users.

6.2.3 User Behavior Model

In practice, the actual duration of a trip depends on a **user behavior model**, which specifies how users take their decisions to complete their trip. In this user behavior model, a user

has a journey from an origin station i to a destination station j, and takes decisions when interacting with the system. The user starts in station i and will either rent a bike there if one is available or walk to neighboring stations to rent a bike. They will then travel to their destination and return the bike there if possible, or ride to a neighboring station and then walk to their destination.

The decisions of a user are greedy and aim at minimizing their actual journey duration based on their current information of the system. Assuming that the user wants to rent a bike from a station i at time t. If $B_i(t) \geq 1$, the user rents a bike at this station and rides to their destination. Otherwise, the user walks to another station k that is chosen so to minimize their total travel time according to (6.1):

$$k^* = \underset{k:B_k(t)>0}{\arg\min}(W_{ik} + T_{kd}) \tag{6.1}$$

It may happen that, when arriving in station k^* at time $t+W_{ik^*}$, no bike is available at station k^* because someone else took it. In this case, the user repeats the same local search, starting from the current station k^* instead of i, until they find a bike or arrive to their destination.

When arriving to the destination station j at time t, if the station has available slots (i.e. $B_j(t) < C_j$), then the user leaves the bike there. Otherwise, the user rides to another station k^* that is chosen to minimize the travel time (6.2) and then walk to their destination station j:

$$k^* = \underset{k:B_k(t)<C_k}{\arg\min}\ (T_{ik} + W_{kj}) \tag{6.2}$$

Again, it may happen that when arriving at station k^* at time $t + T_{ik^*}$, there is no slot available. In such a case, the user repeats their local search, starting from the current station k^* instead of j, until finding a place at a station to return the bike.

6.3 Methodology

6.3.1 Multi-Agent Simulator

Simulation of complex systems are perfect to use multi-agent simulation. The general idea behind multi-agent simulation is to describe algorithmically how users evolve and interact with each other. This avoids having to encode all the simulation step by step. In this work we have two types of agents to detail:

- Users: They follow the user behavior model, they thus interact with stations and bikes but can not interact with each other.
- Stations: They interact users to allow pickups and dropoffs, but also to give to the users the number of bikes actually contained in the station.

The other benefits of multi-agent simulation is the possibility to easily make measurements during and after the simulation. In addition to the excess time used as objective the authors choose to measure:

- The number of ideal rides: The number of user whom the trip can not be improved. This is observed with the total number of rides through the ideal ride ratio.
- The number of shortage and surplus events, along with the average number of shortage and surplus per non-ideal ride. This gives an insight on how the time is spent by the users, if this number is relatively low, not finding a bike (resp. dock) implies a great excess time for the user.

6.3.2 Optimization Process

With the scenario generation explained we can provide a more precise definition of the problem studied by Datner et al. Given a set of demand scenarios find an initial distribution of the bikes among the station (the same for all the scenarios) in order to minimize the average excess time of the system on this set of scenarios. The constraints are naturally extracted from the system (Inventory constraints) and form the users (demands must finish), this leads classically to an Integer Linear Program to solve this *BSIP-SI*, this ILP is intractable for all the instances, they use the linear relaxation of this ILP to get lower bounds on the solutions, those lower bounds are far from the feasible solutions found (see table 6 of (Datner et al., 2017)). We estimate that the numerical reproduction of this ILP was not informative on the main result of this paper, so we choose not to reproduce this part of their work.

To find feasible solutions, the authors propose a guided local search method. In a local search, a solution is locally modified in order to find a new solution that is kept if and only if this new solution shows better performance than the best until then. The guide is used to generate the new solutions to be tested. The authors base their guides on two theorems:

> *For a given demand realization at a station, consider the sequence of shortage and surplus events. Let $n \geq 0$ be the number of shortage events that occur before any surplus event. Then, increasing the initial inventory by l bicycles will result in the elimination of at most $min(l, n)$ shortage events.*

And symmetrically

> *For a given demand realization at a station, consider the sequence of surplus and shortage events. Let $n \geq 0$ be the number of surplus events that occur before any shortage event. Then, decreasing the initial inventory by l bicycles will result in the elimination of at most $min(l, n)$ surplus events.*

This allows to focus the guides on the first shortage or surplus events. For a station i and a given scenario, we consider the four following events:

1. A shortage event occurs before any surplus event.
2. A surplus event occurs before any shortage event.
3. The station becomes empty without any shortage nor surplus event.

4. The station becomes full without any shortage nor surplus event.

Note that for a given scenario, the third and the fourth events can occur whereas if the first or the second event occur, then none of the other can occur. The reason behind these events is that, for a given scenario, if event 1 occurs for station i, then adding a bike in this station can only be better. Similarly, removing a bike if event 2 occurs can only better. The difficulty comes from the fact that when we have many scenarios, there might be stations for which event 1 occurs from some scenarios but event 2 occurs for others. In order to deal with this, the authors propose two guides alternatives: the *occurrence-driven* and the *time-driven*. The first is based on occurrence: for an event $k \in \{1, 2, 3, 4\}$, we denote by $M_{i,k}$ the number of scenarios such that event k occurs. The second is based on measuring the variation of excess time. For a given scenario s, a station i, we denote by:

- $m'_{i,1,s}$ the excess time induced by the first shortage event in case of event 1, that is, if the trip that causes the shortage is from station i to station j at time t, we measure this excess time as $\min_{k:B_k(t)>0} W_{ik} + T_{kj} - T_{ij}$.
- $m'_{i,3,s}$ the excess time induced if the trip creating event 3 is instead a shortage, that is, if the trip that causes the station to be empty is from station i to station j at time t, we measure this excess time as $\min_{k:B_k(t)>0,k\neq o} W_{ik} + T_{kj} - T_{ij}$.

The definition for event 2 and 4 are symmetric. We then define $M'_{i,k}$ as the sum of the $m'_{i,k,s}$ over all the scenarios. Note that if in the original paper the definition of $M'_{i,1}$ and $M'_{i,2}$ is unambiguous, the definition for the events 3 and 4 is subject to interpretation. Our definition of $m'_{i,3,s}$ reflects our understanding of their definition.

The local search then works as follows:

- If $M_{i,1} > M_{i,2} + M_{i,4}$ (or $M'_{i,1} > M'_{i,2} + M'_{i,4}$ for the time-driven search): We add a bike in station i as it should improve the objective function in a majority of scenarios.
- If $M_{i,2} > M_{i,1} + M_{i,3}$ (or $M'_{i,2} > M'_{i,1} + M'_{i,3}$ for the time-driven search): We remove a bike in station i as it should improve the objective function in a majority of scenarios.

For both evaluation the optimization process remains similar, the simulation process allows to compute the M (resp M') variables. With those variables we modify the solution as said previously before re-simulating to measure the improvement. When no improvement is made or when we find a state that has already been visited we modify the stock of each station by adding a random integer between -2 and 2 (independently for all stations). Before starting the guided local search, an initial starting point must be chosen. The authors consider two initial starting points:

- Each station is filled at half its capacity.
- Each station is filled according to the results found in (Raviv et al., 2013).

Figure 6.1 presents a summary of this whole simulation-optimization process.

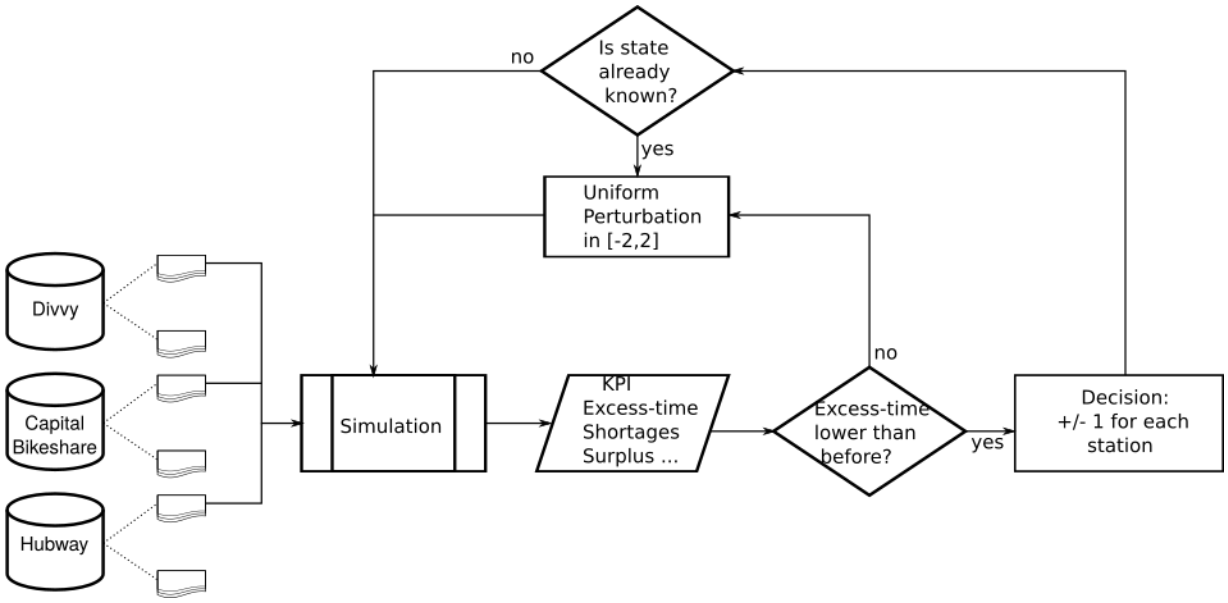


Figure 6.1: Simplified flowchart of the simulation-optimization process

6.3.3 Our Assumptions on Unclear Points

For our reproduction, we made some assumptions on the implementations of the algorithms, because the algorithms are not extensively described in the publication. We asked the original authors for clarifications, but they could not provide us exactly what they implemented. The major assumption we made is the calculation of M_3 and M_4. For a given station if the initial stock is 0 (or if the station starts full) and that this station does not face any shortage nor a surplus we choose to include this extreme starting point in the calculation of M_3 and M_4. Keeping it consistent with the definition given by the authors, the same decision was taken for M_3' and M_4'. Note that these choices impact only the local search, hence they may have an impact on the results in Table 6.3 but not on those in Table 6.2.

Also, we list below other minor points on which we had to make a decision. Yet, we believe that these decisions are the same as those in the original work.

- We assume that the parameters given in the instance files start at midnight and that their index represent the beginning of the time slot (the last one is 11.30 pm).
- Each demand process for any given pair of station is independent and can be generated directly from the given Origin-Destination parameter.

6.4 Data Provided and Scenario Generation

The original authors test their method on three North American bike sharing systems : Capital Bikeshare in Washington, Divvy Bikes in Chicago and Hubway (now BlueBikes) in Boston. For each city, two seasons are considered. In order to estimate demands the authors used the method described in (Raviv et al., 2013), the authors collected the weekdays rental and

averaged over 30 min periods. All the riding times were estimated with Google API except for round trips which where set to 30min, and the walking time were set to twice the riding times. The authors did not detail how they solved the censored demand problem, but they described as :

> We estimated the proportion of time a station was empty or full and inflated the demand rates accordingly.

The authors provide on Tal Raviv's website, the data of the six instances. For each instance, the authors of the original paper gives the following data:

- The capacity of each station, C_i,
- Walking and riding times between each pair of stations (i.e., the matrices W and T). This includes the particular case of round-trips (i.e., the values W_{ii} and T_{ii}),
- The rate of the Poisson process generating the trips between pairs of stations (i.e. λ_{ij}^t). These rates are considered constant for each period of 30 minutes. Hence, this data are a succession of matrix $\Lambda^t = (\lambda_{ij}^t)_{i,j}$, one for each period of 30 minutes,
- The initial state found in (Raviv et al., 2013).

Note that for one of the instance (`Hubway.May`), the initial state from (Raviv et al., 2013) is missing. We were therefore unable to reproduce all the simulations for this instance.

6.4.1 Descriptive Statistics of the Data

The data provided by the authors presents some inconsistencies or errors; For one of the instance (`Hubway.May`), the initial state from (Raviv et al., 2013) is missing. We were therefore unable to reproduce all the simulations for this instance. For all the instances, in some stations the incoming demands do not sum to the given return rate of the same stations. There is a number of inactive stations, being stations with arrival and return rates of 0 over all the horizon. This is not per se a problem (the stations can be used to return bikes when the neighbors stations are full) but is a remarkable result in regard of the method used to estimate the data.

We choose to summarize the descriptive statistics of the instances in table 6.1.

Table 6.1: Summary of the data provided in the 6 instances

	Hubway		Capital		Divvy	
	May	August	April	June	August	October
Number of stations*	131	131	232	232	300	300
Number of stations [R]	131	131	232	232	300	300
Avg. rides per day*	3364	4906	7311	8101	4953	5633
Avg. rides per day [R]	3371	4907	7327	8101	4859	5633
Avg rides per planning horizon*	1823	2493	3536	3800	2505	2964
Avg rides per planning horizon [R]	1823	2491	3536	3800	2498	2960
Number of inactive stations [R]	24	17	80	0	16	0

Note:

[R] This reproduction

* Results from Table 2 of Datner et al. (2017)

In Figure 6.2 we observe also the distribution of bikes in the stations given by (Raviv et al., 2013), we can so observe that the given initial stock is really different from a uniform distribution.

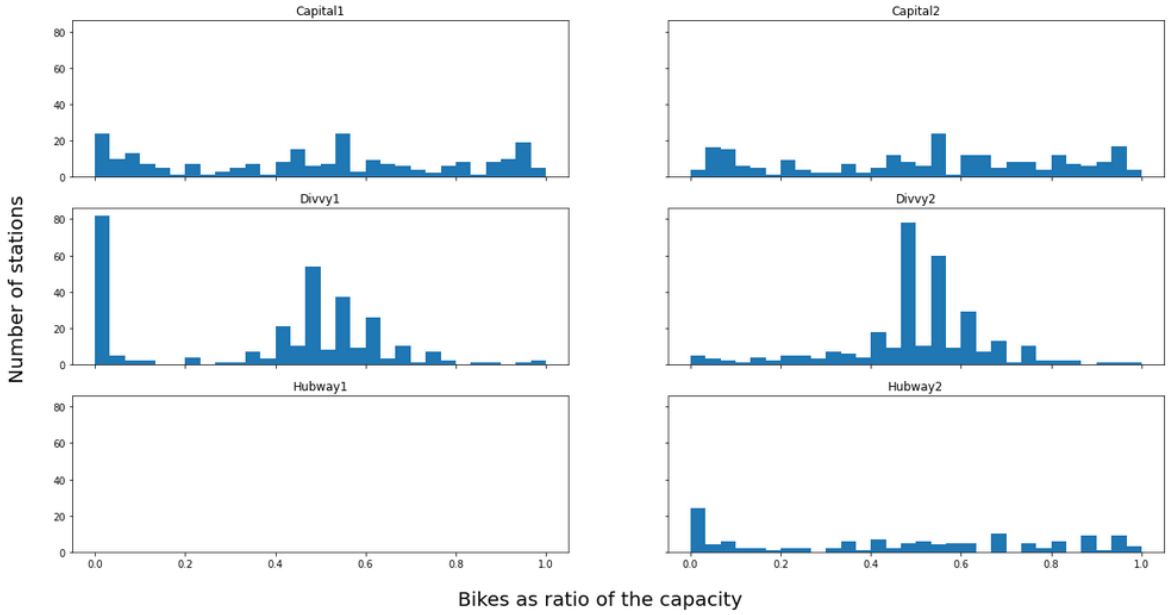


Figure 6.2: Distribution of stations filling as ratio of the total capacity for the different instances

The authors choose to describe the geography of the cities only as riding times, walking times and driving times between stations. This makes hard to link the demand and stock data to geographical information. We see on Figure 6.3 that the three geographies are quite different, for example in Washington the stations are more distant while in Boston the stations are closer to each other.

This corresponds to the results we described in Chapter 4 while studying New-York bike sharing data.

6.4.2 Note on Scenario Generation

For each instance, the rate parameters $\lambda_{ij}(t)$ provided by the authors are considered constant for each 30 minutes time step. For each time step t, origin station i and destination station j we denote $\lambda^t_{i,j}$ the rate of trips between i and j. Those parameters can be stored in a matrix, we call each matrix an instance,

For each instance, we generate a training set of 50 scenarios and a test set of 500 scenarios. The scenarios are generated on a horizon of 9.5 hours starting at 7:00 AM, as explained in the original article. Each scenario corresponds to a list of trips (i, j, t), that are generated according to a Poisson process of time-varying rate λ^t_{od}. To generate trips from the parameters, we consider that all pairs of (origin-destinations) stations are independent. For a given pair (i, j), we generate a Poisson process of time-varying rate λ^t_{ij} following Algorithm 2.

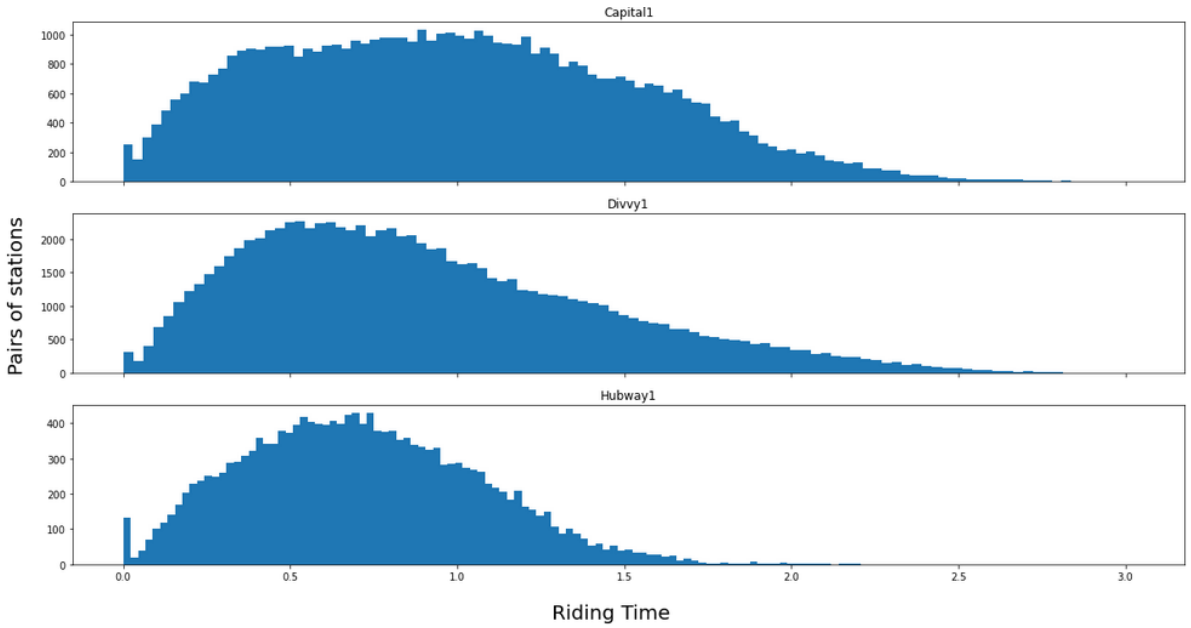


Figure 6.3: Distribution of the riding time between stations

6.5 Optimization Results

In this section, we present the performance that we obtained using our implementation and compare with the results reported in the original paper of (Datner et al., 2017). To obtain the results, we considered the same six original instances. For each instance, and as in the original paper, we generated 500 scenarios that we use for evaluation. We then compute:

- The performance before running the "guided local search" optimization algorithm. This corresponds to the two initial distribution, *Half* or set as in (Raviv et al., 2013).
- The performance after running the two variants of the "guided local search" algorithm. This leads to four sub-cases (two initial conditions plus two heuristics to guide the local search). Note that the guided local search is performed on a set of 50 scenarios that are generated independently of the 500 scenarios that are used to evaluate the policy.

To present a detailed overview of the results, we do as given in the original paper and provide in Table 2 not only the excess time but also other performance indicators:

- Number of ideal rides: Number of users that were able to do their original planned trip,
- Ideal ride ratio: The ratio of ideal rides over the total number of trips,
- Average excess time per non-ideal ride user: The average excess time of the users that did not make their original planned trip,
- Number of shortage and surplus events,
- Average number of shortage and surplus events per non-ideal ride user: The average number of shortage and surplus that a user faces when they did not make their original planned trip.

Note that to produce the results presented below, we made three experiments for each instance

Data: $\lambda_{i,j}^t$ for $t \in \{1 \dots T\}$
next_trip_time := 0
event_list = {}
t := 1
while $t \leq T$ **do**
  if $\lambda_{o,d}^t = 0$ **then**
    t = t+1
  else
    Δ = Generate an exponentially distributed random variable of parameter $\lambda_{o,d}^t$;
    if *next_trip_time* + Δ < *t+1* **then**
      next_trip_time = next_trip_time + Δ
      Append next_trip_time to the event_list
    else
      next_trip_time = t +1
      $t = t + 1$
    end
  end
end

Algorithm 2: Simplified representation of the algorithm used to generate scenarios

(where one experiment corresponds to an independent draw of 500 scenarios for testing and 50 scenarios for the local search). The results of our reproduction are presented with the minimum and maximum of the three experiments.

The original work was coded using MathWorks MATLAB R2011b with Intel Xeon X3450 at 2.67 GHz and 16 GB RAM. All the results presented in this reproduction have been obtained on Python 3.9.7 on an Intel i7-8750H at 2.20GHz with 12 cores and 32 GB RAM.

6.5.1 Performance Before the Local Search Optimization

Table 6.2: Results of initial simulation: Datner et al. versus reproduction

	Capital		Divvy		Hubway	
Method	April	June	August	October	May	August
Excess time (hr/day)						
Half*	274.45	318.03	36.08	75.50	85.57	140.43
Half [R]	255.43-257.95	299.73-301.27	31.18-31.38	63.72-63.83	76.88-77.89	130.25-130.75
R&K*	50.92	47.67	14.57	19.12	22.15	54.48
R&K [R]	59.80-60.63	57.69-58.12	15.04-15.12	22.32-22.38	NA	60.08-60.19
Number of ideal rides						
Half*	2,140	2,269	2,201	2,371	1,321	1,745
Half [R]	2,169-2,169	2,283-2,288	2,195-2,200	2,405-2,406	1,323-1,330	1,748-1,750
R&K*	3,104	3,207	2,385	2,754	1,623	2,119
R&K [R]	3,023-3,026	3,284-3,290	2,353-2,358	2,714-2,714	NA	2,081-2,084
Ideal ride ratio (%)						
Half*	60.50	59.70	87.90	80.00	72.50	70.00
Half [R]	61.24-61.43	60.12-60.19	88.00-88.02	81.25-81.28	72.57-72.81	70.22-70.25
R&K*	87.80	84.40	95.20	92.90	89.00	85.00
R&K [R]	85.44-85.62	86.47-86.53	94.32-94.33	91.66-91.69	NA	83.63-83.66
Avg. number of shortage and surplus events per non-ideal ride user						
Half*	1.50	1.60	1.10	1.30	1.30	1.30
Half [R]	1.48-1.49	1.56-1.56	1.09-1.09	1.24-1.25	1.23-1.23	1.32-1.32
R&K*	1.10	1.20	1.00	1.10	1.10	1.20
R&K [R]	1.13-1.13	1.11-1.12	1.04-1.04	1.10-1.10	NA	1.18-1.18
Number of shortage and surplus events						
Half*	2,137	2,476	334	748	630	998
Half [R]	2,023-2,044	2,364-2,373	326-327	691-692	611-617	980-983
R&K*	476	689	125	232	218	437
R&K [R]	575-584	571-575	148-148	271-273	NA	482-484
Avg. excess time per non-ideal ride user (min)						
Half*	11.80	12.50	7.10	7.60	10.20	11.30
Half [R]	11.22-11.25	11.85-11.90	6.25-6.26	6.88-6.88	9.26-9.32	10.54-10.56
R&K*	7.10	4.80	7.30	5.50	6.60	8.70
R&K [R]	7.01-7.02	6.72-6.75	6.37-6.39	5.40-5.44	NA	8.82-8.83

Note:

Each number represents the average over 500 scenarios, our results are presented as min-max over three experiments.

Note:

[R] This reproduction

* Results from Table 2 of Datner et al. (2017)

In Table 6.2, we report the performance of the two possible initial points (*Half* and (Raviv et al., 2013)). For each instance, we report the numbers from the original paper and the one of our reproduction. We observe a significant difference between our results and the results presented in the original paper. The numbers are not dramatically different but the difference are significant. Note that in all cases, the difference between the minimal and maximal number is small. This suggests that the differences in the results obtained between our reproduction and the original paper is significant and can not be imputed to the variabilility in the scenario generation.

One of the most important difference between the original results and our reproduction is the role of the initial condition. When the stations are filled at half of their capacity, our results show an excess time lower than the one reported in (Datner et al., 2017). This is consistent across instances, and the average difference is 14.2 hour per day. The opposite is true when

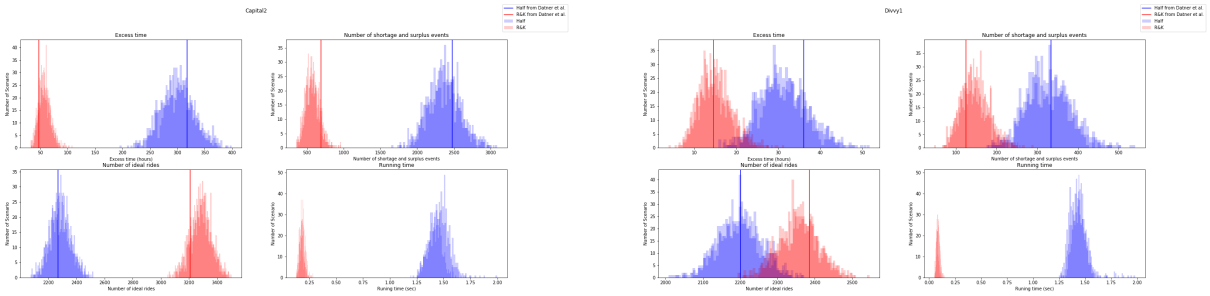
we use the initial stock provided by (Raviv et al., 2013): our reproduction gives a higher excess time compared to the one of the original paper. Again, this difference is consistent acros instances, and the average difference of 5.78 hour per day. Note nevertheless that in all cases, the excess time measure for the initial stock of (Raviv et al., 2013) is lower than the one for the initial stock at *Half* as starting point. This is expected and true for both the results reported in the original paper and for our reproduction.

For the other indicators, the difference between the original value and our reproduction is of the same magnitude. For the *Half* starting point, our simulator outputs more ideal rides, while the excess times per non-ideal ride are lower than the ones of the original authors. Similarly, we observe fewer shortages and surpluses with an average of 105 shortages or surpluses less. For the initial stock given by (Raviv et al., 2013), our simulator outputs less ideal rides while the excess times per non-ideal ride are lower than those of the original authors. Those comparisons are the same on the number of shortages and surpluses.

It must be noted that the average number of shortage and surplus events per non-ideal ride user is not informative, even with significant differences between our solution and the solution proposed by the original authors this indicator presents relatively small differences.

We also inspect the results visually in figures such as 6.4, this visualization allows to observe the distribution of the different indicators over the scenarios. With this observation we can observe that in the majority of the instances, the excess time observed starting from the *Half* initial point is more spread than the excess time obtained starting from the other starting point. With this observation we can conclude that, even if the difference, in average, between the excess time for the (Raviv et al., 2013) distribution and our reproduction is smaller quantitatively than the difference using the *Half* initial distribution, qualitatively (because the (Raviv et al., 2013) tolerates less variability) both represent a real difference in the results. When we observe the distribution of the number of ideal rides over the scenarios, we can see that for the initial point *Half* the difference between the original paper and the reproduction depends highly on the considered instance; for some instances, while existing, this difference can be imputed to a statistical artifact while for others this difference is significant. We can also see that the shape of the distribution of the number of shortage and surplus events over the different scenarios is really similar to the distribution of the excess time. This tends to say, that the number of shortage and surplus will not give more information on the scenarios than the excess time. It is also interesting to note that for all the instances, simulating the 500 scenarios using the Half condition is much faster than simulating the same 500 scenarios with the R&K condition. The simulation time gives an indication on the number of decisions users take during the simulation, indeed each time a user makes a new decision it re-insert the next event of this user into the stack of events which must be emptied. We could so use the simulation time as a performance indicator, firstly because short simulation times implies less energetic consumption for the experiment, secondly because a short simulation time implies an 'easier' scenario for the user which improves the 'user satisfaction'.

In order to find the origin of this difference we looked on the code to ensure we did not have any bug, in the code provided you will find some of those tests. We also contacted the authors who confirmed that we understood correctly their article. All the tests we made were passed correctly and the original authors confirmed that we understood correctly their work,



(a) Capital June

(b) Chicago August

Figure 6.4: Visual representation of simulation results on two instances

we made some minor assumptions on the algorithms which can be found in section 6.3.3. Unfortunately, the authors could not provide us neither their code nor intermediate results that would explain the difference between their results and ours. This difference represents a major gap between our reproduction and the original work.

6.5.2 Performance After the Local Search Optimization

We now consider the performance of the system that is obtained using the initial bike distribution found by the guided local search. In Table 6.3, we present the replication of the Table 2 of the original paper adding the results obtained with our reproduction denoted by [R]. As in the original work, for each instance we report the performance of the best solution among the four cases (2 starting points and 2 guides). This performance is called “Our search” (as in the original paper).

Table 6.3: Results after optimization: Datner et al. versus reproduction

Method	Capital		Divvy		Hubway	
	April	June	August	October	May	August
Excess time (hr/day)						
Our search*	46.08	46.69	14.43	17.35	20.53	49.48
Our search [R]	35.61-36.71	36.29-37.75	12.13-12.43	13.06-13.89	15.34-16.07	33.44-34.63
Excess time reduction vs Half (%)						
Our search*	83.21	85.32	60.00	77.02	76.01	64.77
Our search [R]	85.77-86.16	87.40-87.95	60.39-61.24	78.24-79.53	79.37-80.05	73.41-74.42
Excess time reduction vs R&K (%)						
Our search*	9.51	2.08	0.91	9.23	7.35	9.20
Our search [R]	39.46-40.28	34.55-37.55	17.78-18.94	37.96-41.63	nan-nan	42.47-44.34
Number of ideal rides						
Our search*	3,120	3,377	2,384	2,767	1,644	2,133
Our search [R]	3,191-3,192	3,439-3,445	2,376-2,379	2,776-2,782	1,662-1,673	2,178-2,192
Ideal ride ratio (%)						
Our search*	88.20	88.90	95.20	93.40	90.20	85.60
Our search [R]	90.09-90.26	90.45-90.70	95.06-95.17	93.78-94.00	91.18-91.60	87.53-87.99
Avg. number of shortage and surplus events per non-ideal ride user						
Our search*	1.10	1.10	1.10	1.10	1.10	1.10
Our search [R]	1.07-1.07	1.07-1.07	1.03-1.03	1.04-1.04	1.05-1.05	1.09-1.09
Number of shortage and surplus events						
Our search*	451	459	127	212	197	413
Our search [R]	368-376	378-389	125-128	185-193	162-170	327-340
Avg. excess time per non-ideal ride user (min)						
Our search*	6.60	6.60	7.20	5.30	6.90	8.20
Our search [R]	6.18-6.25	6.14-6.21	6.00-6.04	4.40-4.51	5.94-5.94	6.66-6.66
Running time						
Our search*	2.06	2.08	1.49	1.69	1.02	1.34
Our search [R]	0.23-0.24	0.84-0.82	0.65-0.63	0.75-0.78	0.23-0.24	0.14-0.15

Note:

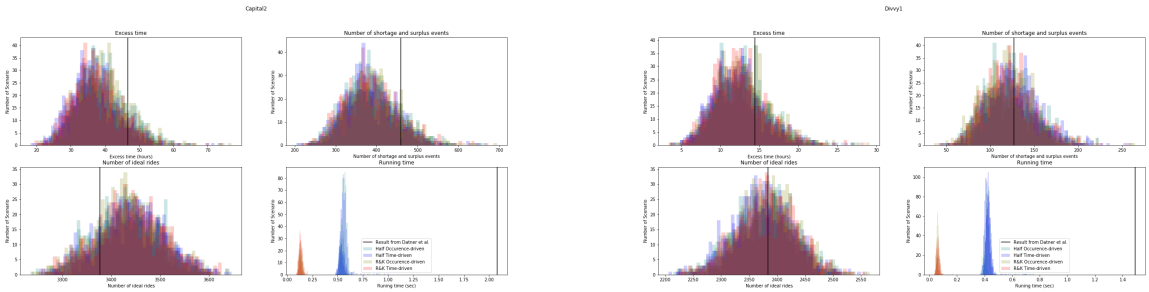
[R] Results obtained in reproduction

* Results from Table 2 of Datner et al. (2017)

The excess times we now obtain after the optimization phase are significantly lower than the results presented in the original work. The reduction is in average 75.6% compared to Half while the authors present a decrease of 62.7% with the same initial point. In comparison to the initial stock given by (Raviv et al., 2013) we again observe a better reduction by far while compared to the original results: in average our reproduction reduces the excess time by 45.5% while the original authors present an average decrease of 6.37%. The other indicators are consistent with this result, our reproduction leads to more ideal rides and the averages of excess time per non-ideal ride are lower. Yet, the number of shortages and surpluses per non-ideal ride provide no relevant information. We are still not able to explain the difference between the results after our optimization phase and those of the original authors. Indeed, since the simulation presented already some difference with the original work we are only able to measure this difference. Nonetheless, although we have a numerical difference with the original work we can conclude that the algorithm proposed gives good quality solutions in term of excess time.

Again we use a visual inspection of the results to detail the optimization result over the scenarios, see Figure 6.5a. In those figures, we see that, for all instance, the results given by the 4 optimizations (2 starting points and 2 guides) give similar results in all the performance indicators. It is nonetheless interesting to note that solutions found by optimizing from the

R&K starting point are much faster than the scenarios found by optimizing the *Half* starting point.



(a) Capital June

(b) Chicago August

Figure 6.5: Visual representation of optimization results on two instances

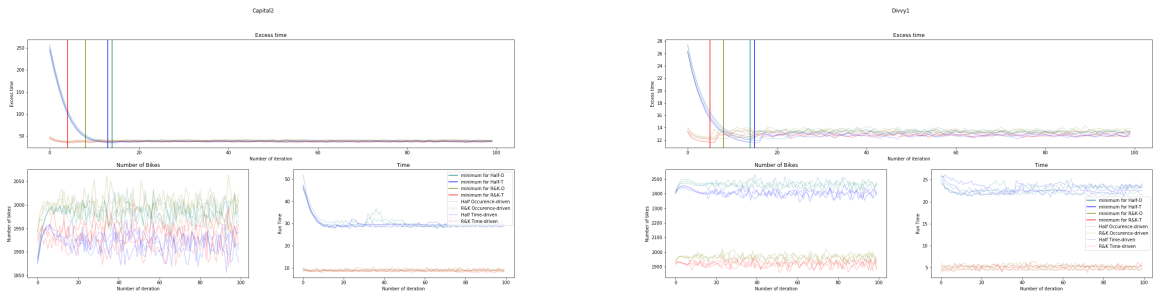
6.6 Evolution of the Optimization Process

As a reminder, the optimization process consists of two steps: one is a simulation of 50 scenarios and the other is the local modification of the stock in order to obtain a better solution. To observe the behavior of the optimization process through the iterations we choose to observe three indicators across the iteration see Figure 6.6a.

- The average excess time over the 50 scenarios,
- The number of bikes in the system,
- The simulation time of the 50 scenarios.

In the publication of (Datner et al., 2017) is only given the total time to make 100 optimization-simulation steps, those number can be found at the end of table 6.3. Concerning this time, we see that the running time of our reproduction is much shorter than the time presented in the original work. This difference is probably due to the choice of the programming language.

On the evolution of the excess time, we see that for a majority of the instances the best solution will be found before the 40th iterations, and that in the rest of the iterations only similar solutions will appear. This tends to say that the optimization time could have been reduced just by doing fewer iterations. When we observe the number of bikes, we see that this number stays in a relatively small interval during all the optimization process, but the behavior depends highly on the instance and on the number of bikes in the initial distribution. In it interesting to note that for some instance (see instance Chicago August of Figure 6.6a) solutions that yield similar solutions on the performance indicators have a very different number of bikes on the system. This could indicate that we lack of an indicator that would differentiate those solutions, indeed having a very different total number of bikes in the system will lead to very different trajectories for the users in the system, and it should appear on the performance indicators. Considering the simulation time, we can see that for all instances, the initial distribution found with an optimization made from the Half starting point are



(a) Capital June

(b) Chicago August

Figure 6.6: Visual representation of evolution of indicators during the optimization process on two instances, are also represented the time when the minimum is found

always longer to simulate than initial distribution founds by optimizing from a R&K starting point. As said previously, simulation time is a way to access the number of decisions users had to make in the scenario. We can so use this other information, to decide which of the solution to choose for a real life implementation.

6.7 Conclusion

We were able to re-code all the algorithms from the scenario generator to the optimization method with the explanations of the publication. We were able to use the generation parameters provided by the authors to re-generate data for the simulation and optimization parts as well. However, it seems that our simulator and the simulator originally coded have differences that lead to numerical differences in the results, before and after optimization. From those differences we could conclude that the reproduction failed as soon as the quantitative differences in the results after the simulation are observed.

On a more positive side, our findings are qualitatively similar to the one of (Datner et al., 2017). Our implementation of the method of (Datner et al., 2017) suggests improvement that are much larger than the one found by the original authors expected (we have a decrease of 45.5% of the objective function while they present a decrease of 6.37%): if one consider that the objective of the publication is not to quantify the efficiency of the method relatively to a particular simulator but rather to evaluate if the proposed method can reduce the excess time, then our findings indeed shows that the method of (Datner et al., 2017) performs well. In this extend, we could argue that our reproduction somehow strengthens the results of (Datner et al., 2017). Nonetheless, failing to reproduce consistently the original work supports the need for more reproducible sciences standards and practices, in order to be able to check step by step all the results to track the origins of discrepancies.

Chapter 7

Simulation

In this chapter we discuss the use of a simulation process in optimization of bike sharing systems. We start by detailing the existing need for evaluation and verification in BSS management strategy optimization, along with the other needs an available simulator could meet. We discuss three verification models chosen from the literature. We then outline requirements a simulator should satisfy, along with the pros and cons of the increasing complexity. The possible trade-offs between generality and simplicity lead us to sketch a modular approach which, we believe, should be used in order to propose a simulator that would be welcome by the optimization community, when sharing their work on BSS. Finally, we choose a simulator to verify the results obtained in Chapter~6. For this comparison, we modify the User Behavior Model (UBM), to one we believe is more realistic. From this case study we conclude that, modifying the UBM impacts the numerical values, but also more importantly, also the definition of the indicators.

7.1 Verification Models

In every computational experiment in optimization, a solution checker is (or should be) used in order to assess the validity and quality of the solutions output by the optimization module(s). In BSS management strategy, we will call such a checker a *Verification Model*.

7.1.1 Need for a Verification Model

Verification Models can be used as modularity tools to meet several needs of the optimization community, in its quest of contributing to coherent and evolving knowledge.

From a reproducibility point of view, a verification model helps make sure that the optimization module(s) actually output what their authors claim they do.

From a performance assessment point of view, a Verification Model allows to run the optimization module an hidden benchmarks, that is, instances different than the ones used by the

designers of the optimization module, so as to limit the risk of overfitting and to assess the robustness of the optimization module.

A Verification Model should be a representation of the physical system in order to verify the constraints of the BSS. It should also be able to compute the chosen performance indicator of the optimized solution. In most of the works, this verification model is often included in the optimization of the solution. In Section~7.1.3, we present three examples of verification models of increasing complexity and abilities.

A verification model is thus voluntarily minimally and mildly defined so as to highlight their similarity rather than their differences.

Nevertheless, when trying to compare different management strategies, such a verification model can not be used as is. Indeed, the underlying goal when comparing different management strategies is to evaluate their performances on real-life systems. The optimization module always relies on some sorts of simplification to allow fast exploration of the strategy space. Thus, the verification model should be able to complement the asumptions of the optimization module, so as estimate the behavior of the system in reality. This need for a verification model has been identified by some authors in the literature (Chemla et al., 2013) but has, to our knowledge, not yet received the attention it deserves.

7.1.2 Requirements of Verification Models

Physical World
The physical world models the studied area along with its components. We find in this description:

- The stations position and capacities,
- The distance and/or time between stations.

User Demand
It describes the will of users to travel from origin to destination stations, as well as when they want to take their trips.

User Behavior Model (UBM)
It defines the choices of the users and how they interact with the system. In particular, their choice when no bike (dock) is found at the origin (destination) station.

Management strategy
It describes how the sytem owner manages the bike fleet movements.

7.1.3 Three Examples of Verification Models From the Literature

Using the optimizer as Verification Model
In (Sayarshad et al., 2012), the authors propose a MILP to solve a Multi-Periodic allocation problem. The inventory Level objective is to maximize the revenue associated to the bike

renting penalized by the cost associated to unmet demand and unused bike. The Bike relocation objective is to minimize the relocation operation cost. Both objective are joined by subtracting the costs from the revenue and solved through classical MILP solving methods.

In this publication we can find a minimal working example of a simulator (used as verification model). The geography of an unoriented graph without costs in which the vertex represents the stations of infinite capacities. With this geography the UBM is each user making one attempt at her origin station, if a bike is available the trip is completed (there is never lack of docks at the return station). If not the user leaves the system. This UBM With this kind of geography and UBM, the management strategies possible are all the static repositioning management strategies (the Multi-Periodic allocation problem is a static repositioning strategy. This constraint is not verified by the Verification Model presented in (Sayarshad et al., 2012) making it not suitable for management strategy comparison.

Co-developing an optimizer and a Simulator
In (Chemla et al., 2013) the authors propose a multi-agent simulator developed in C++. The geography is represented by an oriented multi-graph, the vertices represent the stations associated with their capacity. And for each pair of stations, the values on the arc represent the travel time by car, bike and walk respectively. The authors associate different prices to the users corresponding to the travel options:

- Number of tries for picking a bike,
- Number of tries for leaving a bike,
- Total time spent on finding a bike,
- Total time spent trying to park a bike,
- Total time spent on a bike,
- Total time spent walking.

The users will try to minimize their total price. Even if the authors do not propose to the user to wait at a station, a patience could have been easily added. With such UBM and geography, in addition to the static repositioning strategies, the user-based strategies can be evaluated along with the dynamic management strategies. With this kind of simulator, the main limitations comes from the performance measures that can be computed. The only kind of measures that can be computed are measures that are directly related to the trip (time, number of success) or to the management strategy (cost, greenhouse gas).

Simulator as an independent module
Finally, (Kaziyeva et al., 2021) present a simulator to describe the transport in a city. The geography represents the full city with the different transportation modes and the different activities in the city. From the socio-demographic characteristics of each user, a chain of activities across the city is generated, and the users will try to do all the chain of activities. With this combination of geography and UBM, all the previously mentioned management strategies can be evaluated, but new criterias can be proposed such as the congestion, the accessibility of certain areas or the social impacts of BSS.

7.2 Simulators

A Simulator is a particular case of a Verification Model. In order to compare optimization works, it must be a Verification Model that can be used in the most popular contexts of study. A precise description of the requirements of Simulators together with their relatively good decomposability justifies the modular approach that we advocate for the design of a Simulator that would be used by the optimization community to cross validate their works.

7.2.1 Requirement of Simulators

The requirements presented in this section are designed on the basis of the existing literature.

User Demand Scenarios
In Chapter 3 we explained that, to ensure the comparability of the management strategies, the user demand description must be specified through a common time-stamped format of scenarios. It is thus fundamental that the simulator is suited to take demand scenarios as input.

Physical World
The distance/time between stations should be specified for other modes of transportation, such as walking. Those multi-modal distances/times might not only be required between stations but also between a more general set of origins and destinations. This is required if users require multimodal transportation in large urban area where the the BSS is only available in the center of the city. Different kind of stations must be taken into account by the simulator, finite as well as infinite.

User Behavior Model
A simulator must be adaptable to various UBM. It must be adaptable to different possibilities in most of the cases a combination of few parameters are used:

- Patience of the users to wait for a bike at an empty station,
- Patience of users to wait for a a dock at a full station,
- How users decide to leave the system if they do not find a bike,
- How the users decide where to go if they do not find a bike/dock,
- The reaction of the users to the incentive proposed,
- How the users, make use of the station information (full, empty, etc.).

Informational World
In addition to the requirements of a Verification Model, a simulator must implement an informational world. The informational world represents the information exchange between the users, the system and the management strategy. For instance, in general, the management strategy should not have access to the user demand scenarios (the management strategy can not know the future). But, some management strategies might have access to the past and current state of the system. The informational world also describes more general concepts such as:

- The knowledge of the users on the system (empty/full stations, exact number of bikes, past experience with the system),
- The knowledge of the management strategy on the users (regular or one-time user, trip characteristics),
- The capacity of the user to make a reservation for a bike/dock.

Management Strategies
In order to compare the management strategies presented in Chapter 2, the Simulator should allow to implement them. The static management strategies are the simplest to implement into the simulator since the solution of the optimization can be described through a file. Nonetheless, on special cases of multi-period static relocation, the description of the relocation time between stations can be required from the geography. For dynamic management strategies, the solution of the optimization process is an algorithm. This complexifies the link between the management strategy and the simulator and must be taken into account when designing the simulator. The simulator must thus be easily adaptable to implement the strategy and ensure that the informational world and UBM must be consistent with the strategy. Finally, an adaptation of the UBM make possible the implementation of user-based management strategies.

7.2.2 Modular Design for a Shared Simulator

In the previous Section 7.2.1, we have presented examples for the different requirements of a simulator. But those examples are taken from literature and are not representative neither of the existing literature nor of the future management strategies that could be proposed. Designing an adaptable and open source simulator must hence be a goal. From our review of the literature the more efficient way to match those requirements while keeping the software understandable and adaptable is to use a discrete event simulator. However, the code needed to develop a discrete event simulator is complex and this could limit the re-usability of the simulator. Following our global philosophy to handle complex methods, we argue that a modular approach is the best way to develop a re-usable and maintainable simulator. Such a modular approach is possible since, as we presented in the previous section, the different parts needed for a simulator can be developed independently. This approach could also allow to prospose new features in the simulator, that can be needed for very specific management strategies, but complexify the simulation for the majority of the management strategies. For such features, the modularity could allow to deactivate those features when they are not needed. We think in particular of features such as:

- A geography describing the congestion over the roads, or the socio-economic description of the areas in the city (as in (Kaziyeva et al., 2021)),
- Geography with capacity variation accross the day as done in the New-York system with the vallet stations ("Valet Stations," 2022),
- A UBM that describe the users destination in term of activities and not as stations.

7.2.3 Advantages of Simulator as a Verification Model

Using a simulator could, while answering the need for a simulation model, refine and improve the way BSS management strategies are considered.

Simulation as a way to measure indicators
As mentioned in Chapter 2, when optimizing BSS management strategies, two numerical objective must be chosen. Those objectives, for simplicity reasons are often of similar type (time, money, ...). By using a more precise simulator, new objectives can be used to evaluate a management strategy, we can as example cite (Kaziyeva et al., 2021) who can measure health care access for different populations with their simulator or (Klobucar & Fricker, 2007) who measure real and perceived Bicycle safety.

Simulation as a way to validate hypotheses
As mentioned previously, when optimizing, a verification model should be chosen. This verification model is made under different hypotheses in order to obtain optimized solutions (stations of infinite capacity, equal duration trips, ...). In order to evaluate the relevance of those hypotheses, a simulator could be used to measure the difference of performance with and without those hypotheses.

Simulation as a modularity tool
In the whole thesis we advocate for a highly modular approach to solve complex problems. Indeed, our thought on integrated approaches is that they rather lure us for various reasons (reproducibility, false data, unverified assumptions etc). While seemingly providing better understanding and direct conclusions of the studied system, those conclusions can not be verified easily within the context used is their study. Those conclusions cannot either be generalized to variations of their context (new stations, variations in capacities, number of bikes and demand) nor cross checked by experts of other contexts (cities and models), nor challenged by simplification of hypotheses or optimization methodologies. When studying bike sharing systems from an optimization point of view, many issues are often left as out of the scope of the study because they are too complex to be involved. The point here is not to make an extensive list of all the questions that are relevant for studying bike sharing systems but rather to exemplify how the simulation part of the study could be a way to investigate insights for those issues.
In Chapter 5 we presented the concept of modal share, and made the hypothesis that the modal share is fixed (independent of the system and its management), and thus can be correctly estimated from the historical data. In fact, estimating the modal share is a complex process and the scientific literature is still active on how to make this estimation, concerning the optimization of bike sharing system. Our hypothesis, as the vast majority of the optimization literature, does not take into account the fact that the users will learn the behavior of the system and change their choices depending on the efficiency of the system. Thus, increasing the efficiency of the system could lead to a shift in the modal share (more users, different users) leading to a system with different behavior than the one optimized. From our knowledge, no authors neither on the optimization literature nor in the modal share estimation literature studied this phenomenon. With a simulator it is easier to adapt the work on modal share estimation in order to observe the evolution of the modal share in

response to a new management strategy in the system. Another point that is often ignored in the optimization of bike sharing system is the impact of the system on the agents of the system, in user-based relocation a schedule for operators is defined. This can induce stress on the operators or deteriorate their working conditions. A bike sharing system is also a political process (Sijpesteijn, 2014). A given system can have multiple purposes that are not taken into account in the objective of the optimization. Indeed, a system can be designed to expand the usage of bikes in the city, to offer a replacement for cars in a neighborhood with few public transportation, or to propose a way for tourists to visit a city. All of those issues are studied by diverse scientific communities. It is impossible when trying to solve an optimization problem to be aware of all those political issues, nevertheless they exist and we would benefit from taking them into account. A simulator can be used as an interface between skills and types of expertise in order to identify and integrate various understandings of the studied system.

7.3 Case Study: Starling Simulator

7.3.1 Presentation of the Starling Simulator

The Starling simulator (Leblond et al., 2020) is a discrete event agent-based simulation framework for urban mobility, developed by Tellae. It is released on an open-source license, which is necessary in our work in order to increase the reproducibility potential of our work. Starling is also a versatile simulator (which meets the requirements Section 7.2.1). The agent-based part (in the sense that all the interactions between the agents are defined specifically for each agent) allows a precise observation of the behavior of the system, while the discrete event characteristic allows to have a variable time precision, thus we are able to measure all the indicators we judge necessary while allowing simulations to run quickly on large systems. Last but not least, the Starling code is easily modifiable with a helping development team, so we were able to change the simulator according to our needs. We do not specifically advocate for a usage of Starling as a reference simulator. We rather use it to show the utility and the feasibility of a validating simulator. We also discuss the difficulties we had when trying to implement our results in such a simulator.

7.3.2 Verification Process

As detailed in the first part of this chapter, we use the simulator in order to validate the conclusions obtained in Chapter 6. Obviously, because the simulator is not the same and some hypothesis will change, we don't expect to have the same numerical results, but we test the conclusion:

> On all the instances presented in Chapter 6, the initial inventory obtained by the time-driven method starting with an initial distribution found by (Raviv et al., 2013) results in a lower excess time than all the others tested distribution.

We will also measure the other indicators in order to verify that the evolution of the different indicators evolve with the same intensity with the results found in Chapter 6. By changing the simulator used for the verification we can also verify the improvement made by the optimization process that could help to have a general conclusion on the difference between the results of the original publication and the replication.

7.3.3 Description of the Verification

7.3.3.1 Choices of Simulation Parameters

Even if the Starling simulation is relatively efficient, it is much longer to simulate the system through Starling than using the simulator used for the optimization. We also saw that, testing on less than 500 scenarios already gives a good approximation of the mean on the 500 scenarios. Thus, for each of the instances we choose to simulate 10 scenarios with all the 6 initial bike distributions presented in Chapter 6.
Starling is a discrete event based simulation model, thus the time of the events will be described according to a particular precision. By default, the developers of Starling consider that it is made to simulate at the second level, we choose to keep this default setting. It is important to note that in the reproduction made in Chapter 6 the simulator used is also a discrete event simulator and the simulation were made with a smaller granularity than the simulation made with Starling.

7.3.3.1.1 Changes in the User Behavior Model As in the previous User Behavior Model we make the assumption that all the users try to minimize their total journey time. Thus, after a demand (resp. return) failure, the users will look for the station with available bikes (resp. docks) that would minimize the total travel time. Nonetheless, because the users have the station information when making their decision, we estimated that the users will always use this information. When a user will be picking a bike she will use the station information to decide if she must go to her destination station or to stop in another station that have available docks. We also added patience to the users, it is chosen to be the same for all the users and is set to 5 seconds, the idea behind those 5 seconds is that in a real-context scenario, users that interact with the same station with less than 5 seconds of interval are in sight of each other and thus can wait to pick or drop a bike at the station.

7.4 Effects of the Modification of the User Behavior Model

7.4.1 Changes in the Definitions of the Performance Indicators

In Chapter 6 we proposed different indicators to measure the performance of the initial bike repartition, three of which are direct, in the sense that they are independent of any other

measure and each measure represent an independent indicator. Here we keep only those three indicators as they are enough to fully describe the differences between the two simulators.

Excess Time: The excess time is computed as the difference between the time going directly from origin to destination by bike and the actual time spent by the user in the system. But, the bi-modal geography proposed by (Raviv et al., 2013) does not respect the triangular inequality. Adding this to the fact that the users try to minimize their total travel time, some users will find a shorter way than a going directly for origin to destination thus leading to a negative excess time.

Ideal Rides: The ideal rides are defined in the previous chapter as "The ideal journey from station A to station B is always the one that proceeds via a direct bicycle ride from A to B." This definition does not account for the waiting time that users can have. In the starling simulation, we add a waiting time to take into account interactions between users which does not represent a real dissatisfaction for the users. Nonetheless, to do a fair comparison between the two simulators, we decide to define the number of ideal rides as the number of users that made their trip directly from origin to destination without waiting time.

Number of shortages and surplus (NbSS): The new user behavior model is designed to reduce number of shortage and surpluses. Indeed, always using the station information should reduce the number of failed attempt made by the users, since the users would avoid going to full or empty station. In the previous UBM, the NbSS and the Excess Time measure the same dissatisfaction but, in the new UBM, those two measures represent two different dissatisfaction type since a user could willingly choose to extend their excess-time to avoid a shortage or surplus.

7.4.2 Changes in the Values of the Performance Indicators

We now describe how the new user behavior model impacts the results of the simulation. For a clarity concern, we only present here the Capital April instance. Since the simulation parameters and hypothesis are the same for all the simulations we can expect the other instances to follow the same pattern.

We see in Figure 7.1 that the Excess Time found with the new user behavior model results is relatively close to the excess time originally obtained. As said previously, some users have a negative excess time, but there are exceptions and will not change the average excess time. The reduction of the excess time can be explained by the fact that the users, when checking the station information before taking any decision, anticipate the shortages and surpluses and thus make better decisions.
Concerning the NbSS, the Starling simulation have better performance than the Optimization simulation. This is consistent with the definition of the new UBM and with the result observed in the Excess-Time. It is also interesting to note that since, in the new UBM, the excess Time and the NbSS does not measure the same user dissatisfaction, the reduction of the nbSS is higher than the reduction of the Excess Time.
Finally, in Figure 7.1 we compare the number of non-ideal rides between the two simulations, it appears that the Optimization simulator have a better performance than the Starling

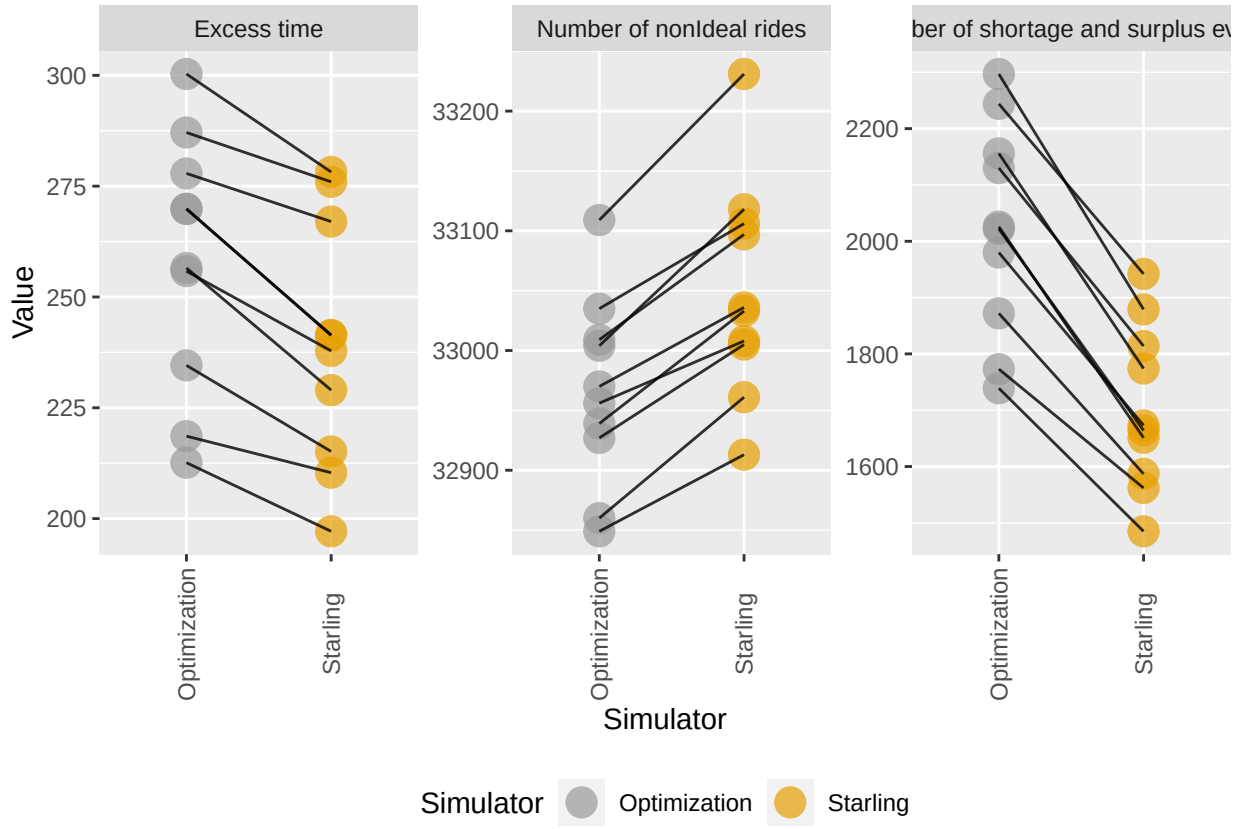


Figure 7.1: Comparison of the performance of the two simulations on the three indicators. Starling have better results on the Excess Time and the NbSS. The Optimization simulator have better results on the number of non ideal rides.

simulator. This is at the opposite of what could be expected from the diminution of the excess time, it is natural to imagine that if the excess time decreases, then number of ideal rides should increase. In fact, because the users use the station information to take their decision, they avoid shortages and surpluses (and thus reduce the excess time) but also deviate more the users from their original trip which reduces the number of ideal rides. Such a result indicates that the correlation of the different indicators rely heavily on the User Behavior-model chosen.

7.4.3 Conclusion on the Modification of the User Behavior Model

The analyses of the results gives two differents results. First, **changing the UBM implies changes in the definition of the indicators.** On the three indicators we chose as main indicators in the previous chapter, when changing the UBM many different definition of the same concept could be implemented. This indicates that when comparing studies the indicators chosen must be precise and defined for different User behavior Models in order to draw coherent conclusions between results. Second, **changing the UBM changes the relation between the indicators**. with the new UBM the NbSS reduces compared to the previous results while the number of non ideal rides also increases between the two simulation.

In (Datner et al., 2017) the authors present those two indicators as correlated, they explain that if the number of shortages and surplus decreases, then the number of non ideal rides would decrease. On a more general perspective, when comparing different optimization methods it is fundamental to find first a common user behavior model. Indeed, comparing results with different UBM could lead to wrong conclusion, both on the efficiency of the management strategy and on the performance of the optimization method.

7.4.4 Performance of the Optimization Methods

In Chapter 6 we obtained different results compared to the results presented by the original authors. Here, we compare the performance of the optimization methods using Starling as the reference simulator. We hope to get insight on the reason of the difference between our solution and the one obtained by (Datner et al., 2017). To ease the comparison, we keep only the Excess Time as performance indicator.

Table 7.1: Comparison after after optimization with three simulators: Original publication, Chapter 6, Starling

	Capital		Divvy		Hubway	
Method	April	June	August	October	May	August
Original Results, extracted from (Datner et al., 2017)						
Half	274.4	318	368	75.5	85.6	140.4
R&K	50.9	47.7	14.57	19.1	22.1	54.5
Search	46.1	46.7	14.43	17.4	20.5	49.5
Reduction vs. Half	83.2	85.3	60	77	76	64.8
Reduction vs. R&K	9.5	2.1	0.91	9.2	7.3	9.2
Results reproduced, extracted from Chapter 6						
Half	258.3	303.4	30.75	64	78.1	128
Search	42.7	46.2	12.17	15.9	20.3	46.6
R&K	60.1	59.8	14.70	22.4	NA	58.2
Reduction vs. Half	83.5	84.8	60.42	75.2	74.1	63.6
Reduction vs. R&K	29	22.8	17.23	29.1	NA	19.9
Results obtained with Starling						
Half	239.3	281.4	23.83	49.7	61.1	103.8
Search	36.2	39.3	8.75	11	14.8	34.7
R&K	52.2	50.9	10.73	16	NA	43
Reduction vs. Half	84.9	86	63.29	77.8	75.7	66.6
Reduction vs. R&K	30.7	22.8	18.44	31.1	NA	19.2

With the results shown in table~7.1 for all the instances, we observe a reduction of the excess time compared to the excess time obtained if no optimization is made. We can also note that the excess time decrease obtained with the optimization when measured with the Starling simulator is comprised between 84% and 19% depending on the instance, even if the number

cannot be strictly compared between the two simulations. This gain is rather in the order of magnitude of the results we found in the reproduction in Chapter 6. This could indicate that the optimization method proposed by the authors have better performances than the original authors expected.

7.5 Conclusion

In this chapter we focused on the importance on the simulation process for comparing management strategies in bike sharing systems. Even if it is not clearly identified by the authors, all the publications use a verification phase. To allow comparison between the various hypotheses of the publications requires a generic simulator. Such simulator can also facilitate the interactions between the optimization community and the other research fields studying bike sharing systems, which in turn can improve the systemic understanding of those systems. To illustrate the importance of this simulation phase we choose to test the results obtained in the reproduction of the work of (Datner et al., 2017) with the Starling simulator. When doing this verification, we changed the User Behavior Model compared to the original work. These changes lead to two different conclusion, the first is that the value of the different indicators are heavily impacted by the User Behavior Model and that all the indicators are not impacted in the same way by the change of User Behavior Model, for example we can have a decrease of the excess time and a decrease in the number of ideal rides, behavior that can not be trivially predicted. The second, and fundamental, conclusion obtained with the results of this new User Behavior Model is that the definition of the indicators depend on the User Behavior Model chosen. This must be taken into account when trying to compare different studies on optimizing bike sharing systems, indeed a definition of the indicators independent of the User Behavior Model must be found, if not the performance and conclusion of a study will be heavily dependent on the User Behavior Model. Finally, we observed the impact of the change of User Behavior Model on the performance of the optimization model presented in (Datner et al., 2017). The performance of the solution found in Chapter~6 are found to be preserved by the change of simulator and User Behavior Model, this can be interpreted as a positive results for the optimization method since this result strengthens the conclusion already obtained in (Datner et al., 2017) and in Chapter 6.

Chapter 8

General Conclusion and Perspectives

8.1 Conclusion on the Work Carried Out

Through the literature review, we have seen that the question of management strategies in BSS has been widely discussed and studied. Several solutions have been proposed by both operators and researchers and, in most systems, management strategies have been put in place. However, few of those management strategies were imagined and studied in the scientific literature, e.g. one may think of the "bike angels" which is an extremely efficient marketing concept.
In this thesis, according to the literature review, we find that the solutions of scientific research lack comparability and reusability. This limits the development and capitalization of knowledge in this filed, since conclusions cannot be re-used from one study to another. To overcome these limits, we have proposed through five chapters (3 to 7) a modular and reproducible methodology for studying and comparing of management strategies. This method is based on the decomposition of existing works into different modules. These modules exist in most of the publications but have not been identified as such in a modular approach. We propose to work on each of these modules independently. With such a modular approach, the objective is to offer the researchers easily reusable solutions to succeed in building a common repository of tools and knowledge. To this end, we have tried to respect several rules of good practice in the design of our methodology:

- The methodology must be simple, otherwise it will not be easily handled.
- The methodology must be general, otherwise it will not be reused or adapted.
- The methodology must be properly specified to ensure the correct sequencing of differents modules.

Through this thesis, in addition to the methodology, a minimal version of each module has been proposed and developed. The first module deals with the reconciliation and usage of real-life data. Since working on synthetic data reduces the scope of the work (and prevents comparability to ground solutions), we favor working on real data. However, those real-life data are often inconsistent between user trips and bike stocks. We propose to solve this issue through a data reconciliation module in which user trip and bike stock data are used

to reconstruct accurate and consistent bike stock data. We are the first, to our knowledge, to make available such a module. Other authors have implemented this kind of module as well but we have not found a re-usable one. We have managed to reconstruct bike stock data consistent with the user trip data. However, on the bike stock data reconciliation process, we have relaxed several constraints and we are still looking for quantifiable performance indicators to evaluate our results.
The second module is on the estimation of the user demand. To make the optimization of management strategies, we cannot use directly the observed demand as an representation of the true demand even touhough we favor using observed data when it is possible. We remind that the observed demand do not contain the censored demand. In our implementation of the second module, we have proposed to use the observed data of the system to estimate the true demand. In the literature, several methods are suggested but no comparative study on the subject is made. Moreover, no code is available on the published methods dealing with the demand estimation. We have proposed to re-code the existing methods and compare them to a method we introduced. We have chosen to focus on a minimal version of the demand estimation problem with only one station. This allows to evaluate the proposed methods and explain their limits. Our proposed estimation method is simple and unbiased on parameters that are time independent. It can be easily adapted to time dependent parameters and provides good results on those experiments. Moreover, from our results, the method we proposed seems rather resilient to the reduction of the number of observations.
The third module of the methodology is the optimization process itself. Being the core module, many proposals have been made in the literature. We propose to use an existing publication as a first example of a basis for comparison. This reproduction is also a case study to test the feasibility of our modular approach and the reproducibility of the contributions of the publication. Thus, we have chosen a publication with clear descriptions of the ideas, correct specification of the algorithms and availability of the data. However, the code is not available. The absence of the original code is enough for us not to have the same results as the authors. This confirms our intuition that, without reproducibility of works we cannot draw conclusions about the state-of-the-art.
The last module we introduced in the methodology is the verification module. It allows us to verify the consistency of the results under various assumptions trying to approximate systems. This module is the final step of the methodology. In fact, it is through this verification module that the comparison between different works can be done. To illustrate the functioning of this module we have chosen to use a third party simulator and to verify the results obtained in the optimization module. Through this example, by varying assumptions on the user behavior model, we discovered that the definintion of indicators are not specified enough. Those problems will recur if we try to compare different research works. Fortunately, none of these problems, questions the results of the optimzation module.

8.2 Perspectives

On the data reconciliation module, it would have been interesting to study the problem through a more formalized model. This would allow to study the complexity of the problem

and assess the performance of the algorithm we have proposed. Our implementation allows to create a first version of this module, makes it available and re-usable. However, even if the problem is interesting and useful, we do not hope that this work will be really resumed. If the data were cleaned and shared by the operators, we could avoid this work. In our opinion, it is rather towards this objective that the community should turn. Nevertheless, even if operators would propose cleaned data, our module could be adapted to assess their quality.
Concerning the demand estimation module, we have worked on the minimal version of the problem with one station and infinite capacity. This makes the work unrealistic. We could have extended the problem to real data with exogenous parameters (weather, season, day of the week, etc.). These extensions are the easy ones to be included. Nevertheless, some other extensions are much more difficult to implement. We think in particular at the problem with several stations and user redirections (users look for bikes/docks at neighbor stations). The complexity of these extensions comes from the fact that the demand on a station could be influenced by the states of neighboring stations. Another possible way to estimate demand would be to start from sociological and urban planning data (Bourgeois, 2019). This could make it easier to build scenarios. It is for this kind of change that the modular methodology takes on its full meaning. Indeed, independently of the way of building the request, it ensures the links with the optimization module.
Regarding the optimization module, it could be interesting to enrich this module with other algorithms. These algorithms could be a reproduction of the literature or new ones. The enrichment of this module could allow the validation of the modular approach in the comparison context.
Regarding the verification module, the choice of the simulator we used, is more a choice of opportunity than an informed one. It would be interesting to push the bibliographical research on these simulators to give a more thoughtful example of usable simulator. However, the simulator we used is developed by a private company and has been proven in real life contexts. Moreover, two major issues that should also be addressed regarding the use of simulation in the verification module: (1) In the example we have shown, the coupling mechanism between an offline optimization and the verification module is straightforward. For online optimization, we have not specified yet the coupling mechanism between optimization and verification modules. A study would be necessary to complete this. (2) We asked the methodology to remain simple. However, a simulator is a complex module, we argued that a modular approach on simulation should keep the module understandable. Nonetheless, a study should be engaged on the way to design a good simulator compromising performance, realism and easy to integrate with the optimization module.
We have seen the limits and upcoming challenges of each module separately. Now we discuss the limits of the methodology.
One missing link is still needed to connect all the modules of our methodology. Our demand estimation module is able to use real an reconciliated data. However, extensions with exogenous parameters need to be added to provide more realistic scenarios to the optimization module.
One of the objectives of the proposal for a modular methodology is to define supports on which different actors of the system can discuss and capitalize knowledges. Throughout this thesis, the collaborations have involved with other researchers and the developers of Starling simulator. The methodology could be enriched with other actors, such as system operators.

Such collaborations have been proposed by the Wien system, the use of their feedback would be particularly useful.
Finally, beyond the comparison of bike sharing systems, more general questions about such systems should be addressed. The arrival of free-floating systems has changed the urban planning issues associated with these systems. On parking issues for instances, to limit the problems of illegal parking, paradoxally stations are now being installed. Furthemore, the environmental impact of these systems is questionable. Motorized bike relocation is often operated, but BSS users mostly replace trips that were made on foot (Ricci, 2015). Thus, installing a BSS in a city increases greenhouse gas emissions and raw material extraction. Moreover, as implemented currently, these systems are very inequal socially in their use. They are use mostly by men from privileged social classes. It raises questions about the purpose of the implementation of such systems and their objectives in the city. The literature on those subjects is scarce. On the reproducibility issue, it would be interesting for the community working the optimization of the BSS management strategies to investigate and adopt best practices from other research communities who have this experience. Our modular approach could be used to gather differents actors and organize discussions and exchanges on these general questions.

References

Albiński, S., Fontaine, P., & Minner, S. (2018). Performance analysis of a hybrid bike sharing system: A service-level-based approach under censored demand observations. *Transportation Research Part E: Logistics and Transportation Review*, *116*, 59–69. https://doi.org/10.1016/j.tre.2018.05.011

Alvarez-Valdes, R., Belenguer, J. M., Benavent, E., Bermudez, J. D., Muñoz, F., Vercher, E., & Verdejo, F. (2016). Optimizing the level of service quality of a bike-sharing system. *Omega*, *62*, 163–175. https://doi.org/10.1016/j.omega.2015.09.007

Angeloudis, P., Hu, J., & Bell, M. G. H. (2014). A strategic repositioning algorithm for bicycle-sharing schemes. *Transportmetrica A: Transport Science*, *10*(8), 759–774. https://doi.org/10.1080/23249935.2014.884184

Arabzad, S. M., Shirouyehzad, H., Bashiri, M., Tavakkoli-Moghaddam, R., & Najafi, E. (2018). Rebalancing static bike-sharing systems: A two-period two-commodity multi-depot mathematical model. *Transport*, *33*(3), 718–726. https://doi.org/10.3846/transport.2018.1558

Ashqar, H. I., Elhenawy, M., Almannaa, M. H., Ghanem, A., Rakha, H. A., & House, L. (2017). Modeling bike availability in a bike-sharing system using machine learning. *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 374–378. https://doi.org/10.1109/MTITS.2017.8005700

Atkins, D. C., & Gallop, R. J. (2007). Rethinking how family researchers model infrequent outcomes: A tutorial on count regression and zero-inflated models. *Journal of Family Psychology*, *21*(4), 726–735. https://doi.org/10.1037/0893-3200.21.4.726

Barth, M. J., Todd, M. D., & Xue, L. (2004). User-Based Vehicle Relocation Techniques for Multiple-Station Shared-Use Vehicle Systems. *Transportation Research Records*, *1887*, 137–144.

Barth, M., & Todd, M. (1999). Simulation model performance analysis of a multiple station shared vehicle system. *Transportation Research Part C: Emerging Technologies*, *7*(4), 237–259. https://doi.org/10.1016/S0968-090X(99)00021-2

BBC. (2018). The problem of China's huge bike graveyards. *BBC News*. https://www.bbc.com/news/av/world-asia-china-43999482

Bike prediction - RStudio :: Solutions. (2022). In *Github*. https://solutions.rstudio.com/example/bike_predict/

Bike share research contributors. (2022). *Bike-share research.* https://bikeshare-research.org/

Boldrini, C., & Bruno, R. (2017). Stackable vs autonomous cars for shared mobility systems: A preliminary performance evaluation. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 232–237. https://doi.org/10.1109/ITSC.2017.8317960

Bordagaray, M., Fonzone, A., dell'Olio, L., & Ibeas, A. (2014). Considerations about the Analysis of ITS Data of Bicycle Sharing Systems. *Procedia - Social and Behavioral Sciences*, *162*, 340–349. https://www.academia.edu/16948364/Considerations_about_the_Analysis_of_ITS_Data_of_Bicycle_Sharing_Systems

Borgnat, P., Abry, P., Flandrin, P., & Rouquier, J.-B. (2009, September). Studying Lyon's Vélo'V: A Statistical Cyclic Model. *ECCS'09*. https://hal-ens-lyon.archives-ouvertes.fr/ensl-00408147

Borgnat, P., Robardet, C., Rouquier, J.-B., Abry, P., Flandrin, P., & Fleury, E. (2011). Shared Bicycles in a City: A Signal Processing and Data Analysis Perspective. *Advances in Complex Systems*, *14*. https://doi.org/10.1142/S0219525911002950

Bourgeois, G. (2019). Les enjeux économiques de la redistribution des véhicules vides dans les services de voitures partagées. *Rencontres de La Mobilité Intelligente*.

Bourgeois, G. (2017). Modélisation des Véhicules Autonomes Partagés. *Internal Document*, 44.

Boyacı, B., Zografos, K. G., & Geroliminis, N. (2015). An optimization framework for the development of efficient one-way car-sharing systems. *European Journal of Operational Research*, *240*(3), 718–733. https://doi.org/10.1016/j.ejor.2014.07.020

Briand, A.-S., Côme, E., El Mahrsi, M., & Oukhellou, L. (2016). A Mixture Model Clustering Approach for Temporal Passenger Pattern Characterization in Public Transport. *International Journal of Data Science and Analytics*, *1*, 37–50. https://doi.org/10.1007/s41060-015-0002-x

Bulhões, T., Subramanian, A., Erdoğan, G., & Laporte, G. (2018). The static bike relocation problem with multiple vehicles and visits. *European Journal of Operational Research*, *264*(2), 508–523. https://doi.org/10.1016/j.ejor.2017.06.028

Caggiani, L., Camporeale, R., Marinelli, M., & Ottomanelli, M. (2019). User satisfaction based model for resource allocation in bike-sharing systems. *Transport Policy*, *80*, 117–126. https://doi.org/10.1016/j.tranpol.2018.03.003

Caggiani, L., Camporeale, R., Ottomanelli, M., & Szeto, W. Y. (2018). A modeling framework for the dynamic management of free-floating bike-sharing systems. *Transportation Research Part C: Emerging Technologies*, *87*, 159–182. https://doi.org/10.1016/j.trc.2018.01.001

Caggiani, L., & Ottomanelli, M. (2012). A Modular Soft Computing based Method for Vehicles Repositioning in Bike-sharing Systems. *Procedia - Social and Behavioral Sciences*, *54*, 675–684. https://doi.org/10.1016/j.sbspro.2012.09.785

Chemla, D., Meunier, F., Pradeau, T., Calvo, R. W., & Yahiaoui, H. (2013, March). *Self-service bike sharing systems: Simulation, repositioning, pricing.* https://hal.archives-ouvertes

.fr/hal-00824078

Chen, L., Zhang, D., Wang, L., Yang, D., Ma, X., Li, S., Wu, Z., Pan, G., Nguyen, T.-M.-T., & Jakubowicz, J. (2016). Dynamic cluster-based over-demand prediction in bike sharing systems. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 841–852. https://doi.org/10.1145/2971648.2971652

Chen, Z., Lierop, D. van, & Ettema, D. (2020). Dockless bike-sharing systems: What are the implications? *Transport Reviews*, *40*(3), 333–353. https://doi.org/10.1080/01441647.2019.1710306

Chiariotti, F., Pielli, C., Zanella, A., & Zorzi, M. (2018). A Dynamic Approach to Rebalancing Bike-Sharing Systems. *Sensors*, *18*(2). https://doi.org/https://doi.org/10.3390/s18020512

Chung, H., Freund, D., & Shmoys, D. (2018). *Bike Angels: An Analysis of Citi Bike's Incentive Program*. 1–9. https://doi.org/10.1145/3209811.3209866

Citibike. (2022). Citi Bike System Data. In *Citi Bike NYC*. http://ride.citibikenyc.com/system-data

Clemens. (2018). Bike Sharing User Communication. In *Mobility as a Service (MaaS)*. https://mobility-as-a-service.blog/bike-sharing-user-communication/

Clemente, M., Fanti, M. P., Mangini, A. M., & Ukovich, W. (2013). The Vehicle Relocation Problem in Car Sharing Systems: Modeling and Simulation in a Petri Net Framework. In J.-M. Colom & J. Desel (Eds.), *Application and Theory of Petri Nets and Concurrency* (pp. 250–269). Springer. https://doi.org/10.1007/978-3-642-38697-8_14

Come, E. (2022). *VLS & Stats*. http://vlsstats.ifsttar.fr/rawdata/

Come, E., Randriamanamihaga, N. A., Oukhellou, L., & Aknin, P. (2014). Spatio-temporal Analysis of Dynamic Origin-Destination Data Using Latent Dirichlet Allocation: Application to Vélib' Bike Sharing System of Paris. *TRB 93rd Annual Meeting*, 20. https://hal.archives-ouvertes.fr/hal-01052951/file/doc00018518.pdf

Contardo, C., Morency, C., & Rousseau, L.-M. (2012). *Balancing a Dynamic Public Bike-Sharing System*. 29.

Coretti Sanchez, N., Martinez, I., Alonso Pastor, L., & Larson, K. (2022). On the simulation of shared autonomous micro-mobility. *Communications in Transportation Research*, *2*, 100065. https://doi.org/10.1016/j.commtr.2022.100065

Correia, G. H. de A., & Antunes, A. P. (2012). 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. https://doi.org/10.1016/j.tre.2011.06.003

Çelebi, D., Yörüsün, A., & Işık, H. (2018). Bicycle sharing system design with capacity allocations. *Transportation Research Part B: Methodological*, *114*, 86–98. https://doi.org/10.1016/j.trb.2018.05.018

Datner, S., Raviv, T., Tzur, M., & Chemla, D. (2017). Setting Inventory Levels in a Bike Sharing Network. *Transportation Science*, *53*(1), 62–76. https://doi.org/10.1287/trsc.2017.07

Dell'Amico, M., Hadjicostantinou, E., Iori, M., & Novellani, S. (2014). The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. *Omega*, *45*, 7–19. https://doi.org/10.1016/j.omega.2013.12.001

Dell'Amico, M., Iori, M., Novellani, S., & Subramanian, A. (2018). The Bike sharing Rebalancing Problem with Stochastic Demands. *Transportation Research Part B: Methodological*, *118*, 362–380. https://doi.org/10.1016/j.trb.2018.10.015

DeMaio, P. (2009). Bike-sharing: History, Impacts, Models of Provision, and Future. *Journal of Public Transportation*, *12*(4), 41–56. https://doi.org/http://doi.org/10.5038/2375-0901.12.4.3

DeMaio, P., & Meddin, R. (2020). *The Meddin Bike-sharing World Map*. https://bikesharingworldmap.com/

Desuilbet, L., Granger, S., Hejblum, B., Legrand, A., Pernot, P., & Rougier, N. (2019). *Vers une recherche reproductible*. https://bookdown.org/alegrand/bookdown/

Di Febbraro, A., Sacco, N., & Saeednia, M. (2012). One-Way Carsharing: Solving the Relocation Problem. *Transportation Research Record*, *2319*(1), 113–120. https://doi.org/10.3141/2319-13

Duthie, J., & Unnikrishnan, A. (2014). Optimization Framework for Bicycle Network Design. *Journal of Transportation Engineering*, *140*(7), 04014028. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000690

Eren, E., & Uz, V. E. (2020). A review on bike-sharing: The factors affecting bike-sharing demand. *Sustainable Cities and Society*, *54*, 101882. https://doi.org/10.1016/j.scs.2019.101882

Fan, W., Machemehl, R. B., & Lownes, N. E. (2008). Carsharing: Dynamic Decision-Making Problem for Vehicle Allocation. *Transportation Research Record*, *2063*(1), 97–104. https://doi.org/10.3141/2063-12

Fernández, A., Billhardt, H., Ossowski, S., & Sánchez, Ó. (2020). Bike3S: A tool for bike sharing systems simulation. *Journal of Simulation*, *14*(4), 278–294. https://doi.org/10.1080/17477778.2020.1718022

Fishman, E., Washington, S., Haworth, N., & Mazzei, A. (2014). Barriers to bikesharing: An analysis from Melbourne and Brisbane. *Journal of Transport Geography*, *41*, 325–337. https://doi.org/10.1016/j.jtrangeo.2014.08.005

Forma, I. A., Raviv, T., & Tzur, M. (2015). A 3-step math heuristic for the static repositioning problem in bike-sharing systems. *Transportation Research Part B: Methodological*, *71*, 230–247. https://doi.org/10.1016/j.trb.2014.10.003

Fricker, C., & Gast, N. (2012). Incentives and Redistribution in Homogeneous Bike-Sharing Systems with Stations of Finite Capacity. *arXiv:1201.1178 [Nlin]*. http://arxiv.org/abs/1201.1178

Gallop, C., Tse, C., & Zhao, J. (2012). A Seasonal Autoregressive Model of Vancouver Bicycle Traffic Using Weather Variables. *Transportation Research 91st Annual Meeting*. http://docs.trb.org/prp/12-2119.pdf

Garcia-Gutierrez, J., Romero-Torres, J., & Gaytan-Iniestra, J. (2014). Dimensioning of a Bike Sharing System (BSS): A Study Case in Nezahualcoyotl, Mexico. *Procedia - Social and Behavioral Sciences*, *162*, 253–262. https://doi.org/10.1016/j.sbspro.2014.12.206

Gast, N., Massonnet, G., Reijsbergen, D., & Tribastone, M. (2015, October). *Probabilistic Forecasts of Bike-Sharing Systems for Journey Planning*. https://doi.org/10.1145/2806416.2806569

Gómez Márquez, H. R., López Bracho, R., & Ramirez-Nafarrate, A. (2021). A simulation-optimization study of the inventory of a bike-sharing system: The case of Mexico City Ecobici's system. *Case Studies on Transport Policy*, *9*(3), 1059–1072. https://doi.org/10.1016/j.cstp.2021.01.014

Guo, Y., Zhou, J., Wu, Y., & Li, Z. (2017). Identifying the factors affecting bike-sharing usage and degree of satisfaction in Ningbo, China. *PLOS ONE*, *12*(9), e0185100. https://doi.org/10.1371/journal.pone.0185100

Haider, Z., Nikolaev, A., Kang, J. E., & Kwon, C. (2018). Inventory rebalancing through pricing in public bike sharing systems. *European Journal of Operational Research*, *270*(1), 103–117. https://doi.org/10.1016/j.ejor.2018.02.053

Hebbert, F. (2022). Citibike-hackers. In *Google Groups*. https://groups.google.com/g/citibike-hackers/about

Ho, S. C., & Szeto, W. Y. (2014). Solving a static repositioning problem in bike-sharing systems using iterated tabu search. *Transportation Research Part E: Logistics and Transportation Review*, *69*, 180–198. https://doi.org/10.1016/j.tre.2014.05.017

Javier, M. (2022). *Bike Sharing Services Predictions*. https://neuralbike.app/

Jian, N., Freund, D., Wiberg, H. M., & Henderson, S. G. (2016). Simulation optimization for a large-scale bike-sharing system. *2016 Winter Simulation Conference (WSC)*, 602–613. https://doi.org/10.1109/WSC.2016.7822125

Justin Tyndall. (2020). The Open Bus. In *The Open Bus*. https://www.theopenbus.com

Kaltenbrunner, A., Meza, R., Grivolla, J., Codina, J., & Banchs, R. (2010). Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive and Mobile Computing*, *6*(4), 455–466. https://doi.org/10.1016/j.pmcj.2010.07.002

Kaspi, M., Raviv, T., & Tzur, M. (2014). Parking reservation policies in one-way vehicle sharing systems. *Transportation Research Part B: Methodological*, *62*, 35–50. https://doi.org/10.1016/j.trb.2014.01.006

Kaspi, M., Raviv, T., Tzur, M., & Galili, H. (2016). Regulating vehicle sharing systems through parking reservation policies: Analysis and performance bounds. *European Journal of Operational Research*, *251*(3), 969–987. https://doi.org/10.1016/j.ejor.2015.12.015

Kaziyeva, D., Loidl, M., & Wallentin, G. (2021). Simulating Spatio-Temporal Patterns of Bicycle Flows with an Agent-Based Model. *ISPRS International Journal of Geo-Information*, *10*(2), 88. https://doi.org/10.3390/ijgi10020088

Kek, A. G. H., Cheu, R. L., & Chor, M. L. (2006). Relocation Simulation Model for Multiple-Station Shared-Use Vehicle Systems. *Transportation Research Record: Journal of the Transportation Research Board*, *1986*(1), 81–88. https://doi.org/10.1177/0361198106198600111

Kek, A. G. H., Cheu, R. L., Meng, Q., & Fung, C. H. (2009). A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, *45*(1), 149–158. https://doi.org/10.1016/j.tre.2008.02.008

Keser, A. (2021). *Extracting and Transforming Citi Bike Data for Analysis*. https://github.com/alhankeser/citibike-analysis

Keser, C. (2017). Using NYC Citi Bike Data to Help Bike Enthusiasts Find their Mates. In *Data Science Blog*. https://nycdatascience.com/blog/student-works/using-nyc-citi-bike-data-help-bike-enthusiasts-find-mate-dating-app/

Klobucar, M. S., & Fricker, J. D. (2007). Network Evaluation Tool to Improve Real and Perceived Bicycle Safety. *Transportation Research Record*, *2031*(1), 25–33. https://doi.org/10.3141/2031-04

Kowald, M., Gutjar, M., Röth, K., Schiller, C., & Dannewald, T. (2022). Mode Choice Effects on Bike Sharing Systems. *Applied Sciences*, *12*(9), 4391. https://doi.org/10.3390/app12094391

Kranish, C. (2021). *Interpolating NYC Bike Share Data to Discover Rebalancing Movements*. https://towardsdatascience.com/interpolating-nyc-bike-share-data-to-discover-rebalancing-movements-6cf8a80eb902

Laporte, G., Meunier, F., & Calvo, R. W. (2018). Shared mobility systems: An updated survey. *Annals of Operations Research*, *271*(1), 105–126. https://ideas.repec.org/a/spr/anopr/v271y2018i1d10.1007_s10479-018-3076-8.html

Leblond, V., Desbureaux, L., & Bielecki, V. (2020). *A new agent-based software for designing and optimizing emerging mobility services : application to city of Rennes*. 16.

Li, A., Gao, K., Zhao, P., Qu, X., & Axhausen, K. W. (2021). High-resolution assessment of environmental benefits of dockless bike-sharing systems based on transaction data. *Journal of Cleaner Production*, *296*, 126423. https://doi.org/10.1016/j.jclepro.2021.126423

Li, X., Ma, J., Cui, J., Ghiasi, A., & Zhou, F. (2016). Design framework of large-scale one-way electric vehicle sharing systems: A continuum approximation model. *Transportation Research Part B: Methodological*, *88*, 21–45. https://doi.org/10.1016/j.trb.2016.01.014

Lin, Y.-K., & Liang, F. (2017). Simulation for Balancing Bike-Sharing Systems. *International Journal of Modeling and Optimization*, *7*(1), 4.

Liu, J., Sun, L., Chen, W., & Xiong, H. (2016). Rebalancing Bike Sharing Systems: A Multi-source Data Smart Optimization. *KDD '16*, 1005–10014. https://doi.org/http://dx.doi

.org/10.1145/2939672.2939776

Liu, X., & Pelechrinis, K. (2021). Excess demand prediction for bike sharing systems. *PLOS ONE*, *16*(6), e0252894. https://doi.org/10.1371/journal.pone.0252894

Loaiza-Monsalve, D., & Riascos, A. P. (2019). Human mobility in bike-sharing systems: Structure of local and non-local dynamics. *PLOS ONE*, *14*(3), 1–17. https://doi.org/10.1371/journal.pone.0213106

Luca Di Gaspero, Andrea Rendl, & Tommaso Urli. (2016). *Balancing bike sharing systems with constraint programming SpringerLink*. https://link.springer.com/article/10.1007/s10601-015-9182-1

Martin, E. W., & Shaheen, S. A. (2014). Evaluating public transit modal shift dynamics in response to bikesharing: A tale of two U.S. Cities. *Journal of Transport Geography*, *41*, 315–324. https://doi.org/10.1016/j.jtrangeo.2014.06.026

Meireles, R., Silva, J., Teixeira, A., & Ribeiro, B. (2013). An e.Bike design for the fourth generation bike-sharing services. *2013 World Electric Vehicle Symposium and Exhibition (EVS27)*, 1–6. https://doi.org/10.1109/EVS.2013.6915040

Mesbah, M., Thompson, R., & Moridpour, S. (2012). Bilevel Optimization Approach to Design of Network of Bike Lanes. *Transportation Research Record*, *2284*(1), 21–28. https://doi.org/10.3141/2284-03

Mukai, N., & Watanabe, T. (2005). Dynamic Location Management for On-Demand Car Sharing System. *Knowledge-Based Intelligent Information and Engineering Systems*, 768–774.

NABSA. (2022). North American Bike Share Association. In *GitHub*. https://github.com/NABSA

Nair, R., & Miller-Hooks, E. (2014). Equilibrium network design of shared-vehicle systems. *European Journal of Operational Research*, *235*(1), 47–61. https://doi.org/10.1016/j.ejor.2013.09.019

Negahban, A. (2019). Simulation-based estimation of the real demand in bike-sharing systems in the presence of censoring. *European Journal of Operational Research*, *277*(1), 317–332. https://ideas.repec.org/a/eee/ejores/v277y2019i1p317-332.html

Nikitas, A. (2019). How to Save Bike-Sharing: An Evidence-Based Survival Toolkit for Policy-Makers and Mobility Providers. *Sustainability*, *11*(11), 3206. https://doi.org/10.3390/su11113206

O'Brien, O., Yu, C., DeMaio, P., Rabello, R., Chou, S., & Benichhio, T. (2021). *The Meddin Bike sharing World Map Mid-2021 Report*.

O'Mahony, E., & Shmoys, D. (2015). Data Analysis and Optimization for (Citi)Bike Sharing. *Proceedings of the AAAI Conference on Artificial Intelligence*, *29*(1). https://ojs.aaai.org/index.php/AAAI/article/view/9245

Pal, A., & Zhang, Y. (2017). Free-floating bike sharing: Solving real-life large-scale static rebalancing problems. *Transportation Research Part C: Emerging Technologies*, *80*, 92–116.

https://doi.org/10.1016/j.trc.2017.03.016

Pan, Y., Zheng, R. C., Zhang, J., & Yao, X. (2019). Predicting bike sharing demand using recurrent neural networks. *Procedia Computer Science*, *147*, 562–566. https://doi.org/10.1016/j.procs.2019.01.217

Pfrommer, J., Warrington, J., Schildbach, G., & Morari, M. (2013). Dynamic Vehicle Redistribution and Online Price Incentives in Shared Mobility Systems. *IEEE Transactions on Intelligent Transportation Systems*, *15*. https://doi.org/10.1109/TITS.2014.2303986

Rainer-Harbach, M., Papazek, P., Hu, B., & Raidl, G. R. (2013). Balancing Bicycle Sharing Systems: A Variable Neighborhood Search Approach. In M. Middendorf & C. Blum (Eds.), *Evolutionary Computation in Combinatorial Optimization* (pp. 121–132). Springer. https://doi.org/10.1007/978-3-642-37198-1_11

Randriamanamihaga, A., Come, E., Oukhellou, L., & Govaert, G. (2013). *Clustering the Velib' origin-destinations flows by means of Poisson mixture models*. 273–278. https://hal.archives-ouvertes.fr/hal-00863151

Raviv, T., Tzur, M., & Forma, I. A. (2013). Static repositioning in a bike-sharing system: Models and solution approaches. *EURO Journal on Transportation and Logistics*, *2*(3), 187–229. https://doi.org/10.1007/s13676-012-0017-6

Repoux, M., Kaspi, M., Boyacı, B., & Geroliminis, N. (2019). Dynamic prediction-based relocation policies in one-way station-based carsharing systems with complete journey reservations. *Transportation Research Part B: Methodological*, *130*, 82–104. https://doi.org/10.1016/j.trb.2019.10.004

Ricci, M. (2015). Bike sharing: A review of evidence on impacts and processes of implementation and operation. *Research in Transportation Business & Management*, *15*, 28–38. https://doi.org/10.1016/j.rtbm.2015.03.003

Romero, J. P., Moura, J. L., Ibeas, A., & Alonso, B. (2015). A simulation tool for bicycle sharing systems in multimodal networks. *Transportation Planning and Technology*, *38*(6), 646–663. https://doi.org/10.1080/03081060.2015.1048946

Rudloff, C., & Lackner, B. (2014). Modeling Demand for Bikesharing Systems: Neighboring Stations as Source for Demand and Reason for Structural Breaks. *Transportation Research Record Journal of the Transportation Research Board*, *2430*, 1–11. https://doi.org/10.3141/2430-01

Rybarczyk, G., & Wu, C. (2010). Bicycle facility planning using GIS and multi-criteria decision analysis. *Applied Geography*, *30*(2), 282–293. https://doi.org/10.1016/j.apgeog.2009.08.005

Saltzman, R., & Bradford, R. (2016). Simulating a More Efficient Bike Sharing System. *Simulating a More Efficient Bike Sharing System*, *Journal of Supply Chain and Operations Management*, 36–47.

Sayarshad, H., Tavassoli, S., & Zhao, F. (2012). A multi-periodic optimization formulation for bike planning and bike utilization. *Applied Mathematical Modelling*, *36*(10), 4944–4951. https://doi.org/10.1016/j.apm.2011.12.032

Schlote, A., Chen, B., & Shorten, R. (2015). On Closed-Loop Bicycle Availability Prediction. *IEEE Transactions on Intelligent Transportation Systems*, *16*(3), 1449–1455. https://doi.org/10.1109/TITS.2014.2365492

Schuijbroek, J., Hampshire, R. C., & Hoeve, W. J. van. (2017). Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research*, *257*(3), 992–1004. https://doi.org/10.1016/j.ejor.2016.08.029

Sean, B. (2020). General Bikeshare Feed Specification (GBFS) [Github]. In *NABSA*. https://github.com/NABSA/gbfs/blob/master/gbfs.md

Shaheen, S., Nelson, C., Apaar, B., & Cohen, A. (2015). *Shared Mobility: Definitions, Industry Developments, and Early Understanding Transportation Sustainability Research Center*. https://tsrc.berkeley.edu/publications/shared-mobility-definitions-industry-developments-and-early-understanding

Shu, J., Chou, M. C., Liu, Q., Teo, C.-P., & Wang, I.-L. (2013). Models for Effective Deployment and Redistribution of Bicycles Within Public Bicycle-Sharing Systems. *Operations Research*, *61*(6), 1346–1359. https://doi.org/10.1287/opre.2013.1215

Shui, C. S., & Szeto, W. Y. (2020). A review of bicycle-sharing service planning problems. *Transportation Research Part C: Emerging Technologies*, *117*, 102648. https://doi.org/10.1016/j.trc.2020.102648

Sijpesteijn, B. (2014). *Political Factors in Bicycle Sharing Systems*. http://mobility-workspace.eu/wp-content/uploads/Political_Factors_in_Bicycle_Sharing_Systems.pdf

Singla, A., Santoni, M., Bartok, G., Mukerji, P., Meenen, M., & Krause, A. (2015). *Incentivizing Users for Balancing Bike Sharing Systems*. 7.

Sohn, K. (2011). Multi-objective optimization of a road diet network design. *Transportation Research Part A: Policy and Practice*, *45*(6), 499–511. https://doi.org/10.1016/j.tra.2011.03.005

Szeto, W. Y., Liu, Y., & Ho, S. C. (2016). Chemical reaction optimization for solving a static bike repositioning problem. *Transportation Research Part D: Transport and Environment*, *47*, 104–135. https://doi.org/10.1016/j.trd.2016.05.005

Tang, Y., & Dai, B.-R. (2018). A Partial Demand Fulfilling Capacity Constrained Clustering Algorithm to Static Bike Rebalancing Problem. *Advances in Data Mining. Applications and Theoretical Aspects*, 240–253. https://doi.org/10.1007/978-3-319-95786-9_18

Teodorović, D., & Dell'Orco, M. (2005). Bee colony optimization - A cooperative learning approach to complex transportation problems. *Advanced OR and AI Methods in Transportation*, 51–60.

Valet Stations. (2022). In *Citi Bike Help*. https://help.citibikenyc.com/hc/en-us/articles/360032341751-Valet-Stations

Vogel, P., Greiser, T., & Mattfeld, D. C. (2011). Understanding Bike-Sharing Systems using Data Mining: Exploring Activity Patterns. *Procedia - Social and Behavioral Sciences*, *20*,

514–523. https://doi.org/10.1016/j.sbspro.2011.08.058

Wang, B., & Kim, I. (2018). Short-term prediction for bike-sharing service using machine learning. *Transportation Research Procedia*, *34*, 171–178. https://doi.org/10.1016/j.trpro.2018.11.029

Wang, J., Tsai, C.-H., & Lin, P.-C. (2016). Applying spatial-temporal analysis and retail location theory to public bikes site selection in Taipei. *Transportation Research Part A: Policy and Practice*, *94*, 45–61. https://doi.org/10.1016/j.tra.2016.08.025

Waserhole, A., & Jost, V. (2012). *Vehicle Sharing System Pricing Regulation: A Fluid Approximation*. https://hal.archives-ouvertes.fr/hal-00727041

Waserhole, A., & Jost, V. (2016). Pricing in vehicle sharing systems: Optimization in queuing networks with product forms. *EURO Journal on Transportation and Logistics*, *5*(3), 293–320. https://doi.org/10.1007/s13676-014-0054-4

Waserhole, A., Jost, V., & Brauner, N. (2013). *Vehicle Sharing System Optimization: Scenario-based approach*. https://hal.archives-ouvertes.fr/hal-00727040v4/file/Waserhole_VSS_ScenarioBasedApproach_May2013.pdf

Winters, M., Hosford, K., & Javaheri, S. (2019). Who are the "super-users" of public bike share? An analysis of public bike share members in Vancouver, BC. *Preventive Medicine Reports*, *15*, 100946. https://doi.org/10.1016/j.pmedr.2019.100946

Yan, S., Lu, C.-C., & Wang, M.-H. (2018). Stochastic fleet deployment models for public bicycle rental systems. *International Journal of Sustainable Transportation*, *12*(1), 39–52. https://doi.org/10.1080/15568318.2017.1324586

Yoon, J. W., Pinelli, F., & Calabrese, F. (2012). Cityride: A Predictive Bike Sharing Journey Advisor. *2012 IEEE 13th International Conference on Mobile Data Management*, 306–311. https://doi.org/10.1109/MDM.2012.16

Zhang, D., Yu, C., Desai, J., Lau, H. Y. K., & Srivathsan, S. (2017). A time-space network flow approach to dynamic repositioning in bicycle sharing systems. *Transportation Research Part B: Methodological*, *103*, 188–207. https://doi.org/10.1016/j.trb.2016.12.006

Abstract

Bike Sharing Systems (BSSs) are nowadays installed in many cities. In such a system, a user can take any available bike and return it to wherever there is an available parking spot. The Operations Research literature contains many papers that study optimization questions related to BSS, and in particular how to maximize the availability of bikes where and when the users need them. Yet, the optimization methods proposed by these papers are difficult to compare because most papers use their own problem instances and define their own metrics. This thesis aims to fill this gap by building a reproducible research methodology for BSSs. In this work, we divide this methodology in four modules: use of historical data, demand estimation, optimization methods, and performance evaluation. We study each module separately. In each case, we propose a prototype implementation and compare existing solutions when they are available.

The first module handles the use of data from real systems. For many systems, two types of data are usually available: trips made by users, and records of the number of bikes available in each station. We observe that in general these data are inconsistent and we propose a method to correct this and detect relocation operations. The second module is the demand estimation one. In optimizing a BSS, it is essential to estimate the demand of the users for whom the system is designed. Most of the optimization works in the literature use historical demand to estimate the demand of the system. We experiment with the few existing methods of the literature along with a newly introduced method to detect censored demand. The third module is bike availability optimization. We implement a published optimization algorithm for this module as an example. We illustrate the challenges of reproducible research by trying to replicate the results. This chapter shows that, although the original authors made the data about their experiments available, we did not get the same quantitative results as the original publication. This difference highlights the need for better publication standards to produce more reproducible results. Finally, our fourth and last module is used to validate the optimization methods implemented in the 3rd module. We advocate that a simulator having all the requirements (user behavior models, demand scenarios, management strategies, etc.) can be a validation model. We use a third-party simulator to illustrate this module.

We observe throughout this thesis that making research reproducible is not always handled with due diligence while being fundamental to produce valuable knowledge. In this work, we try our best efforts to specify and provide reproducible tools to ensure that researchers could obtain the same results with the same data. We give links to the data, codes, environments and analyses needed to reproduce the experiments.

Keywords: *Bike Sharing Systems, Demand Estimation, Optimization-Simulation, Reproduction, Modular Approach, Comparison*

Résumé

Les systèmes de vélos en libre-service (VLS) sont aujourd'hui installés dans de nombreuses villes. Dans un tel système, un utilisateur peut prendre n'importe quel vélo disponible et le rendre là où il y a une place de parking libre. La littérature de recherche opérationnelle contient de nombreux articles qui étudient les questions d'optimisation liées aux SBS, et en particulier maximiser la disponibilité des vélos où et quand les utilisateurs en ont besoin. Cependant, les méthodes d'optimisation proposées par ces articles sont difficiles à comparer car la plupart des articles utilisent leurs propres instances de problèmes et définissent leurs propres métriques. Cette thèse vise à combler cette lacune en construisant une méthodologie de recherche reproductible pour les VLS. Dans ce travail, nous divisons cette méthodologie en quatre modules : utilisation de données historiques, estimation de la demande, méthodes d'optimisation et évaluation des performances. Nous étudions chaque module séparément. Dans chaque cas, nous proposons un prototype d'implémentation et comparons les solutions existantes lorsqu'elles sont disponibles.

Le premier module traite de l'utilisation de données provenant de systèmes réels. Pour de nombreux systèmes, deux types de données sont en général disponibles : les trajets effectués par les utilisateurs, et les enregistrements du nombre de vélos disponibles dans chaque station. En général ces données sont incohérentes, nous proposons une méthode pour corriger cela et détecter les opérations de relocalisation. Le deuxième module est l'estimation de la demande. Pour optimiser un VLS, il est essentiel d'estimer la demande des utilisateurs pour lesquels le système est conçu. La plupart des travaux d'optimisation de la littérature utilisent la demande historique pour estimer la demande du système. Nous expérimentons les quelques méthodes de la littérature existantes ainsi qu'une méthode nouvellement introduite pour détecter la demande censurée. Le troisième module est l'optimisation de la disponibilité des vélos.A titre d'exemple, nous re-implémentons un algorithme d'optimisation publié. Nous illustrons les défis de la recherche reproductible en essayant de reproduire les résultats. Ce chapitre montre que, même si les auteurs originaux ont mis à disposition un grand nombre de données sur leurs expériences, nous n'avons pas obtenu les mêmes résultats quantitatifs que la publication originale. Cette différence souligne la nécessité d'améliorer les normes de publication afin de produire des résultats plus reproductibles. Enfin, notre quatrième et dernier module est utilisé pour valider les méthodes d'optimisation implémentées dans le 3ème module. Nous considérons qu'un simulateur ayant toutes les exigences (modèles de comportement des utilisateurs, scénarios de demande, stratégies de gestion, etc.) peut être un modèle de validation. Nous utilisons un simulateur tiers pour illustrer ce module.

Nous avons observé tout au long de cette thèse que la reproductibilité des recherches n'est pas toujours traitée avec la diligence requise alors qu'elle est fondamentale pour produire des connaissances. Dans ce travail, nous nous efforçons de spécifier et de fournir des outils reproductibles afin de garantir que des chercheurs puissent obtenir les mêmes résultats avec les mêmes données. Nous fournissons des liens vers les données, les codes, les environnements et les analyses nécessaires à la reproduction des expériences.

Mot-clés: *Systèmes de vélo en libre service, Estimation de la demande, Optimisation-Simulation, Reproduction, Approche modulaire, Comparaison*