CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Integrating Resource Management in Service Network Design for Bike Sharing Systems

Bruno Albert Neumann-Saavedra
Teodor Gabriel Crainic
Bernard Gendron
Dirk Christian Mattfeld
Michael Römer

March 2018

CIRRELT-2018-17

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Integrating Resource Management in Service Network Design for Bike Sharing Systems

Bruno Albert Neumann-Saavedra[1], Teodor Gabriel Crainic[2,3,*], Bernard Gendron[2,4], Dirk Christian Mattfeld[1], Michael Römer[5]

[1] Decision Support Group, University of Braunschweig, Mülehnpfordtstraβe 23, 38106 Braunschweig, Germany

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[3] Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

[4] Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

[5] Institute of Business Information Systems and Operations Research, Martin Luther University, Halle-Wittenberg, Universitätsring 3, 06108 Halle (Saale), Germany

Abstract. Station-based bike sharing systems rely on an intra-day redistribution process to provide bikes for rentals and free racks for returns whenever demanded by users. Tactical planning of bike sharing systems aims to answer service level expectations of users through a cost-efficient use of redistribution vehicles. In this paper, we propose a novel service network design formulation that coordinates redistribution and vehicle routing decisions in space and time, thus producing master tours. This formulation explicitly integrates resource management decisions into master tours by including both the rational use of limited resources as well as an accurate time representation of pickups and deliveries of bikes at station visits. We propose a matheuristic which relies on neighborhood search to obtain good-quality solutions for large problem instances in reasonable time. To produce starting solutions, we propose a construction heuristic. The construction heuristic decomposes the redistribution process into three phases: determine pickup and delivery operations, link pickups and deliveries of bikes into transport services, and assign transport services to master tours. Our computational study provides evidence that the proposed solution methodology produces solutions of high quality for real-world sized problem instances. Finally, we evaluate the trade-of between resource usage and service level expectations.

Keywords: Bike sharing systems, intra-day bike redistribution, service network design, matheuristic.

Acknowledgements. This research has been supported by the German Research Foundation (DFG) through the Research Training Group SocialCars (GRK 1931). The focus of the SocialCars Research Training Group is on significantly improving the city's future road traffic, through cooperative approaches. This support is gratefully acknowledged. Partial funding for this project also comes from the Discovery Grant and the Discovery Accelerator Supplements Programs of the Natural Science and Engineering Research Council of Canada (NSERC), and the Strategic Clusters program of the Fonds québécois de la recherche sur la nature et les technologies (FQRNT). The authors thank the two institutions for supporting this research. While working on this project, the second author was Adjunct Professor with the Department of Computer Science and Operations Research, Université de Montréal.

---

* Corresponding author: teodorgabriel.crainic@cirrelt.net

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2018

1 Introduction

Municipalities are striving towards enhancing urban mobility with environmental and territorial impact as small as possible (Mrkajić and Anguelovski, 2016). Station-based bike sharing systems (BSSs) have emerged as an inexpensive and sustainable complement to traditional public transport (Ricci, 2015). BSSs saturate urban areas with stations providing two assets: bikes for rentals and racks for returns. Users can thus perform one-way bike trips between any station pair. In order to encourage a high use of bikes during the day, these bike trips are typically free within a short time span, e.g., the first half hour (Martinez et al., 2012). In particular, BSSs provide a suitable solution for mid-term distances as well as for the "first and last mile" between transport hubs and points of interest (Kumar et al., 2016).

The long-term viability of a BSS strongly depends on the service level offered to users. The service level reports how many users perform trips as demanded. Providing a high service level is challenging because the provision of bikes is subject to the number of bike racks at stations. In this paper, we define a station's fill level as the ratio between the number of provided bikes and its capacity. Unfortunately, one-way trips in combination with spatial and temporal characteristics of mobility patterns lead to stations that regularly run full or empty during the day (Datner et al., 2017). One important mobility pattern is induced by commuters who ride to working districts in the morning and to residential areas in the afternoon (O'Brien et al., 2014). In the districts, users compete for the bikes and free racks depending on the time of day. Asset shortages lead to failed requests if either a user cannot rent a bike at an empty station, or if a user cannot return a bike to a full station.

To mitigate the number of failed requests, operators employ resources such as vehicles and driving staff for redistributing bikes among stations (Shu et al., 2013). In the redistribution process, operators pick up and deliver bikes at stations with shortages of assets. We call these pickup and delivery operations fill level alterations (FLAs). Each FLA involves a bike volume and a time window for visiting the respective station. Most state-of-the-art studies addressing the redistribution process focus on the overnight case, assuming that users do not request assets within the redistribution process (Raviv et al., 2013). Although the overnight redistribution process is interesting from an academic perspective, in practice it does not suffice to significantly reduce failed requests over the course of the day (de Chardon et al., 2016). Therefore, this work focuses on intra-day redistribution process, where both users and vehicles move bikes within the underlying station network.

An efficient intra-day redistribution process relies on insights about mobility patterns to anticipate when and where users demand assets. Several research studies provide evidence that mobility patterns are stable and predictable (O'Brien et al., 2014; Vogel et al., 2011). For example, stations close to working districts typically suffer from several

(a) morning (b) evening

Figure 1: Number of bikes in racks at stations in New York's Citi Bike, reflecting mobility patterns in a summer season's weekday (http://bikes.oobrien.com/)

return requests in the morning and rental requests in the evening. The opposite behavior is typically observed at stations close to residential districts. These mobility patterns lead to accumulations of bikes in different districts of the city according to the time of day. For New York's City Bike BSS, Figure 1 shows the number of bikes in racks at stations in the morning (on the left chart) and evening (on the right chart) for one summer weekday. Each circle depicts a single station: the bigger the circle, the more bikes in racks at the respective station. We observe that bikes are accumulated in different areas in the morning and evening, coinciding with the working and residential districts, respectively. Actually, we would note a similar distribution for every weekday over a season.

This regularity of mobility patterns provides the basis for tactical planning in the redistribution process. Tactical planning aims to meet service level expectations in a cost-efficient way. To this end, tactical planning determines regular master tours for daily redistribution operations. We believe that tactical planning for bike redistribution comes with several advantages. Exploiting mobility patterns should reduce pitfalls in redistribution operations, e.g., placing bikes where they are not requested. Operators should become more efficient over time as they gain experience and knowledge in performing the regular tours, compare Bard and Jarrah (2009). Besides, tactical planning should seemingly increase the reliability of service since users learn when and where operators increase the asset provision of stations.

Achieving a suitable trade-off between costs and service level involves a complex synchronization of redistribution and vehicle routing decisions in space and time that we can model as a service network design problem. The resulting service network design for bike sharing system (SNDBSS) determines 1) FLAs counteracting expected user demand, 2) transport services pairing pickups with deliveries of FLAs, and 3) master tours sequencing FLAs to implement the selected transport services within the underlying station infrastructure.

Operators must allocate resources to implement master tours at the operational phase. Therefore, integrating resource management decisions in SNDBSS is crucial to obtain reliable master tours that require few adjustments within the operational phase. Resource management decisions consider the rational use of vehicles with respect to given budget limitation as well as the necessary handling time to perform pickup and delivery operations. According to de Chardon et al. (2016), operators spend a significant part of their workday in handling bikes. Therefore, a master tour does not only consider a time-ordered sequence of station visits but also indicates handling time at each of such visits. Although the relevance of resource management suggests itself, state-of-art formulations addressing the tactical planning process of BSSs either neglect or make simplified assumptions of at least one of these decisions. In fact, de Chardon et al. (2016) put emphasis on the minor success of such formulations in practice as operators outperform them by their own collected experience. This clearly reflects the need for more research efforts to integrate resource management at the tactical planning level.

The contribution of this work is twofold. First, we introduce a new and comprehensible formulation of SNDBSS which integrates resource management decisions at the core of the tactical planning process. The SNDBSS is formulated as a time-expanded network describing redistribution and vehicle routing decisions. This time-expanded network takes the form of a mixed-integer programming (MIP) problem. Second, we present a solution methodology to deal with real-world sized problem instances of the MIP problem. This methodology relies on a neighborhood search based matheuristic to produce solutions of near-optimal quality. The matheuristic carefully chooses and fixes variables associated with vehicle routing decisions thus defining reduced MIP problems which can be solved using exact methods. To speed up the search for high-quality solutions, we propose a construction heuristic for feasible starting solutions of reasonable quality. The construction heuristic breaks down the decision-making process into three steps: the generation of FLAs, the pairing of FLAs, and the sequencing of FLAs. While the construction heuristic drastically reduces computational effort, it also suggests a reasonable number of redistribution vehicles. Computational experiments provide evidence about the efficiency of the proposed solution methodology.

The paper is organized as follows. Section 2 introduces business models adopted for most BSSs. In Section 3, we describe the scope of the tactical planning process for BSSs. Section 4 briefly reviews the existing literature in bike redistribution and service network design. In Section 5, we introduce the time-expanded network formulation. The solution methodology is presented in Section 6. In Section 7, we report on the computational experiments. Finally, conclusions are given in Section 8.

2 Business Models

In particular, BSSs adopt user-based business models which have a major impact on the redistribution process. Since most BSSs aim to enhance mobility in urban areas instead of generating profits, bike rentals are typically inexpensive or even free for short travel times. These scarce revenue obtained through bike rentals limit the budget for bike redistribution.

The business model establishes both the organizational and contracting structure of a BSS (Institute for Transportation and Development Policy, 2014). The organizational structure defines tasks and responsibilities of each entity involved in the project. The contracting structure formalizes the relationships between such entities. In particular, the organizational structure is conformed of the municipality, contractors, and operators. The municipality is the public entity that supports the planning and implementation of a BSS. Contractors are the entities which participate in the ownership, financing, and operation of a BSS. Finally, the operator is the entity responsible for daily operators.

The contracting structure specifies which entities own the assets of the BSSs and which entities are responsible for operations. Typically, BSSs are owned by municipalities but are operated by private operators. According to Institute for Transportation and Development Policy (2014), a major part of BSSs follows this contracting structure. Hence, municipalities do not depend on third parties to make decisions, while they delegate critical processes (like bike redistribution) to efficient operators. We note that the contracting structure influences the objectives of a BSS: while public owners aim at enhancing public transport, private owners usually follow profit-oriented basis. In this paper, we focus on the public-ownership-and-private-operated contracting structure.

In principle, the municipalities and operator negotiate the monetary budget to perform redistribution as well as service level expectations. Although this negotiation process is out the scope of this paper, it is clear that the redistribution process suffers from a trade-off between the achieved level of service and the given budget. If the budget is scarce, operators should focus on high-demanded stations where each redistribution operation has a major impact on the level of service. By enlarging the budget, operators may increase their operational flexibility by addressing low-demanded stations as well.

In Figure 2, we illustrate aspects influencing the bike redistribution process. Increasing the monetary budget permits more redistribution as operators can acquire more vehicles. In turn, this permits to increase the asset provision at stations. Thus, the service level is the result of successful and failing user requests given the asset provision. Although It would be expected that a high level of service leads to more revenues they are scarce in BSSs because of most trips are free-of-charge. Finally, the revenues can be invested in additional redistribution resources.

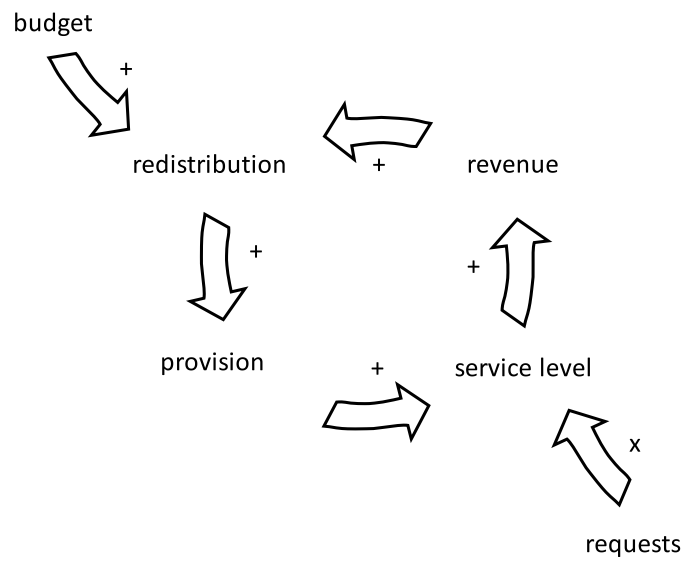


Figure 2: Aspects influencing the redistribution process.

3 Tactical Planning Process

The redistribution process involves logistical challenges associated with trade-offs among stakeholder objectives related to resource usage and service level. Therefore, we consider convenient to frame tactical planning decisions in management processes which aid the decision-making in bike redistribution. We identify four processes: data acquisition as well as demand, service, and resource management. In the following, we sketch each of these management processes.

3.1 Data Acquisition

Tactical planning relies on a suitable prediction of expected demand to produce reliable master tours. We understand expected demand in terms of users' desired trips between stations during the day. Predicting expected demand is nowadays possible by analyzing operational data recorded by information systems (O'Brien et al., 2014; Vogel et al., 2011). Operational data include the asset provision at each station and time of day, together with recorded bike trips characterized by rental and return stations with their respective timestamps (see Table 1). Research studies provide evidence that the spatial and temporal characteristics of trips strongly depends on mobility patterns. Mobility patterns answer when, where, and for which purpose users perform trips. Trip purposes include commuting, tourist, leisure, errands, sport, and so on. Weather conditions have an influence on trips since users prefer cycling in warm and dry days (Corcoran et al., 2014; Faghih-Imani et al., 2014). To generate input data for tactical planning, we aggregate recorded trip data into ride flows as an abstract representation of user's desired trips.

Table 1: Exemplary bike trip data.

id rental station	id return station	rental timestamp	return timestamp
78	176	08 Jun 2015 19:51	08 Jun 2015 20:00

We recognize some limitations in deriving expected demand from operational data. Since only successfully rental and return requests are recorded, information about censored demand is not available, e.g., we do not know if a user roams to another station if the targeted station is empty or full. Although more research efforts and data sources are necessary to yield a more reliable representation of expected demand, this is out the scope of this paper. In this paper, we assume that expected demand derived from operational data is a suitable input for the tactical planning process.

3.2 Demand Management

Demand management aims to synchronize expected user demand with bike redistribution at stations. Expected user demand is used to predict shortages of assets over the day. Demand management determines the necessary FLAs to counteract the expected user demand. Demand management also designs time windows where FLAs should be performed. Time windows tell limited daily periods where FLAs have a major impact in reducing failed requests at the targeted stations. Certainly, the time window length strongly depends on the station's fill level. On the one hand, an FLA should occur before the station becomes full or empty to avoid failed requests. On the other hand, a time window aims to avoid pitfalls by visiting a station too early, where the target bike volume to pick up or delivery is not yet available. In addition, time windows impact on the operational flexibility to coordinate station visits within the day.

The upper chart of Figure 3 illustrates the demand management process at a single station. The solid line shows the number of bikes in racks over time if no redistribution is performed. Notice that at some point in time this station becomes imbalanced after several return requests. An FLA picks up bikes to avoid failed return requests, see the dashed line. The given time window suggests the earliest and latest time where this FLA should take place. If the FLA occurs later than the given time windows, there exists failed return requests because of a full station. If the FLA is performed before this time window, it will lead to an insufficient bike volume for avoiding that the station becomes full later on. In the same way, a delivery FLA is depicted for the second station in the lower chart of Figure 3.

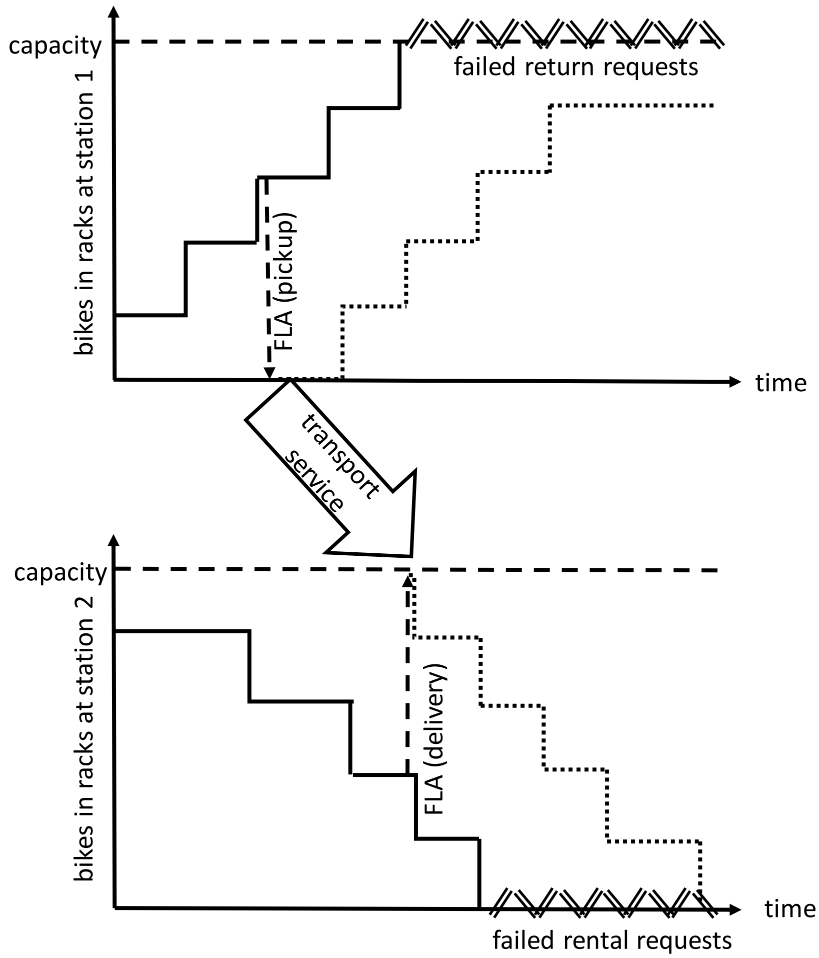


Figure 3: Redistribution process between two stations.

3.3 Service Management

Service management deals with spatial and temporal interdependencies between stations within the redistribution process. Notice that bike redistribution takes place in a closed system wherein users and vehicles can move a single bike several times per day. Hence, bikes that are picked up at one station must be delivered to another station later on. Service management relies on an assignment problem to determine time-dependent transport services, thus pairing pickups and deliveries of bikes.

The interplay of upper and lower charts of Figure 3 illustrates the service management process at two stations with monotone demand patterns. Here, the solution is to pick up bikes at station 1 and deliver them to station 2. We observe interdependencies regarding the time, space as well as and bike volume. On the one hand, if the pickup of bikes occurs too early at station 1, the redistribution volume does not suffice to significantly reduce failed rental requests at station 2. On the other hand, delivering bikes too early at station 2 is not possible due to insufficient free racks. This is clearly a trivial example: in real-world sized BSS, the number of stations is usually in the hundreds while monotone demand patterns at stations are rarely observed. Hence, determining adequate transport services in practice is challenging.

3.4 Resource Management

This management process aims to use resources in a cost-efficient manner. Since resource management permits the implementation of transport services within the operational process, its integration at the tactical planning is of utmost importance. Notice that the redistribution process is one of the most costly operations in BSSs (Intelligent Energy Europe, 2011).

Resource management considers fixed costs for acquiring vehicles as well as fuel costs for operating them. Driving staff is paid according to either the working time or the number of bikes redistributed per day (de Chardon et al., 2016). Each vehicle is assigned to a master tour. A master tour is a time-ordered sequence of stations visited, starting and ending at the depot. Apart from the necessary driving time for station visits, considering handling time for picking up or delivering bikes is crucial to obtain reliable master tours. Notice that driving staff spends a significant part of their workday on handling operations at each station visit (de Chardon et al., 2016). The task of generating master tours can be addressed as a pick-up and delivery problem where service times are related to the duration of handling operations at stations.

3.5 Interdependencies within Management Processes

The management processes comprising tactical planning of BSSs are strongly interwoven. This implies that decisions determined at one management process may have a major impact on the other processes. Figure 4 shows the interdependencies between management processes. Each square frames a management process. An arrow relates a decision with two management processes: each arrow comes from the management process where the respective decision takes place and ends at the management process which is affected by this decision. Ovals depict synchronization concerns within pairs of management processes.

The relation between demand and service management mainly concerns with time synchronization. The design of time windows for performing FLAs plays a fundamental role. The tighter the time windows, the more lack of synchronization between the pickup and delivery FLAs are expected since there is less operational flexibility for establishing transport services. Enlarging time windows increases the operational flexibility at the

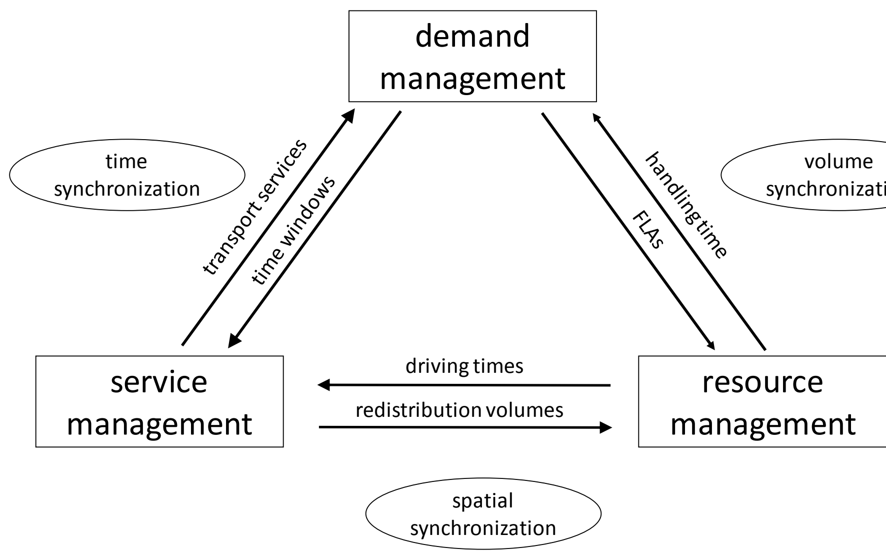


Figure 4: Interdependencies of redistribution decisions between management processes.

expense of reducing the bike volume to pick up or deliver at FLAs.

The relation between demand and resource management mainly impacts on volume synchronization. Given certain provision of resources, adding FLAs to the redistribution plan leads to less time for handling bikes because vehicles must spend more time for visiting stations. Therefore, rationalizing the number of FLAs is crucial for an efficient use of resources.

The interdependencies between service and resource management are described in terms of space synchronization. A transport service with its respective redistribution volume implies that a vehicle drives between two stations. This means that the distance covered by transport services must be reasonable to avoid excessive driving time.

To produce a reliable redistribution plan, we must consider all these aspects when designing a formulation of SNDBSS. In the next section, we show that existing formulations neglect or made simplifying assumptions to represent these aspects and their interdependencies. In addition, we give a sound basis for catching the redistribution aspects into a service network design problem.

4 Literature Review

First, we review the literature on bike redistribution in BSSs. Then, we highlight the role of service network design in transport planning as well as current advances to integrate resource management decisions.

4.1 Redistribution in Bike Sharing Systems

In recent years, shared mobility systems have been a source of intensive research. Laporte et al. (2015) provide a rich review of related literature over the past ten years on car- and bike sharing systems. These studies include size and location of stations, number of shared vehicles in the system as well as redistribution to mitigate failed requests.

We notice that a major part of these studies focuses on overnight redistribution, where the demand management is omitted since user demand is assumed as negligible when vehicles are operating (Raviv et al., 2013). In overnight redistribution, the main goal is to achieve target station's fill levels. There is a wide range of solution methods addressing formulations of the static case including among others neighborhood-search based schemes (Rainer-Harbach et al., 2015), tabu search (Ho and Szeto, 2014), exact methods based on Bender cuts (Erdoğan et al., 2015), and decomposition schemes (Forma et al., 2015). The reader is referred to Espegren et al. (2016) for a more extensive literature review. Nevertheless, overnight redistribution does not suffice to achieve a high quality of service. In practice, high-demanded BSSs require redistribution over the day (de Chardon et al., 2016).

In comparison with the static case, there exist few studies addressing intra-day redistribution. Although this paper focuses on the tactical planning process, we recognize that bike redistribution constitutes an operational process as well. In the operational process, short-term decisions are made in a dynamic and stochastic environment, where vehicle routing decisions and individual trips must be accurately represented in time and space. Instead of tactical planning, operational decisions are subject to day-to-day information with respect to weather as well as traffic and special events.

Regarding literature addressing the operational process, Brinkmann et al. (2015) propose a single-vehicle, stochastic inventory routing problem, where the objective is to minimize the gap between the number of bikes in racks at stations and a certain safety stock of assets. Pfrommer et al. (2014) investigate the trade-off between the costs of offering incentives for returning shared vehicles at low-demanded stations and the costs of redistributing bikes. Kloimüllner et al. (2015) introduce a cluster-first route-second approach with the strong assumption that operators only address full and empty stations. The approach relies on a Benders decomposition scheme. The master problem takes the form of an assignment problem that links stations with vehicles, whereas subproblems take the form of traveling salesman problems. Solutions of the subproblems are used to generate Benders cuts for the master problem. Zhang et al. (2017) introduce a non-linear formulation to take redistribution decisions based on a short-term forecast of user demand. The authors linearize the underlying formulation for addressing large problem instances. In addition, they propose a methodology which solves the linear relaxation in a first step and solves set covering problems to assign given vehicles to routes in a second step. This paper also recognizes the existence of pitfalls in short-term decisions because of forecasting errors. These errors are reduced by embedding the formulation into a rolling horizon approach. Although research efforts can be devoted to mitigate these pitfalls in the operational process, we believe that the generation of redistribution plans at the tactical planning is highly beneficial.

Existing tactical formulations make simplifying assumptions with respect to one or several management processes that were highlighted in Section 3. For example, Contardo et al. (2012) present a MIP formulation over a time-expanded network, where nodes encode station and time period. Each of such nodes involves a demand in terms of rental or return requests. Service management is neglected because the MIP formulation assumes that bikes are artificially added or removed from the BSS if assets do not suffice to meet demand at nodes. The objective function minimizes such artificial operations, whereas costs related to resource management are not taken into account. To address the formulation, Dantzig-Wolfe and Benders decomposition is used to derive lower and upper bounds.

Kloimüllner et al. (2014) adapt the formulation of the static case presented in Rainer-Harbach et al. (2015) in order to cope with intra-day redistribution at the tactical planning. Given a certain provision of vehicles as input data, the objective function relies on different weights to minimize (1) failed requests, (2) the difference between achieved and final number of bikes in racks at stations at the end of the redistribution time, (3) driving time, and (4) redistributed bikes. Resource management decisions associated with driving and handling time are not accurately represented since the workday is discretized into hourly time periods.

Vogel et al. (2014) propose a resource allocation problem that produces time-dependent transport services with the aim of minimizing shortage of assets at stations. The underlying MIP formulation neglects resource management since it does not take vehicle routing decisions into account. Therefore, the resulting redistribution plan suffers from a lack of synchronization of the transport services (Neumann-Saavedra et al., 2015).

Brinkmann et al. (2016) propose a multi-vehicle inventory routing problem assuming perfect demand knowledge over the target planning horizon. The authors propose the notion of fill level intervals at stations. A fill level interval indicates a desired number of bikes in racks at the respective time. They assume that such intervals are given by external decision support systems. Costs associated with resource management are not taken into account. The aim of redistribution bikes is to minimize the gap between actual fill levels and the desired intervals. Solution methodology combines spatial and temporal decomposition of redistribution decisions. At each time period, the network infrastructure is geographically partitioned, where each partition is served by one vehicle.

In Angeloudis et al. (2014), the redistribution plan is produced in two steps. The first step yields tours by solving a multi-vehicle traveling salesman problem. Then, the second step relies on a flow assignment model for determining redistribution volumes. The goal is to achieve a target number of bikes at racks at stations as close as possible. Regarding resource management, this work does not consider an accurate representation of handling time, whereas redistribution costs are not taken into account.

4.2 The Role of Service Network Design in Transport Planning

Service network design has its roots in the tactical planning of transport carriers. To be competitive in the growing market, less-than-load carriers must transport goods such that they ensure a reliable service while rationalizing transport activities. As the volume of goods requested by a customer is typically small with respect to the capacities of vehicles, consolidated transport is crucial to operate in a cost-efficient manner. The role of service network design is to coordinate consolidated transports in space and time. The result is a service network, that is, services that route demand among a network of terminals according to policies and guidelines established at the tactical planning (Crainic and Laporte, 1997). Service network design have been used in maritime transport (Christiansen et al., 2007), rail transport (Cordeau et al., 1998), intermodal transport (Crainic and Kim, 2007), and freight transport (Crainic, 2000), among others.

Implementing the service network requires resources like vehicles, drivers, and power units. Therefore, integrating resource management decisions in service network design have become an issue of utmost relevance in the research community. In recent years, service network design has been extended to coordinate vehicle movements with the service network (Andersen et al., 2009) and to recognize that resources to route demand at terminals are limited and involve fixed costs (Crainic et al., 2014b). Integrating these resource-management decisions to service network design leads to more consistent plans which require minor adjustments within the operational process.

Time-expanded network formulations are extensively used to represent time-dependent decisions. In this way, terminals are replicated once per each discretized time period, whereas flows connect these time-dependent nodes. Hence, nodes encode terminal and time period, whereas flows encode decisions with respect to the service network and use of resources. This time-expanded network typically takes the form of MIP formulations which, in most cases, are computationally intractable using exact methods. Sophisticated solution methods are necessary to produce solutions of good quality in a reasonable runtime. The literature reports solution methodologies based on tabu search (Pedersen et al., 2009; Vu et al., 2013), cutting-plane methods (Chouman and Crainic, 2014), decomposition schemes (Teypaz et al., 2010), and scope scaling (Crainic et al., 2014a), among others.

In this paper, we consider a service network design problem to support the bike sharing redistribution process. We follow advances in transport planning to adequately depict master tours by means of a time-expanded network formulation. Advances in transport planning lay the foundations to integrate resource management decisions in SNDBSS. For the intra-day redistribution process, however, we require a more sophisticated time-expanded network formulation to accurately represent management processes at the tactical planning level.

5 Time-Expanded Network Formulation

We embed the decision-making process of bike redistribution in a time-expanded network formulation. We start defining notation for the physical nodes (vehicle depot and stations) of the BSS infrastructure. Let $S_0 = S \cup \{0\}$ be the set of physical nodes. The set $S = \{i\} = \{1, 2.., |S|\}$ represents stations, whereas the physical node $\{0\}$ represents the vehicle depot. Each station $i \in S$ has a capacity of c_i racks. The vehicle depot has no capacity to place bikes, meaning that $c_0 = 0$.

We state that the redistribution process occurs within a defined workday length, which is divided into chronologically and equal-length time periods $\mathcal{T} = \{t\} = \{0, 1, ..., \mathcal{T}\}$. The workday length is cyclic assuming that time period 0 follows time period $\mathcal{T}$.

We define the time-expanded network $\mathcal{G}$ as follows. We represent each station $i \in S$ and time period $t \in \mathcal{T}$ by two node types: 1) the rack node $(i, t) \in \mathcal{N}_S$, and 2) the vehicle node $\widehat{(i, t)} \in \mathcal{N}_V$. A rack node encodes the number of bikes in racks at the respective station and time period. A vehicle node encodes the number of loaded bikes in a vehicle if it parks at the respective station and time period. Then, we also replicate the depot over the time periods. The joint of the rack node set $\mathcal{N}_S$ with the depot replications in time represents the node set $\mathcal{N}_0$. With this notation, we can depict the interplay between vehicles and stations in a clear and consistent manner.

Let b denote the size of the bike fleet. We assume that the same number of bikes are placed at stations both at the beginning and end of the workday length. In time period 0, empty vehicles park in the depot. Within the workday length, the number of bikes in racks and vehicle at the respective nodes are updated at the end of each time period according to the inflows and outflows of bikes.

Table 2: Flow definition in the time-expanded network.

source node	destination node	flow layer	process	decision
vehicle	vehicle	redistribution	service management	transport service
rack	rack	vehicle	resource management	master tour
rack	vehicle	redistribution	resource management	(pickup) FLA
vehicle	rack	redistribution	resource management	(delivery) FLA

Let $\mathcal{A}$ represent the set of links connecting nodes of the time-expanded network. We characterize each link $a \in \mathcal{A}$ by a origin-destination node pair. In this way, links encode time-dependent flows. These flows are grouped in three layers representing aspects associated with demand, service, and resource management. In this work, they are called ride, vehicle, and redistribution flow layer, respectively. Table 2 shows the definition of each flow according to source and destination nodes. In the following, we describe each flow layer in more detail.

Ride Flow Layer consists of flows between rack nodes, where each ride (bike) flow encodes expected demand. Notice that a ride flow unit obeys to an abstract representation of a user's desired trip at the demand management process. Let $\mathcal{A}_U$ be the ride link set. A ride link $a \in \mathcal{A}_U$ indicates the existence of ride flow units ζ_a. These ride flows are given by input data.

Vehicle Flow Layer connects rack nodes depicting master tour decisions associated with the resource management process. The vehicle link set $\mathcal{A}_V$ permits decisions with respect to vehicle movements and duration of station visits within the workday length. This set is comprised by the vehicle movement link set $\mathcal{A}_{VM}$, the holding link set $\mathcal{A}_{HO}$, and the handling link set $\mathcal{A}_{HA}$. A vehicle movement link $e \in \mathcal{A}_{VM}$ allows the decision that one vehicle moves between two nodes. A holding arc link $e \in \mathcal{A}_{HO}$ permits that one or more vehicles park at the depot during one time period. A handling link $e \in \mathcal{A}_{HA}$ permits that one loaded vehicle stays at a station to either pick-up or deliver bikes during one time period. The vehicle flows are scheduled such that they represent feasible master tours.

Redistribution Flow Layer encodes either transport services between two vehicle nodes, or FLAs as an interplay between rack and vehicle nodes (pickup) or vice versa (delivery). In this way, this layer is associated with service and resource management decisions. We denote the redistribution link set by $\mathcal{A}_R$ to represent bike transportation and handling operations. A transportation link $a \in \mathcal{A}_T$ permits the movement of a certain bike volume only if the corresponding transport service is set. A transportation link connects two vehicle nodes from different stations to move the redistribution volume

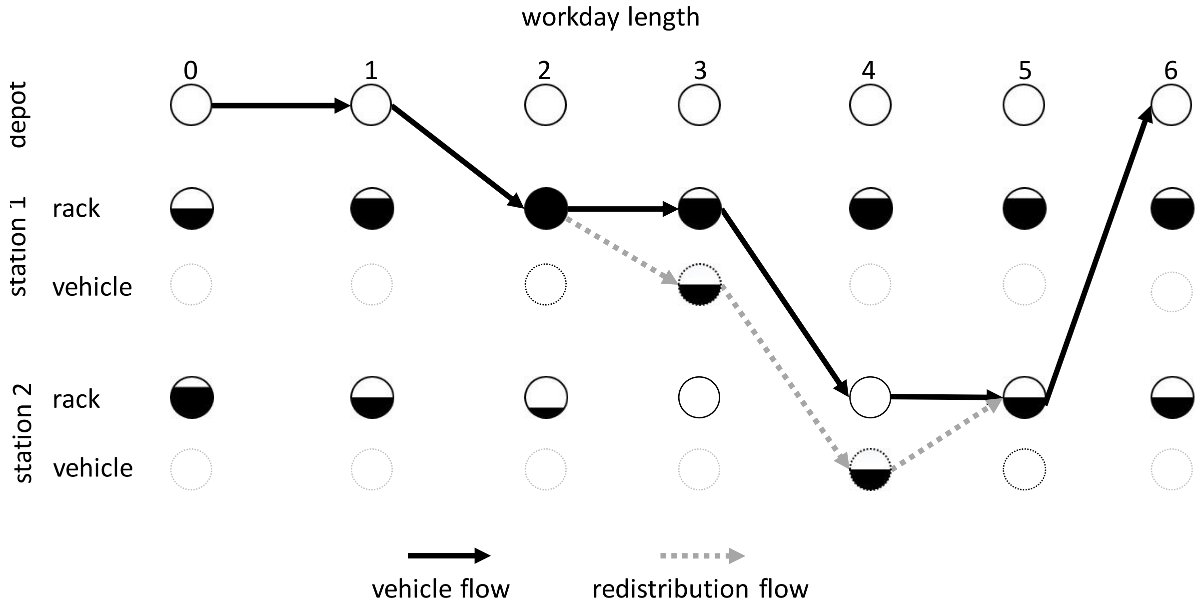


Figure 5: The cyclic time-expanded network of SNDBSS.

from one station to another. Handling operations are depicted by links representing the interplay between rack and vehicle nodes. To this end, we define the pick-up link set $\mathcal{A}_P$ and the delivery set $\mathcal{A}_D$. If a handling link $e \in \mathcal{A}_{HA}$ is set, either the corresponding pick-up link $a \in \mathcal{A}_P$ allows pick-up flow units of bikes from a rack node to a vehicle node during one time period, or the delivery flow of bikes is permitted by the corresponding delivery link $a \in \mathcal{A}_D$ from a vehicle node to a rack node. We denote the pick-up and delivery link associated by a handling link e by $\mathcal{P}(e)$ and $\mathcal{D}(e)$, respectively.

Notice that pickup and delivery activities must not simultaneously occur if the corresponding handling link is set. If the formulation considers handling costs in the objective function, simultaneous pickup and delivery activities will never occur in an optimal solution. Nevertheless, if handling costs are excluded from the objective function, the topology of the search space must generate consistent solutions with respect to such a behavior as well. Therefore, to obtain the desired behavior regardless of the objective function, the formulation presented below considers the binary variable $\alpha_e \in \mathcal{A}_{HA}$. This binary variable equals one if pick-up bike flows though $a \in \mathcal{P}(e)$ are permitted, otherwise the delivery of bikes though $\mathcal{D}(e)$ is permitted.

In Figure 5, we show a time-expanded network with two stations and seven time periods. The y-axis represents physical nodes of a BSS infrastructure, whereas the x-axis represents the workday length by time periods. The black fill of each node depicts the station's fill level or vehicle load, accordingly. Hence, a completely white node means that the station (or vehicle) is empty, whereas a completely black node tells that the station is full. We observe that stations 1 and 2 becomes respectively full and empty due to ride flows that are not shown for the sake of illustration. A master tour is depicted

Table 3: Variables and parameters for the MIP formulation of SNDBSS.

variable	type	description
v	integer	vehicle fleet size
x_a	real	bike flow units though link $a \in \mathcal{A}$
z_a	real	unmet ride flow units though link $a \in \mathcal{A}_U$
y_e	integer	vehicle flow units though link $e \in \mathcal{A}_V$
I_i^t	real	bikes at rack node $(i, t) \in \mathcal{N}_S$
ω_i^t	real	bikes at vehicle node $(\hat{i,t}) \in \mathcal{N}_V$
α_e	binary	one if pick-up flow unites are permitted though link $a \in \mathcal{P}(e)$, zero otherwise
parameter	type	description
c_i	integer	capacity of station $i \in S$
b	integer	total number of bikes in the BSS
ζ_a	integer	demand of ride flow units though link $a \in \mathcal{A}_U$
V_{MAX}	integer	maximal size of the vehicle fleet
C	integer	vehicle capacity
μ_e	integer	maximal bike flow capacity through the link $a \in \mathcal{R}_e$
F	real	fix cost per vehicle used
q_a	real	cost per pick up o deliver flow unit though link $a \in \mathcal{A}_P \cup \mathcal{A}_D$
k_e	real	fix cost per vehicle flow unit through link $e \in \mathcal{A}_{VM} \cup \mathcal{A}_{HA}$
β_a	real	penalization cost per unmet ride flow unit

as a schedule of vehicle flows. First, the vehicle stays at the depot one period, which is represented by a holding vehicle link. Then, the vehicle requires one time period for arriving at station 1. Notice that the vehicle parks at station 1 for one time period. Here, a pickup flow tells that bikes are moved from the station to the vehicle, thus varying the fill of the respective rack and vehicle nodes. After that, the loaded vehicle drives to station 2. Again, the vehicle stays there one time period to move bikes from the vehicle to the station. Finally, the vehicle must return to the depot.

We now defines variables of the formulation. Let I_i^t and w_i^t represent the number of bikes at nodes $(i, t) \in \mathcal{N}_S$ and $(\hat{i,t}) \in \mathcal{N}_V$, respectively. Each vehicle has a load capacity of C bikes. Let v be the vehicle fleet size. The maximal vehicle fleet size is defined by V_{MAX}

Let y_e represent the design decision determining the vehicle flow units assigned to the link $e \in \mathcal{A}_V$. The variable x_a represents the bike flow units through the link $a \in \mathcal{A} \cap \mathcal{A}_V$. We represent unmet ride flow units using the flow variable $z_a, \forall a \in \mathcal{A}_U$. If a handling link $e \in \mathcal{A}_{HA}$ is set, a maximal of μ_e bikes can be picked up or delivered through link e. Additionally, let $\mathcal{R}(e)$ be the set of redistribution links that can generate bike flows when the vehicle link $e \in \mathcal{A}_V$ is set.

We consider the following redistribution costs. Let F represent the fixed cost for using one vehicle. We define k_e as the fixed costs to set a link $e \in \mathcal{A}_M \cup \mathcal{A}_{HA}$. Finally q_a is cost per handled bike with $a \in \mathcal{A}_P \cup \mathcal{A}_D$. Let $\mathcal{A}^+(i,t)$ and $\mathcal{A}^-(i,t)$ represent the outgoing and incoming link set at node $(i,t) \in \mathcal{N}_0$, respectively. Variables and parameters of the formulation are summarized in Table 3. With the notation described before, the MIP formulation reads as follows:

$$\min \{Fv + \sum_{e \in \mathcal{A}_{VM} \cup \mathcal{A}_{HA}} k_e y_e + \sum_{a \in \mathcal{A}_P \cup \mathcal{A}_D} q_a x_a + \sum_{a \in \mathcal{A}_U} \beta_a z_a\} \tag{1}$$

s.t.

$$x_a + z_a = \zeta_a, \quad \forall a \in \mathcal{A}_U \tag{2}$$

$$I_i^t \leq c_i, \quad \forall (i,t) \in \mathcal{N}_S \tag{3}$$

$$\sum_{i \in S} I_i^0 = b \tag{4}$$

$$I_i^{t+1} = I_i^t - \sum_{a \in \mathcal{A}_U^+(i,t) \cup \mathcal{A}_P^+(i,t)} x_a + \sum_{a \in \mathcal{A}_U^-(i,t+1) \cup \mathcal{A}_D^-(i,t+1)} x_a, \quad \forall (i,t) \in \mathcal{N}_S : t < \mathcal{T} \tag{5}$$

$$\sum_{a \in \mathcal{A}_U^+(i,t) \cup \mathcal{A}_P^+(i,t)} x_a \leq I_i^t, \quad \forall (i,t) \in \mathcal{N}_S : t < \mathcal{T} \tag{6}$$

$$\omega_i^{t+1} = \omega_i^t - \sum_{a \in \mathcal{A}_T^+(\hat{i,t}) \cup \mathcal{A}_D^+(\hat{i,t})} x_a + \sum_{a \in \mathcal{A}_T^-(\hat{i,t+1}) \cup \mathcal{A}_P^-(\hat{i,t+1})} x_a, \quad \forall (\hat{i,t}) \in \mathcal{N}_V : t < \mathcal{T} \tag{7}$$

$$\omega_i^t = 0, \quad \forall i \in S, \forall t \in \{0, \mathcal{T}\} \tag{8}$$

$$\sum_{a \in \mathcal{A}_{VM}^+(\hat{i,t}) \cup \mathcal{A}_D^+(\hat{i,t})} x_a \leq \omega_i^t, \quad \forall (\hat{i,t}) \in \mathcal{N}_V : t \neq \{0, \mathcal{T}\} \tag{9}$$

$$\omega_i^t \leq C \sum_{e \in \mathcal{A}_V^-(i,t)} y_e, \quad \forall (\hat{i,t}) \in \mathcal{N}_S, : t \neq \{0, \mathcal{T}\} \tag{10}$$

$$\sum_{a \in \mathcal{R}(e)} x_a \leq C y_e, \quad \forall e \in \mathcal{A}_{VM} \cup \mathcal{A}_{HA} \tag{11}$$

$$x_a = 0, \quad \forall a \in \mathcal{A}_{VM} : i(a) = 0 \vee j(a) = 0 \tag{12}$$

$$x_a \leq \mu_e (1 - \alpha_e), \quad \forall e \in \mathcal{A}_{HA} : a \in \mathcal{P}(e) \tag{13}$$

$$x_a \leq \mu_e \alpha_e, \quad \forall e \in \mathcal{A}_{HA} : a \in \mathcal{D}(e) \tag{14}$$

$$\sum_{e \in \mathcal{A}_V^+(\hat{i,t})} y_e = \sum_{p \in \mathcal{A}_V^-(\hat{i,t})} y_e, \quad \forall (\hat{i,t}) \in \mathcal{N}_0 : t \neq \{0, \mathcal{T}\} \tag{15}$$

$$\sum_{e \in \mathcal{A}_V^+(0,0)} y_e = \sum_{p \in \mathcal{A}_V^-(0,\mathcal{T})} y_e = v \tag{16}$$

$$v \leq V_{MAX} \tag{17}$$

$$I_i^t \geq 0. \quad \forall (i,t) \in \mathcal{N}_S \tag{18}$$

$$\omega_i^t \geq 0. \quad \forall (\hat{i,t}) \in \mathcal{N}_V \tag{19}$$

$$x_e \geq 0, \quad \forall a \in \mathcal{A} \backslash \mathcal{A}_{HO} \tag{20}$$
$$z_a \geq 0, \quad \forall a \in \mathcal{A}_U \tag{21}$$
$$y_e \geq \mathbb{Z}^+, \quad \forall e \in \mathcal{A}_{HO} \tag{22}$$
$$y_e \geq \{0, 1\}, \quad \forall e \in \mathcal{A}_V \backslash \mathcal{A}_{HO} \tag{23}$$
$$v \in \mathbb{Z}^+ \tag{24}$$

The objective function (1) minimizes the sum of costs associated with the redistribution plan and the demand of ride flows that are not met. Equation (2) determines whether the ride flow demand are met. Equation (3) ensures that the number of bikes at a rack node cannot exceed the station capacity. Equation (4) ensures that all bikes are placed at some station at the beginning of the workday length. The flow conservation of bikes at rack nodes is ensured by Equation (5). Equation (6) imposes that the bike outgoings from a rack node do not exceed the number of bikes that are currently placed at the station. Equation (7) ensures the flow conservation of bikes at vehicle nodes. Equation (8) enforces that vehicles are empty at the beginning and end of the workday length. Equation (9) ensures that the bike outgoings from a vehicle node do not exceed the number of bikes of the vehicle load. Equation (10) determines that the vehicle node can have bikes only if it is visited by a vehicle. Equation (11) connects the vehicle links with the possibility to move bike. Bikes can be moved neither from nor to the depot (12). Equations (13) and (14) ensures that bikes are not simultaneously picked-up and delivered at a station and time period. The vehicle-balance constraints are established in (15). Equation (16) ensures that all vehicles start and end master tours at the depot. The vehicle fleet size is bounded by Equation (17). Finally, the variable domain is established in (18 - 24).

6 Solution Methodology

By integrating resource management decisions to SNDBSS, even moderately-sized problem instances become computationally prohibitive with the sole use of exact techniques. Since commercial solver must balance vehicle flows at nodes, they slowly converge to feasible solutions, compare Andersen et al. (2009). However, we note that by fixing a major part of vehicle flow variables, a solver can find the optimal solution of the resulting *reduced MIP problem* quickly. We take advantage of this observation to design a matheuristic. The matheuristic relies on a neighborhood-search scheme that iteratively builds and solves reduced MIP problems to exploit areas of the search space efficiently, see for example Hewitt et al. (2010).

It is a matter of experience that neighborhood-search schemes require some starting solution of reasonable quality to speed up the exploration process. In addition, we note that for service network design models, neighborhood-search schemes cannot adequately balance fixed costs of using vehicles with other costs associated with the problem. In other words, such schemes have cannot adequately determine the vehicle fleet size (Hoff et al., 2010). Therefore, we propose a construction heuristic which breaks down the decision-making process of bike redistribution into three phases: the generation of FLA (demand management), the pairing of FLA into transport services (service management), and the sequence of transport services into master tours (resource management).

6.1 Matheuristic

Given a starting solution, the matheuristic successively builds and solves neighborhoods until it reaches a termination criterion. A neighborhood comprises a set of variables: all variables outside the neighborhood are fixed, whereas the variables inside the neighborhood define a reduced MIP problem.

To obtain feasible solutions with respect to the vehicle-balance constraints, the matheuristic carefully chooses vehicle flow variables depicting vehicle detours. Given master tours of the incumbent solution, a feasible detour tells the time period where the vehicle leaves its assigned master tour, a time-ordered sequence of station visits within the detour, the handling time at each station visit, and the time period when the vehicle returns to the original master tour.

A neighborhood contains all vehicle flow variables that define master tours in the incumbent solution, as well as all vehicle flow variables that defines detours opportunities chosen by the matheuristic. All vehicle flow variables outside the neighborhood are fixed. Other continuous variables comprising the SNDBSS instance are part of the neighborhood. Hence, a solver determines which set of vehicle flow variables inside the neighborhood yields the lowest objective value, while satisfying the constraints sets of the reduced MIP problem. Notice, that the matheuristic can address very large problem instances of SNDBSS since it does not require the full problem instance to be in computational memory.

It is well-known that neighborhood-search schemes require a suitable calibration of size and structure of neighborhoods for a reasonable trade-off between solution quality and computational efforts (Hewitt et al., 2010; Ahuja et al., 2002). In the following, we introduce neighborhood operators that carefully choose a promising and reasonably-sized set of detours to define reduced MIP problems.

6.2 Neighborhood Operators

To design the neighborhood operators, we follow Hoff et al. (2010) who introduce a variable neighborhood search (VNS) for stochastic service network design problems in freight transportation. Their VNS regularly swaps apriori-defined neighborhoods within iterations. The authors propose two neighborhood structures: the first one exploits promising areas of the search space, that is, vehicle movements that involve a high transport volume of commodities between location pairs, whereas the second one allows random moves to explore solutions that the first structure cannot reach. Although the idea behind both structures can be adapted to address the SNDBSS, defining neighborhoods a priori seems an inferior strategy because of strongly interdependencies between flow layers and station's fill levels. Therefore, the proposed matheuristic dynamically builds neighborhoods according to the incumbent solution.

We define a vehicle link to be "active" if the respective vehicle flow variable takes a positive value in the incumbent solution. A rack node is active in the case of a vehicle visits it. A vehicle link becomes "candidate" if a neighborhood operator chooses it. In one iteration, both candidate and active vehicle flow variables comprise the neighborhood.

Neighborhood operators choose candidate vehicle links so that feasible detours are obtained with respect to the design-balance constraints. Figure 6 shows in a portion of a time-expanded network how a neighborhood can look like. For the sake of simplicity, vehicle nodes and redistribution flows are not depicted. Solid lines depict active vehicle links, whereas dashed lines depict candidate vehicle links. We observe that candidate vehicle links define two possible detours. Notice, that a detour may involve two station visits as long as the detour returns somewhere to the original master tour. Certainly, there exist more detour possibilities that can be derived from the incumbent solution.

Each detour is suggested as follows. First a sub-path connecting non-active rack nodes is chosen. The sub-path involves one or two stations within the detour. We assign handling time at each station by a certain number of successive handling links. The maximal number of time periods that a vehicle stays at a station cannot exceed the number of time periods required to completely load a vehicle. To respect the design-balance constraints, we finally determine candidate vehicle links to connect the generated sub-path with one of the vehicle tours in the incumbent solution. Notice that connecting the generated sub-path with all master tours is possible but may lead to very large neighborhood sizes. Therefore, the sub-path is assigned to a master tour according to the driving cost of leaving and returning to the original master tour based on the build sub-path. With a roulette wheel selection criteria, we establish that the lower vehicle movement costs, the higher probability that the corresponding detour is chosen. Allowing costly vehicle movements allows for exploring detours which may generate good decisions from the service level point of view.

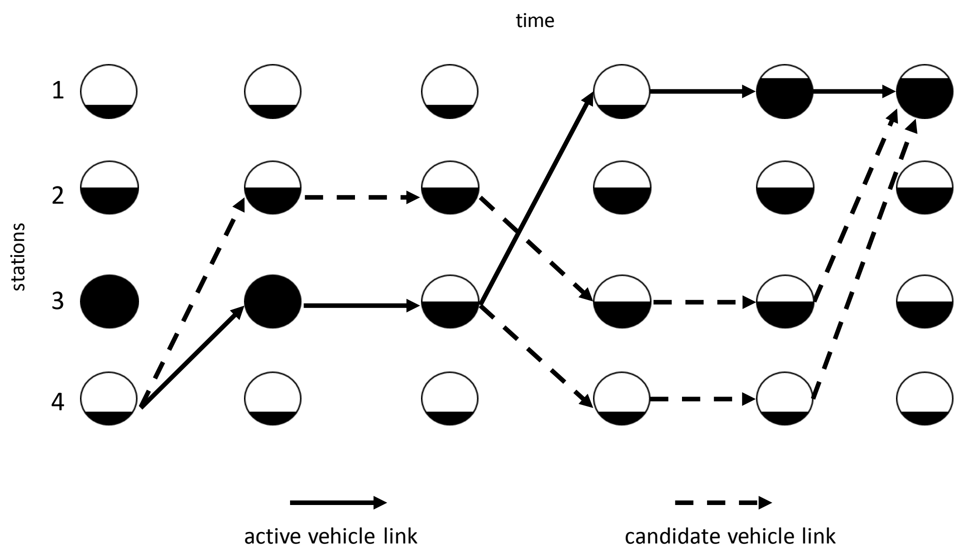


Figure 6: Feasible detours derived from an incumbent solution.

The way how sub-paths are chosen is crucial to conduct promising neighborhoods. Three neighborhood operators are proposed with the following criteria:

- Random-period operator (RPO): detours through different stations in one time period.
- Random-station operator (RSO): detours through one station for different time periods.
- Fill-level based operator (FBO): detours whose corresponding redistribution decisions probably involve a high number of bikes.

In the following, we describe these neighborhood operators in detail.

RPO permits vehicle detours through different stations in a randomly chosen time point t_{random}. Each vehicle detour consists of: 1) one candidate vehicle movement link connecting one active rack node with one non-active rack node $(i,t) \in \mathcal{N}_S \mid t = t_{random}$, 2) a randomly chosen number of candidate handling links to allow pick-up or delivery of bikes, and 3) a vehicle movement link to return to one active rack node of the original vehicle tour. Since the number of vehicle detour possibilities can get very large, the number of detours chosen in an iteration is limited by parameter ϖ.

RSO explores several detours through a randomly chosen target station i_{random} over different time periods. The way how vehicle detours are composed only differs from the RPO in that one candidate vehicle movement link connects one active rack node to one non-active rack node $(i, t) \in \mathcal{N}_S \mid i = i_{random}$. Similarly to RPO, a maximal number of φ are chosen.

FBO relies on the intuitive idea that stations with a deficit of bikes may receive them from a station with a surplus of bikes. FBO chooses a maximal of ϱ vehicle detour possibilities. The FBO conducts a vehicle detour as follows. First, the neighborhood operator generates a sub-path connecting non-active rack nodes. The sub-path represents a transport service, together with the corresponding handling time at each station. Stations are chosen in the hope that a large bike volume between the origin and destination station is redistributed. FBO tries to connect the generated sub-path with an existing vehicle tour in order to yield the detour. If this connection fails because no feasible detour is obtained with respect to the design-balanced constraints, the sub-path is discarded. The rack nodes associated with the origin and destination stations are chosen using roulette wheel selection. The selection of the first rack node is based on the fill level $I_i^t/c_i, \forall(i, t) \in \mathcal{N}_S$, whereas the second rack node is based on the ratio between the number of free bike racks and the station capacity, i.e., $(1 - I_i^t)/c_i, \forall(i, t) \in \mathcal{N}_S$.

6.3 Defining a Suitable Mix of Neighborhood Operators

Given the characteristics of the neighborhood operators, FBO outperforms RPO and RSO the first iterations of the matheuristic since it can identify promising transport services. However, FBO loses its effectiveness in later iterations because it easily stagnates in a local optimum. Randomization provided by RPO and RSO helps in escaping from local optima. Thus, there still exists the option that FBO contribute to find solutions of good quality after escaping from local optima. We make use of these observations to determine a suitable combination of the neighborhood operators for each iteration.

Combining the above neighborhood operators to build neighborhoods permits to take advantage of their outstanding features. A suitable mix of neighborhood operators can guide an effective exploration of the search space even with small neighborhoods, thus reducing the computational effort. Let n denote the total number of detours chosen to build a neighborhood, that is, $n = \varpi + \varphi + \varrho$. Given a value of n, we dynamically adjust ϖ, φ, and ϱ according to the performance of neighborhood operations in previous iterations.

Changing the relative frequency of applying neighborhood operators increases the efficiency of the matheuristic. To this end, we derive probabilities that control the number of vehicle detours chosen by each neighborhood operator. The probabilities depend on the success of the corresponding neighborhood operator, that is, the number of vehicle flow variables chosen by the neighborhood operator that comprises a new solution. After defining the probabilities, a roulette wheel selection determines how many detours are allowed by each neighborhood operator in the next iteration. We spin the roulette wheel n times. The value of a detour parameter is the number of wins of the corresponding neighborhood operator in the roulette wheel. We determine a high probability of FBO against RPO and RSO in early iterations while decreasing the usage of FBO in further iterations. The decrease rate of the FBO probability depends on its contribution finding solutions of improved quality.

Let $\{o\} = \{RPO, RSO, FBO\}$ define the set of neighborhood operators. We define the probability $\pi^o \in [0, 1]$ for choosing a detour using the neighborhood operator o. In iteration p, the neighborhood operator o chooses a number of c^o candidate vehicle links to build the neighborhood. The number of candidate vehicle links c^o the solver sets to one is denoted by s^o. Based on these definitions, we define the success ratio $r^o = \frac{(s^o)^2}{c^o} \gg 0$. Notice, that we use the squared term $(s^o)^2$ to determine the ratio in order to prevent that the respective probability π^o quickly decreases fast. In addition, let $\Theta^o \in [0, 1]$ be a parameter that avoids that the probability of neighborhood operator o decreases to zero.

Thus, at each iteration p, we calculate the probability π^{FBO} for choosing a detour using the FBS with the following equation:

$$\pi^{FBO} := 1 - \frac{\Theta^{FBO} p}{r^{FBO} + p} \tag{25}$$

In Equation (7), probability π^{FBO} approximates $1 - \Theta^{FBO}$ over iterations. Parameter r^{FBO} controls how quick π^{FBO} converges. The higher the value of π^{FBO}, the slower the probability converges to $1 - \Theta^{FBO}$. Figure 7 illustrates curves depicting values of π^{FBO} within iterations for different values of r^{FBO} with $\Theta^{FBO} = 0.7$.

Regarding the probability of RPO and RSO, we follow Niehaus and Banzhaf (2001) to define the following equation:

$$\pi^o = (1 - \pi^{FBO})\big(\Theta^o + r^o \frac{1 - 2\Theta^o}{\Gamma}\big), \quad \forall o \in \{RPO, RSO\} \tag{26}$$

In Equation (26), we define $\Gamma = r^{RPO} + r^{RSO}$ and $\Theta^{RPO} = \Theta^{RSO}$. On the right side of this equation, the first term indicates that the sum of π^{RPO} and π^{RPO} equals $(1 - \pi^{FBO})$. The second term adjusts the probability with respect to the success of the respective neighborhood operator.

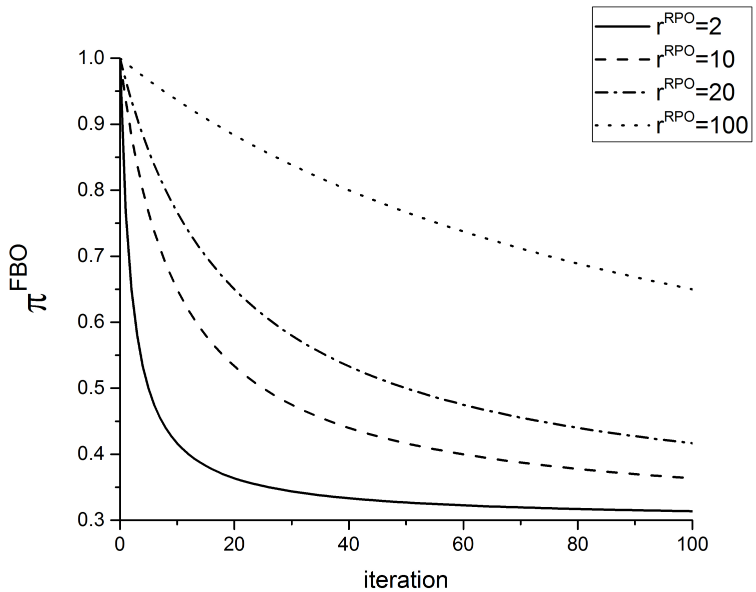


Figure 7: Behavior of π^{FBO} for different values r^{FBO}, with $(1 - \Theta^{FBO}) = 0.3$.

6.4 Construction Heuristic

The construction heuristic splits the redistribution process into three phases. Figure 8 sketches the information flow between phases: each box represents a phase, whereas unidirectional information flows depict the mechanism to coordinate phases. Outputs of one phase are used as input for the next phase. This top-to-bottom coordination mechanism breaks down interdependencies of management processes highlighted in Figure 4 since there is no information flow between resource and demand management. On the one hand, neglecting some of these interdependencies deteriorates time, space, and volume synchronization throughout phases. On the other hand, breaking down these interdependencies allows for substitute problems that are easier to solve than the original problem instance of SNDBSS.

The first phase determines the *maximal FLAs*, that is, the necessary FLAs to fulfill given ride flows. This phase concerns with the demand management process determining FLAs according to the needs of every single station. We depict FLAs by black circles with a “P” and “D” inside for pickups and deliveries of bikes, respectively. The maximal FLAs feed the second phase of the construction heuristic.

The second phase is concerned with service management decisions within the scope of the redistribution flow layer. The aim is to perform as many FLAs determined in the first phase as possible. Notice that it is possible that we cannot fulfill the maximal FLAs because of a volume discordance between the bikes to pick up and deliver over time, as

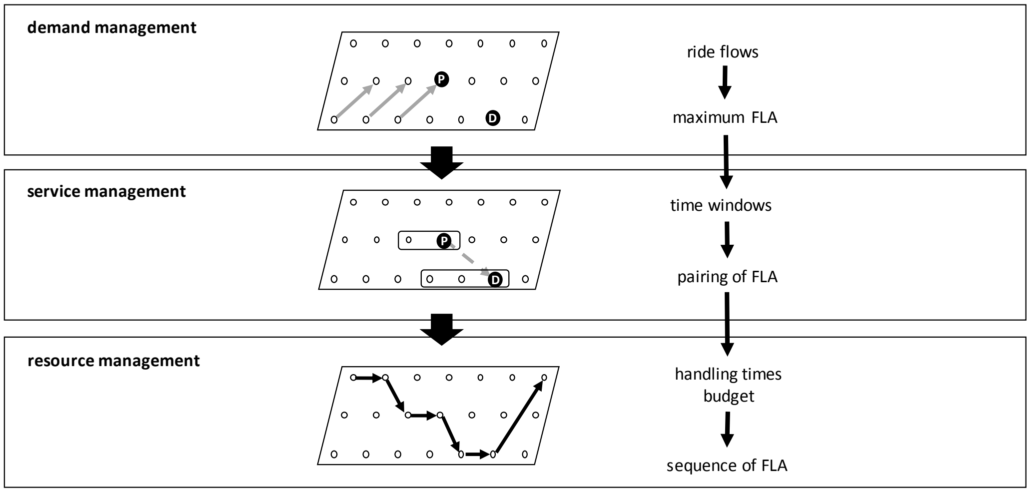


Figure 8: Framework of the construction heuristic.

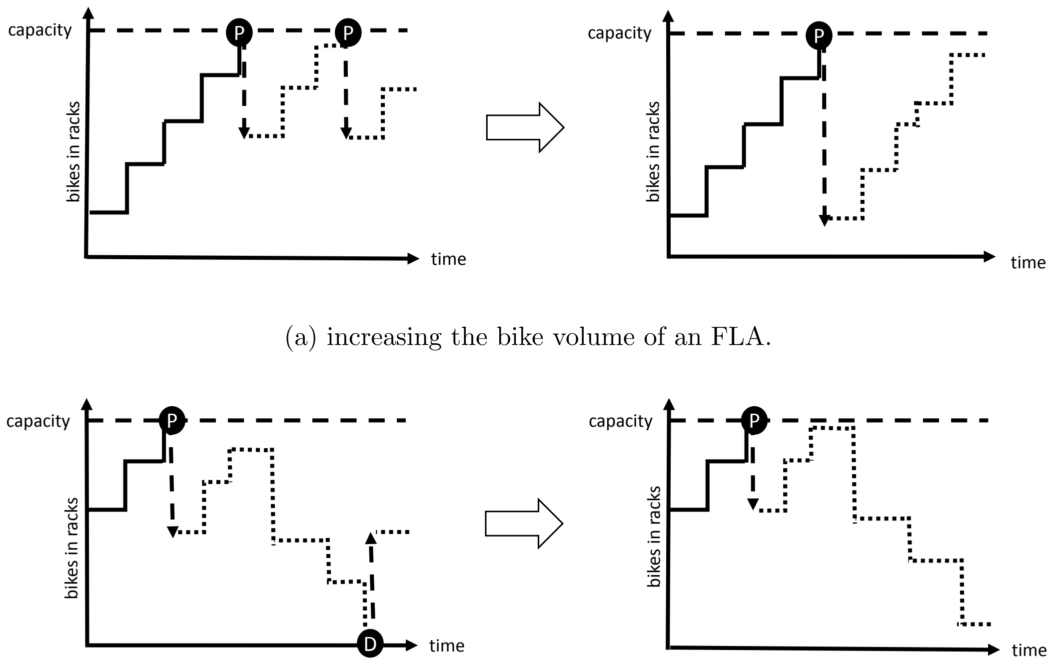
well as a weak time synchronization to redistribute the desired bike volumes between stations. To alleviate these deficiencies, we design *time windows* where FLAs can be performed. Time windows increase the operational flexibility for station visits. Notice that time windows involve tradeoffs between time and volume synchronization because station's fill levels vary over time. This phase obtains *pairs of FLAs* defining transport services.

Finally, the third phase addresses resource management decisions by determining master tours, that is, a *sequence of FLAs* assigned to vehicles. In this phase, we consider handling times as well as resource limitations when determining master tours. In principle, the problem to solve takes the form of the pick-up and delivery problem with time windows (PDPTW) presented in Ropke and Pisinger (2006).

To produce a solution of SNDBSS, we derive vehicle flow variables that depict master tours obtained from the PDPTW instance. Then, we just need to fix the respective vehicle flow variables such that we can obtain these master tours for the problem instance of SNDBSS. Thus, a solver can quickly determine the redistribution flows and met ride flows. In the following, we proceed to describe formally each phase of the construction heuristic.

6.4.1 First Phase: Determining Maximal FLA.

We define a procedural approach to determine FLAs at stations. The procedural approach relies on recursively equations to trigger an FLA each time period that a station does not provide sufficient assets as a consequence of bike rentals and returns. In this way, failed requests are avoided on a per station consideration.



(a) increasing the bike volume of an FLA.

(b) decreasing the bike volume of an FLA.

Figure 9: Adjusting bike volumes to rationalize FLAs.

Clearly, the number of FLAs to be performed impacts on the redistribution costs: the more FLAs, the more station visits and driving time is required within the workday length. Therefore, we aim to rationalize the number of FLA to be performed such that failed requests are avoided. Consider first the case depicted in Figure 9a. On the left side, we observe that two pickup FLAs are triggered each time that the station runs full. On the right side, we note that we can omit the second FLA if we increase the pickup volume of the first FLA because we avoid that this station becomes full or empty in further time periods. In Figure 9b, we observe on the left side that a pickup FLA is firstly triggered at one station, followed by a delivery FLA later. On the right side, we note that the delivery FLA can be omitted if we decrease the pickup volume of the first FLA such that the station does not suffer from failed requests. We observe the following four cases:

1. Increasing the volume in a pick-up FLA may avoid further pick-up FLAs.
2. Increasing the volume in a delivery FLA may avoid further delivery FLAs.
3. Decreasing the volume in a pick-up FLA may avoid further delivery FLAs.
4. Decreasing the volume in a FLA operation may avoid further pick-up FLAs.

We consider these cases to design a procedural approach with the aim of rationalizing FLAs. First, we introduce the recursive equations to obtain FLAs. Then, we describe the mechanism to adjust pickup and delivery volumes for reducing the number of FLA per station. Finally, the procedural approach is introduced.

Recursive Equations. Let I_i^t be the number of bikes in racks at station i and time period t. We assume a given number of bikes I_i^0 on station i and time period 0. In further time periods, the value of I_i^t depends on the number of bikes in station i and time period $t-1$, as well as bike rentals and returns. Let f_i^{t+1} be the difference between bike returns in time period $t+1$ and the bike rentals in time period t. Such difference is computed by

$$f_i^{t+1} = -\sum_{a \in \mathcal{A}_U^+(i,t)} \zeta_a + \sum_{a \in \mathcal{A}_U^-(i,t+1)} \zeta_a, \quad \forall i \in S, t < \mathcal{T} \tag{27}$$

Recall, that ζ_a indicates the ride flow demand on the link $a \in \mathcal{A}_U$ in the time-expanded network of SNDBSS. A positive value of f_i^t says that expected ride flows lead to an increase of the number of bikes in racks on station i and time period t, whereas a negative value represents a decrease of bikes. If f_i^t equals zero, there is no variation in the number of bikes in racks. For time period 0, the value of f_i^0 is zero.

If the number of bike in racks at station i is out $[0, c_i]$, we trigger an FLA picking up or delivering a certain bike volume. We denote the volume of handled bikes in station i and time period t by r_i^t. A negative value of r_i^t means that a pick-up of bikes is required, whereas a positive value indicates the volume of bikes to deliver. In the case that r_i^t equals zero, no FLA is necessary. Notice, that this phase does not consider resource limitations to perform FLA over time. Let θ_i^t be the target number of bikes in racks on station i and time period t if a FLA is performed.

We determine the number of bikes in racks at a station over time periods with

$$I_i^t = \begin{cases} I_t^0, & \text{if } t = 0 \\ I_i^{t-1} + f_i^t, & \text{if } I_i^{t-1} + f_i^t \in [0, c_i] \\ \max\{0, \min\{c_i, I_i^{t-1} + f_i^t + r_i^t\}\}, & \text{otherwise} \end{cases} \tag{28}$$

I_t^0 denotes the number of bikes in racks of station i in time period zero. If the number of bikes in racks stay in the interval $[0, c_i]$ in subsequent time periods, no FLA is required. Otherwise, an FLA is triggered. In this case, the pick-up or delivery volume is determined by

$$r_i^t = \begin{cases} 0, & \text{if } t = 0 \\ \min\{C, \theta_i^t - [I_i^{t-1} + f_i^t]\}, & \text{if } I_i^{t-1} + f_i^t \notin [0, c_i] \\ 0, & \text{otherwise} \end{cases} \tag{29}$$

No FLA is required the first time period. In subsequent time periods, a pick-up or delivery volume is triggered if required, considering that this bike volume cannot exceed the vehicle load capacity C. If no FLA is necessary, the bike volume r_i^t is zero.

Adjusting FLA. Now, we formally describe the procedure to adjust one FLA based on the next FLA performed at the same station. Let r_i^t and $r_i^{\bar{t}}$ bike volumes for FLA on station i, with $t < \bar{t}$. No FLA occurs between this time interval. In addition, we define the minimum and maximum number of bikes in racks between time periods t and $\bar{t}$ by I^{MIN} and I^{MAX}, respectively. In the following, we define how to adjust r_i^t according to the four aforementioned cases:

1. This case involves two pickup FLAs. The bike volume to pick up $|r_i^t|$ may increase up to I^{MIN} units.
2. This is the case if both FLAs delivers bike volumes at station i. Now, the bike volume to deliver $|r_i^t|$ may increase up to $c_i - I^{MAX}$ units.
3. This case first involves a pickup FLA and then a delivery FLA. Thus, the bike volume to pick up $|r_i^t|$ may decrease to $\min\{I^{MIN}, c_i - I^{MAX}\}$ units.
4. Finally, first a delivery FLA and then a pickup FLA occurs. The procedure to define the decrease in volume of bikes to deliver $|r_i^t|$ is the same that we consider for case 3.

Initial Number of Bikes on Stations. The procedural approach requires as input the number of bikes in racks I_i^0 at station i as input. Certainly, the value of I_i^0 has an impact in the time period where the first FLA occurs, depending if it involves a pick-up or delivery of bikes.

In order to adjust I_i^0, we proceed as follows. Let t the time period when the first FLA occurs. Two cases are distinguished:

- If the first FLA involves a bike volume to pick up, I_i^0 decreases until I^{MIN}.
- If the first FLA involves a bike volume to deliver, I_i^0 increases until I^{MAX}.

The Procedural Approach starts with an initial fill level of 50% in time period 0, that is, $I_i^0 := \lfloor \frac{c_i}{2} \rfloor$. Then, we determine the initial setting of FLAs in the time interval $[0, \mathcal{T}]$ using the recursive equations. We stop if no FLA is triggered at station i. If at least one FLA is triggered, we adjust I_i^0 and determine FLAs using the recursive equations, now in the interval $[1, \mathcal{T}]$.

After that, if two or more FLA are triggered at station i, we proceed as follows. Let $[t^{INIT}, \mathcal{T}]$ be the target time interval where the recursive equations trigger FLAs. Let t and $\bar{t}$ be the time periods where the first and second FLAs are triggered in this target time interval, respectively. We adjust the bike volume of FLA in time period t according to the four cases presented above. Then, we set $t^{INIT} := t^{INIT} + 1$ run the recursive equations in the time interval $[t^{INIT} + 1, \mathcal{T}]$, fixing all FLAs defined in previous time periods.

Then, if two or more FLAs are triggered in the time interval $[t^{INIT}, \mathcal{T}]$, we repeat the procedure described in the previous paragraph in the remaining time interval. The procedure terminates if less than two FLA are triggered within the target time interval.

6.4.2 Determining Time Windows for FLAs.

Since station interdependencies are neglected in the first phase, there is no time synchronization between pickup and delivery FLAs.

For each FLA (i, t), t represents the latest time period where the FLA must occur to avoid failed requests. However, the FLA may certainly occur earlier than t. Although this can lead to a smaller bike volume since the number of bikes in racks at station i variates over time, designing time windows for FLAs may enhance the time synchronization for linking pickup and delivery FLAs into transport services.

We design time windows for every FLA as follows. Given an FLA (i, t), the time window is generated by replicating nodes on the corresponding station i for a certain number of previous δ time periods. Thus, δ defines the time window length. Thus, all nodes $\{(i, t-\delta), (i, t-\delta+1), ..., (i, t)\}$ constitutes a request.

For each request, we determine the parameter δ as follows. We firstly set $\delta := 1$. Then, we check if the number of bikes in racks $I_i^{t-\delta}$ – which is given from phase one – permits that at least a certain percentage of the desired bikes volume to pick up or deliver is available. If so, the node $(i, t-\delta)$ is included to constitute the time window. After that, we set $\delta := \delta - 1$. We repeat the procedure again until: a maximal time window length δ_{MAX} is achieved, the minimal bike volume to pick up or deliver is not available in node $(i, t-\delta)$, or if there already exist a pick-up or delivery FLA in node $(i, t-\delta)$ to avoid the time overlapping of requests. FLA generated in this phase are used as inputs for the second phase, where the transport services are determined.

6.5 Second Phase: Determining Pairs of FLAs

The second phase relies on a transportation problem to pair pickup and delivery FLAs. The transportation problem aims to redistribute as much bike volume as possible with respect to the maximal FLA defined in the first phase. Recall, that FLAs are given by all variables r_i^t that take negative (pick-up) or positive (delivery) values in the first phase. Thus, we define nodes (i, t) representing the station i, time period t, and bike volume $|r_i^t|$ to pick-up or deliver.

We denote the set of pick-up and delivery nodes by N_P and N_D, respectively. The available bike volume to pick-up at node $n_P \in N_P$ is denoted by a_{n_P}, whereas the bike volume to delivery at node $n_D \in N_D$ is b_{n_D}. Since we consider homogeneous bikes, a delivery node may receive bikes from any pick-up node. We refer to these nodes as pickup and delivery requests.

We denote pick-up and delivery requests by R_P and R_D, respectively. The available bike volume to pick-up at request $r_p \in R_P$ is denoted by A_{r_p}, whereas bike volume required at delivery node $r_d \in R_D$ is B_{r_d}. As aforementioned, each request is defined by a set of nodes. Let $N_P(r_D)$ and $N_D(r_D)$ the node sets which belong to the pick-up request r_P and r_D, respectively. Since the first phase do not establish station interdependencies when defining FLA, it is possible that the total bike volume to pick up is not equal to the bike volume to deliver, i.e., $\sum_{r_p \in R_P} A_{r_p} \neq \sum_{r_d \in R_D} B_{r_d}$. For this reason, we introduce the non-negative variables α_{r_P} and β_{r_D} indicating dummy bike volume, i.e., volume that is not picked-up or delivered from requests $r_P \in R_P$ and $r_D \in R_D$, respectively. Each lost bike volume unit is penalized with a cost θ.

Figure 10 shows a solution of the transportation problem and the benefits of time windows. The y-axis represents stations, whereas the y-axis represents time periods. FLAs are depicted by black nodes (i, t) representing station i and time period t. We observe that the alternatives to pair FLAs are limited to the way in which they are spread over time. Incorporating time windows alleviates this timing deficiency. Gray circles depict replications of the given FLAs in previous time periods thus defining time windows. Each dashed oval wraps a certain number of nodes thus depicting a request. Black lines represent transport services involving a certain volume of bikes from a pick-up node to a delivery node.

The decision of establishing a redistribution request between two nodes is modeled by the binary variable $y_{kl}, k \in N_P, l \in N_D$, whereas the respective redistribution volume is represented by the non-negative continuous variable $x_{kl}, k \in N_P, l \in N_D$. The volume to redistribute is bounded by the capacity of the redistribution request $C_{kl}, k \in N_P, l \in N_D$.

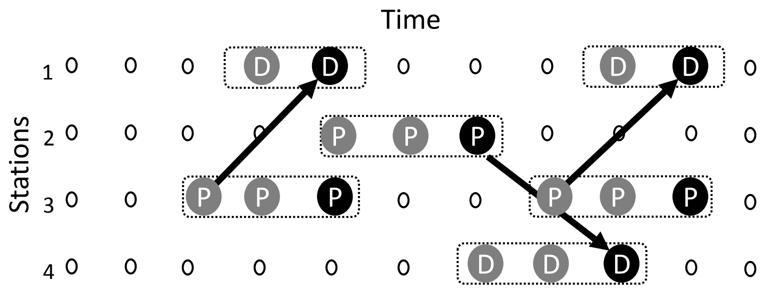


Figure 10: Exemplary solution of the transportation problem.

The allowed volume for a redistribution request depends on the minimum bike volume of the corresponding pick-up and delivery node, i.e., $\min\{a_k, b_l\}$, the number of available time periods to perform the redistribution between the targeted stations, and the vehicle load capacity C. To avoid excessive driving time for performing redistribution, a fix cost f_{kl} is considered if the redistribution request y_{kl} is set.

With the aforementioned notation, the transportation problem reads as follows:

$$\min \quad z = \sum_{k \in N_P} \sum_{l \in N_D} f_{kl} y_{kl} + \theta\big(\sum_{r_P \in R_P} \alpha_{r_P} + \sum_{r_D \in R_D} \beta_{r_D} \big) \tag{30}$$

s.t.

$$\sum_{k \in N_P(r_P)} \sum_{l \in N_D} x_{kl} + \alpha_{r_P} = A_{r_P}, \quad \forall r_P \in R_P \tag{31}$$

$$\sum_{k \in N_P} \sum_{l \in N_D(r_D)} x_{kl} + \beta_d = B_{r_D}, \quad \forall r_D \in R_D \tag{32}$$

$$\sum_{k \in N_P(r_P)} \sum_{l \in N_D} y_{kl} \leq 1, \quad \forall r_P \in R_P \tag{33}$$

$$\sum_{k \in N_P} \sum_{l \in N_D(r_D)} y_{kl} \leq 1, \quad \forall r_D \in R_D \tag{34}$$

$$x_{kl} \leq C_{kl} y_{kl}, \quad \forall k \in N_P, l \in N_D \tag{35}$$

$$x_{kl} \geq 0, \quad \forall k \in N_P, l \in N_D \tag{36}$$

$$y_{kl} \geq \{0, 1\}, \quad \forall k \in N_P, l \in N_D \tag{37}$$

$$\alpha_{r_P} \geq 0, \quad \forall r_P \in R_P \tag{38}$$

$$\beta_{r_D} \geq 0, \quad \forall r_D \in R_D \tag{39}$$

The objective function (30) minimizes the fixed costs redistribution requests and the penalty cost associated with lost volumes of bikes that cannot be picked up or delivered by the requests. Equation (32) indicates that the sum of all redistributed bikes from a pick-up request plus the unmet bike volume is equal to the expected bike volume to pick up. In the same way, Equation (33) says that the sum of all redistributed bikes to a delivery request plus the unmet bike volume is equal to the expected bike volume to deliver. Equations (34-35) enforce that only one redistribution decision can be established from or to a request. Finally, Equations (36-39) indicate the domain of the variables.

Clearly, each redistribution decision obtained from the transportation model involves the minimum bike volume between the respective pick-up and delivery requests. Notice that at each of such requests, we can actually pick-up or deliver this minor volume in an earlier time period than established by the respective time window. This means that we are able to enlarge time windows to perform the obtained volumes. This increase the timing flexibility for assigning vehicles to redistribution decisions in the third step. For this purpose, we determine the length of the enlarged time windows in the same way we proceed toward defining the transportation problem.

6.6 Third Phase: Determining Sequences of FLAs

In the third phase, a PDPTW assigns vehicles to transport services, thus obtaining master tours for the redistribution process. The follow the large neighborhood search of Ropke and Pisinger (2006) to solve problem instances. In this work, the LNS uses a combination of greedy insertion and shaw removal. The reader is referred to the respective paper for more detail. To depict feasible master tour decisions, the LNS builds solutions which respect the operational policy that two vehicles cannot park at one station at the same time.

7 Computational experiments

We report the effectiveness of our solution methods as well as the trade-off between service level and resources suggested by SNDBSS. All the algorithms were implemented in C++ using ILOG Concert Technology to access CPLEX 12.5 as MIP solver. Experiments are performed on a computer cluster with Intel Xeon 2.6 GHz processors and 64 GB of RAM that operate under SuSe Linux Enterprise Server version 11.

Table 4: Description of input data after filtering.

	Bay Area	Nice Ride	Hubway	Capital Bikeshare
city	San Francisco	Minneapolis	Boston	Washington
number of stations	35	190	140	354
min - Max - Avg. bike racks per station	15 - 27 - 19	15 - 35 - 18	11 - 46 -17	8 - 40 - 17
year period	01 Mai - 31 Aug.	01 Mai - 31 Aug.	01 Mai - 31 Aug.	01 Jul. - 30 Sep.
avg. trips per day	1,167	2,305	4,551	10,946

7.1 Input Data

We generate problem instances of SNDBSS using the 2015 free available data of four BSS located in the USA: San Francisco's Bay Area (`BA`)[1], Minneapolis's Nice Ride (`NR`), Boston's Hubway (`HU`), and Washington's Capital Bikeshare (`CA`). Available data include size and location of stations, together with recorded user trips. We assume Euclidean distances between physical nodes. The reader is referred to the respective website of each BSS for a more detailed description of the system.

We remove unintended trips, that is, recorded trips whose duration is less than 5 minutes. Weekend and holiday trips are also removed because they present different user behavior in contrast to workdays. In addition, we exclude stations that operate less than one month. Recorded trips related to these stations are removed as well.

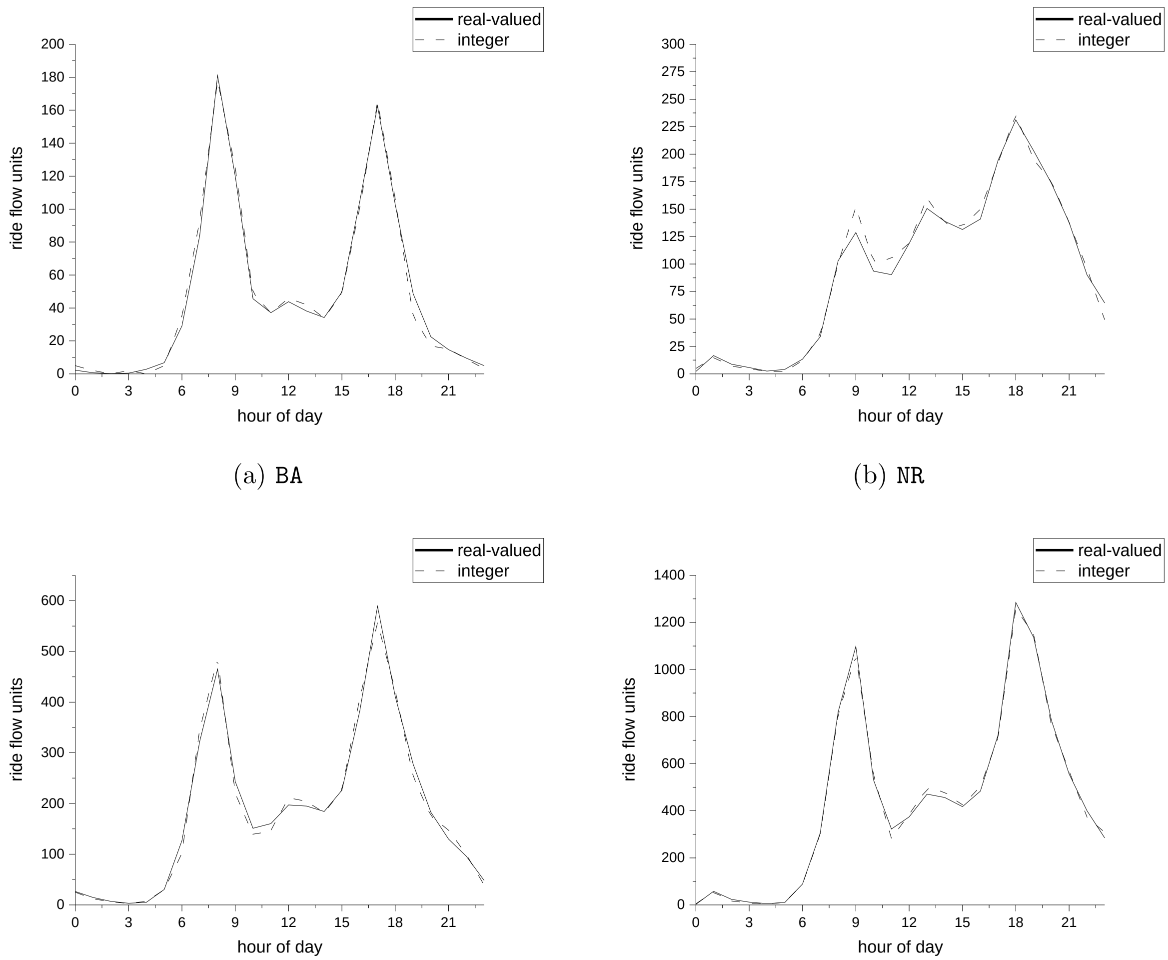
Table 4 provides a general description of the filtered data. In terms of average daily trips and number of stations, `BA` is the smallest and `CA` the largest BSS. We note that the average number of bikes in racks per station does not exceed the value of 20. In terms of user demand, we note that `HU` almost doubles the average daily trips compared to `NR`, although both BSS have a similar number of stations. Furthermore, `NR` only generates around 12 trips per station, whereas all other BSS exceed 30 trips per station. This can be explained by the usage patterns of `NR` which do not depict a clear commuter dominance, but users mainly perform trips for leisure purposes. For an extensive analysis of trip data in several cities, the reader is referred to O'Brien et al. (2014).

7.2 Design of Problem Instances

Problem instances are defined in terms of the network infrastructure, ride flow demand, and all candidate redistribution and master tour decisions. To obtain a detailed representation of time for the redistribution decisions, we split the workday length into 5-minutes time periods.

The network infrastructure includes the vehicle depot, which is artificially located in

[1]In 2017, San Francisco's Bay Area changed its name to Ford GoBike.



(a) BA (b) NR (c) HU (d) CA

Figure 11: Ride flow units according to hour-of-day.

the center of the coverage area. The total number of bikes available in the BSS is around 50 percent of the total number of bike racks available in the respective BSS.

We consider the following parameters when generating problem instances. The cost of using a vehicle for the given schedule length is 250.00 €. The cost per kilometer driven is 0.50€. Using a handling link of the time-expanded network costs 0.10 €. Regarding the handling of bikes, we consider one minute for loading or unloading a bike into the vehicle, and a handling cost of 1.00 € per bike. Each vehicle has a capacity of 20 bikes. We assume a constant driving time of 30 kilometers per hour. Each unmet ride flow unit implies 100.00 € in order to express the high relevance of service in a BSS.

We generate the ride flow layer for every problem instance of SNDBSS as follows. First, we aggregate recorded trip data over days obtaining real-valued flows through every possible ride link of the time-expanded network. Then, real-valued ride flows are discretized by a Poisson distribution process. Figure 11 depicts the sum of real-valued and integer ride flows departing at each time period. We observe that both lines run almost parallel to each other. As expected, hour-of-day flow curves depict morning and afternoon peaks whereas minor BSS use is observed during the nighttime. Since we are interested in the intra-day redistribution process, we consider a workday length of 16 hours starting at 6:00 AM.

7.3 Performance of the Construction Heuristic

We investigate whether the construction heuristic can obtain feasible solutions in reasonable runtime. We consider the improvement ratio as a measure of comparison. The improvement ratio is defined as the percentage reduction of unmet ride flow units in relation to no redistribution carried out. An improvement ratio of 1.0 indicates that ride flow demand is completely satisfied.

In the first phase, the procedural approach requires no parameters. We consider a time windows length of three hours, i.e., $\delta_{MAX} = 36$ In the second phase, transport problems of the second phase are solved using CPLEX. In the third phase, the LNS algorithm used to address instances of PDPTW stops after either 5,000 iterations or 30 minutes runtime. In this experiment, we run the construction heuristic assuming no limits on resources to obtain an upper bound of the number of vehicles.

Table 5 shows for each BSS the improvement ratio, vehicles, as well as runtime in seconds for each phase. For the third phase, we report the time at which the best solution of the respective PDPTW instance is obtained. We observe that the construction heuristic obtains improvement ratios between 0.60 and 0.70 for all BSS except for of `NR`, where the improvement ratio is 0.86. We observe a relationship between the number of vehicles and the size of the BSS. For instance, we note that `CA` requires the largest number of vehicles. This was expected since `CA` approximately doubles the number of stations and daily trips compared with the other BSS. Regarding runtime, we note that the first two phases consume relatively small computational effort only. The first phase only takes 6.42 seconds in average, which was expected since no optimization takes place. The transportation problem defined in the second phase is solved to optimality using CPLEX after few seconds even for the largest BSS. The third phase is, in general, more time-consuming: for `HU` and `CA`, the best solution is obtained after several iterations of the LNS algorithm. We conclude that the construction heuristic produces good starting solutions for the matheuristic.

Table 5: Results of the construction heuristic.

			runtime in seconds			
BSS	imp. ratio	vehicles	1st phase	2nd phase	3rd phase	total
BA	0.61	2	1.33	0.21	0.11	1.65
NR	0.86	3	6.17	0.12	0.18	6.47
HU	0.61	4	5.65	1.05	65.73	51.43
CA	0.66	9	12.54	8.98	875.21	676.73

7.4 Performance of the Matheuristic

I We follow the guidelines of Section 6.3 to build neighborhoods. Regarding the neighborhood sizes, we test on each BSS for $n = 50$; 100; and 200 to define the neighborhood size. Recall, that the parameter n indicates the number of vehicle detours considered in a neighborhood. We set the maximal time for exploring a neighborhood by 600.00 seconds. The matheuristic stops after 10 iterations without improvement by CPLEX, or if it exceeds the runtime limit of 10 hours.

7.4.1 Starting Solution

We investigate whether the construction heuristic is actually needed. Therefore, we evaluate the performance of starting the matheuristic from a solution in which all vehicles are parked, that is, no redistribution occurs. Then, we run the matheuristic five times for each BSS and value of n until 1) it reaches the value of improvement ratio obtained from the construction heuristic or 2) it does not find solutions of better quality after 10 iterations. Table 6 displays the required iterations to the target improvement ratio as well as the average runtime per iteration. Bolded numbers indicate that the at least one run of the matheuristic did not reach the target improvement ratio.

Table 6: Computational efforts devoted by matheuristic to reach improvement ratios obtained from the construction heuristic.

BSS	iterations			runtime in seconds		
	$n = 50$	$n = 100$	$n = 200$	$n = 50$	$n = 100$	$n = 200$
BA	12	5	2	490	101	306
NR	**46**	32	5	**2,076**	1,875	787
HU	17	8	4	2,084	1,760	1,984
CA	32	15	8	10,582	8,518	6,367
average	27	15	5	3,808	3,064	2,361

We note that the matheuristic reaches the target improvement ratio in almost all cases. Although it is not shown in Table 6, we remark that for `NR` and n equals 50, the matheuristic early stagnates three times to a solution of minor quality. In general, the matheuristic requires fewer iterations and less runtime to obtain the desired improvement ratio by increasing the value of n. As expected, large neighborhoods provide a wider exploration of the search space at each iteration.

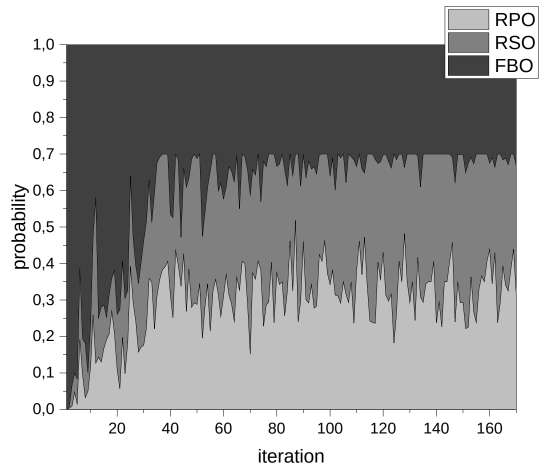
In average, the construction heuristic is at least 10 times faster compared to the matheuristic. To sum up, we can save computational efforts by producing a solution from the construction heuristic. Therefore, we use starting solutions obtained from the construction heuristic for the following experiments.

7.4.2 Neighborhood Operators

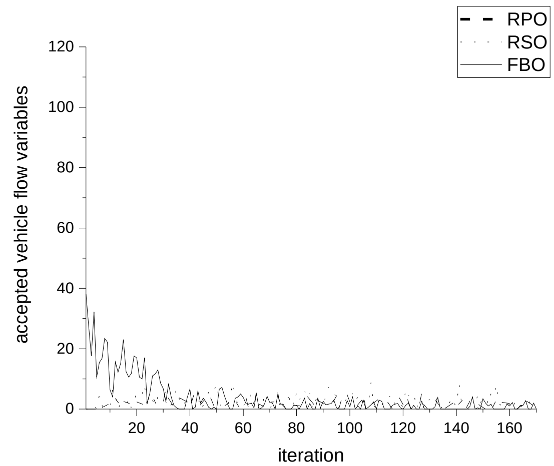
Next, we study the performance of neighborhood operators over iterations with respect to the neighborhood size. The results for `HU` are shown in Figure 12. In all graphs, the x-axis represents iterations of the matheuristic. On the left side of each figure, the stacked area graphs indicate the application of each neighborhood operator over iterations. The graphs on the right side show how many candidate vehicle flow variables are accepted in each iteration, i.e. the success rate. As expected, we observe that the application of FBO quickly decreases within the first iterations since the proposed equations which derive probabilities encourage this behavior. Nevertheless, the decay of application is slightly slower if n equals 200 because a large portion of vehicle flow variables are accepted within the first iterations. As a consequence, the success rate keeps high the application of FBO keeps close to one. In later iterations, no major variations in the application of FBO are observed. Meanwhile, we do not identify a clear superiority of one operator over the other one. However, variations of these applications over iterations indicate that the matheuristic finds solution of better quality on later phases of search.

We note that the number vehicle flow variables accepted by FBO quickly decreases. However, this neighborhood operator is able to improve the incumbent solution even in later phases of search. Interesting, we observe a higher success rate of RSO than for RPO in improving the incumbent solution. Since RPO conducts at each iteration vehicle detours through different stations in one time interval, a vehicle can just accept one of such detours. RSO allows detours through one station in different time intervals, and therefore one vehicle may select more than one possible detour. Nevertheless, both neighborhood operators are beneficial to escape from local optimums.

if n equals 50

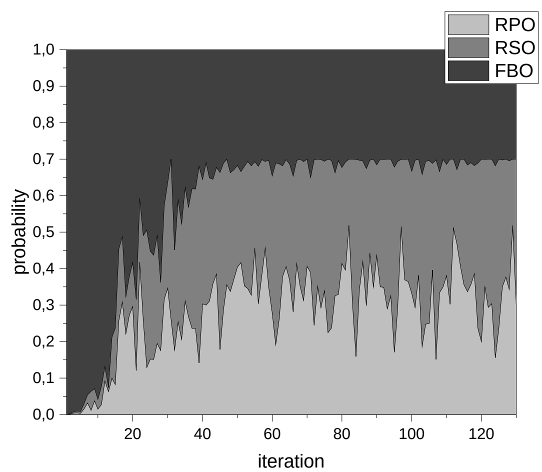


(a) application

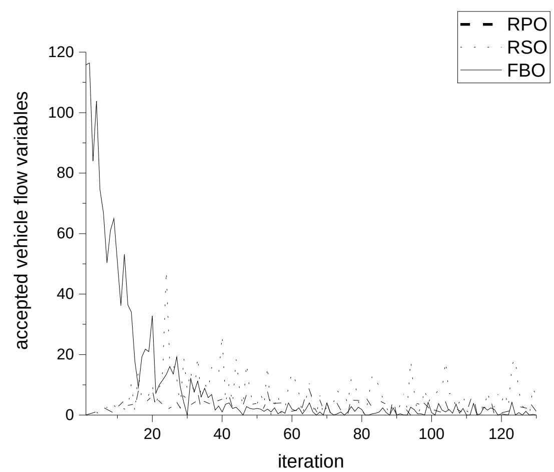


(b) success

if n equals 200



(c) application



(d) success

Figure 12: Performance of neighborhood operators for HU.

7.4.3 Numerical Results

Table 7 shows the average of 5 runs for each input setting including BSS, vehicles, and value of n. We report improvement ratio, redistribution costs, number of iterations, and runtime in seconds. We observe that an improvement ratio of 1.0 is obtained at eight of twelve input settings. CA is the only BSS for which an improvement ratio of 1.0 is not achieved. Comparing solutions whose improvement ratio is 1.0, we note that increasing the value of n leads to a slight reduction of redistribution costs. This suggests that applying large neighborhoods leads to solutions of better quality. However, we observe for CA that solutions of minor improvement ratio are obtained if n equals 200.

Table 7: Numerical results of matheuristic.

input setting			results			
BSS	vehicles	n	imp. ratio	costs	iterations	runtime
BA	2	50	1.00	733.68	118	596
		100	1.00	726.54	180	1,551
		200	1.00	724.21	124	4,206
NR	3	50	1.00	889.89	53	307
		100	1.00	885.73	72	1,130
		200	1.00	881.76	116	7,042
HU	4	50	0.99	1,604.97	204	31,056
		100	1.00	1,588.16	201	35,330
		200	1.00	1,581.37	144	36,062
CA	9	50	0.95	3,803.34	92	35,961
		100	0.95	3,873.01	55	35,947
		200	0.93	3,863.90	46	36,184

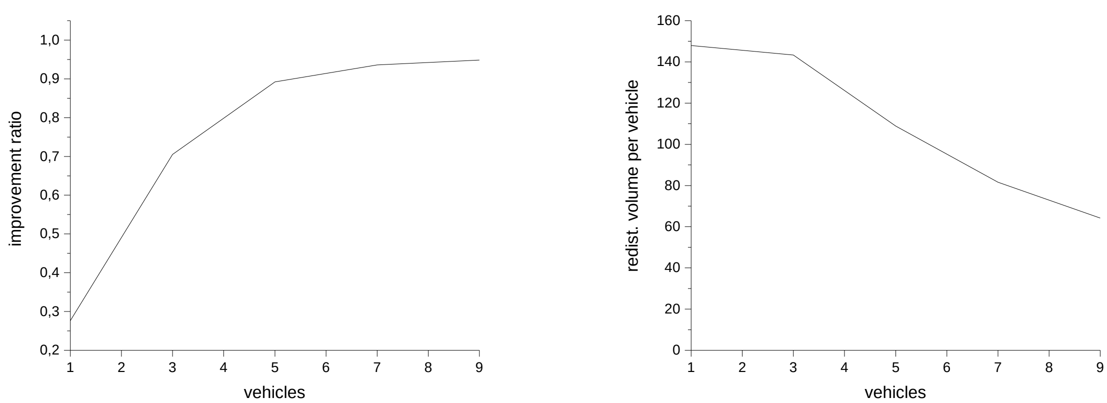
Regarding computational efforts, we note for BA and NR that the runtime increases according to the value of n. For these BSSs, n equals 50 leads to a quick convergence at early iterations. Enlarging the value of n avoids that the matheuristic quickly stagnates at a local optimum and therefore more iterations are conducted in general. We note that for BA, setting $n = 200$ is more time-consuming than $n = 100$, but reaches solutions of minor costs. For HU and CA, we observe that more runtime is required to obtain solutions of good quality. In turn, the number of iterations decreases by increasing the value of n since more neighborhoods become more and more time-consuming. To sum up, the matheuristic is able to find solutions of high quality in a reasonable runtime.

7.5 Reducing the Number of Vehicles

Finally, we investigate the impact of resource management in the redistribution process. To this end, we determine the deterioration of solution quality by reducing the size of the vehicle fleet. We can thus observe whether a similar service level is obtained with fewer resources. In this context, comparing values of the objective function is difficult since they include the cost of unmet ride flow units and bike redistribution. Instead, we compare improvement ratios as well as redistribution volumes.

For each BSS, we run the matheuristic as before but impose a different number of vehicles. We test on the value of n which yields the better improvement ratio at the respective BSS. Table 8 shows the results for each input setting including the improvement ratio, total redistribution volume, and redistribution volume per vehicle. We observe that adding vehicles to the fleet increases the improvement ratio. Notice that the objective function is parametrized such that the cost per unmet ride flow unit is much higher than the redistribution costs. As a consequence, the additional vehicle reduces in most cases the total costs of unmet ride flow units, leading to solutions of superior quality.

For BA, we observe that an improvement ratio of 1.0 can be obtained with just one vehicle. The same behavior can be observed for NR. In HU and CA, we note that adding vehicles to the first vehicle really contributes to increase the improvement ratio. The increase rate of the improvement ratio tends to decrease with respect to the vehicle fleet size. The additional vehicle does not increase the improvement ratio, meaning that the redistribution effort is just distributed within more vehicles. In other words, each vehicle is less utilized in terms of operation time and redistribution volumes.



(a) improvement ratio (b) redistribution volume per vehicle

Figure 13: Effects of varying the number of vehicles for CA.

Figure 13 displays the results obtained for CA given different number of vehicles. We can clearly observe that increasing the vehicles from 1 to 3 is beneficial in this case because of the improvement ratio increases from 0.28 to 0.71 percent. However, the increase rate of the improvement ratio decreases by adding more vehicles to the fleet. Furthermore, we do not observe significant variations by adding the 9th vehicle to the fleet since. Regarding the redistribution volumes, we observe that the third vehicle is still suitably used. Beyond, we note that the redistribution volume per vehicle unit quickly decreases.

How to determine a suitable vehicle fleet size to perform redistribution? The answer depends on the goals of the operator. If the operator aims at maximizing service level regardless resource usage, a large number of vehicles is required. However, in most cases, a trade-off between redistribution costs and level of service is subject to negations between the municipality and the operator. We believe that the developed approach may be used as a tool to support such negotiations.

Table 8: Reducing the number of vehicles.

input setting			results		
BSS	n	vehicles	imp ratio	total vol.	vol. per vehicle
BA	200	1	1.00	104	104
		2	1.00	93	47
NR	200	1	1.00	41	41
		2	1.00	41	21
		3	1.00	41	14
HU	200	1	0.70	158	158
		2	0.97	225	113
		3	0.99	231	77
		4	1.00	231	58
CA	50	1	0.28	148	148
		3	0.71	430	143
		5	0.89	544	109
		7	0.94	571	82
		9	0.95	578	64

8 Conclusions

We introduced a novel service network design problem to address the redistribution process in bike sharing systems at the tactical planning level. Tactical planning exploits repetitive and predictable mobility patterns to produce regular master tours for redistribution operations. We explicitly integrate resource management in the formulation to produce an accurate representation of routing decisions, putting special emphasis on the necessary handling time at each station visit, as well as resource limitations to acquire and operate vehicles within the day.

The proposed MIP formulation is based on a time-expanded network representing redistribution and vehicle routing decisions in terms of flows. Addressing problem instances of this rich formulation is very challenging given the complex interplay between coexisting flows in the time-expanded network, the incorporation of vehicle-balance constraints to generate feasible master tours, and the fine time granularity required to represent handling operations.

Therefore, we propose two solution methods to address large problem instances. The first one is a construction heuristic which breaks down the decision-making process of bike redistribution into three phases. The respective phases successively determine FLAs, transport services, and master tours. The main virtue of the construction heuristic is that it quickly produces a good starting solution for the proposed matheuristic. Thus, the second solution method is a matheuristic which relies on a neighborhood-search scheme to find qualitative solutions within iterations. A neighborhood takes the form of a reduced MIP problem which involves a minor number of vehicle flow variables of the original problem instance. In this way, we can solve such reduced MIP problems using a solver. We introduced neighborhood operators which carefully choose promising transport services while considering randomization to diversify the search.

We provided evidence the proposed solution methods are able to produce solutions of good quality in a reasonable runtime. In addition, we have shown that the marginal benefits and utilization of vehicles decrease by enlarging the vehicle fleet size. In principle, fulfilling expected user demand is theoretically possible with a large number of vehicles, but impracticable due to excessive redistribution costs. We claim that service network design supports decision-makers to find an adequate trade-off between service level and resource usage.

Acknowledgements

This research has been supported by the German Research Foundation (DFG) through the Research Training Group SocialCars (GRK 1931). The focus of the SocialCars Research Training Group is on significantly improving the citys future road traffic, through cooperative approaches. This support is gratefully acknowledged. Partial funding for this project also comes from the Discovery Grant and the Discovery Accelerator Supplements Programs of the Natural Science and Engineering Research Council of Canada, and the Strategic Clusters program of the Fonds québécois de la recherche sur la nature et les technologies. The authors thank the two institutions for supporting this research. While working on this project, the second author was Adjunct Professor with the Department of Computer Science and Operations Research, Université de Montréal.

References

R. K. Ahuja, Ö. Ergun, J. B. Orlin, and A. P. Punnen. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, 123(1):75–102, 2002.

J. Andersen, T. G. Crainic, and M. Christiansen. Service network design with management and coordination of multiple fleets. *European Journal of Operational Research*, 193(2):377–389, 2009.

P. Angeloudis, J. Hu, and M. G. Bell. A strategic repositioning algorithm for bicycle-sharing schemes. *Transportmetrica A: Transport Science*, 10(8):759–774, 2014.

J. F. Bard and A. I. Jarrah. Large-scale constrained clustering for rationalizing pickup and delivery operations. *Transportation Research Part B: Methodological*, 43(5):542–561, 2009.

J. Brinkmann, M. W. Ulmer, and D. C. Mattfeld. Short-term strategies for stochastic inventory routing in bike sharing systems. *Transportation Research Procedia*, 10:364–373, 2015.

J. Brinkmann, M. W. Ulmer, and D. C. Mattfeld. Inventory routing for bike sharing systems. *Transportation Research Procedia*, 19:316–327, 2016.

M. Chouman and T. G. Crainic. Cutting-plane matheuristic for service network design with design-balanced requirements. *Transportation Science*, 49(1):99–113, 2014.

M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Maritime transportation. In C. Barnhart and G. Laporte, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 189 – 284. North-Holland, Amsterdam, 2007.

C. Contardo, C. Morency, and L.-M. Rousseau. Balancing a dynamic public bike-sharing system. Publication CIRRELT-2012-09, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Université de Montréal, Montréal, QC, Canada, 2012.

J. Corcoran, T. Li, D. Rohde, E. Charles-Edwards, and D. Mateo-Babiano. Spatio-temporal patterns of a public bicycle sharing program: the effect of weather and calendar events. *Journal of Transport Geography*, 41:292–305, 2014.

J.-F. Cordeau, P. Toth, and D. Vigo. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32(4):380–404, 1998.

T. G. Crainic. Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288, 2000.

T. G. Crainic and K. H. Kim. Intermodal transportation. In C. Barnhart and G. Laporte, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 467 – 537. North-Holland, Amsterdam, 2007.

T. G. Crainic and G. Laporte. Planning models for freight transportation. *European Journal of Operational Research*, 97(3):409–438, 1997.

T. G. Crainic, M. Hewitt, and W. Rei. Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design. *Computers & Operations Research*, 43 (2):90–99, 2014a.

T. G. Crainic, M. Hewitt, M. Toulouse, and D. M. Vu. Service network design with resource constraints. *Transportation Science*, 50(4):1380–1393, 2014b.

S. Datner, T. Raviv, M. Tzur, and D. Chemla. Setting inventory levels in a bike sharing network. *Transportation Science*, 2017. (article in advance).

C. M. de Chardon, G. Caruso, and I. Thomas. Bike-share rebalancing strategies, patterns, and purpose. *Journal of Transport Geography*, 55:22–39, 2016.

G. Erdoğan, M. Battarra, and R. W. Calvo. An exact algorithm for the static rebalancing problem arising in bicycle sharing systems. *European Journal of Operational Research*, 245(3):667–679, 2015.

H. M. Espegren, J. Kristianslund, H. Andersson, and K. Fagerholt. The static bicycle repositioning problem-literature survey and new formulation. In A. Paias, M. Ruthmair, and S. Voß, editors, *Computational Logistics*, pages 337–351, Cham, 2016. Springer International Publishing.

A. Faghih-Imani, N. Eluru, A. M. El-Geneidy, M. Rabbat, and U. Haq. How land-use and urban form impact bicycle flows: evidence from the bicycle-sharing system (BIXI) in montreal. *Journal of Transport Geography*, 41:306–314, 2014.

I. A. Forma, T. Raviv, and M. Tzur. A 3-step math heuristic for the static repositioning problem in bike-sharing systems. *Transportation Research Part B: Methodological*, 71: 230–247, 2015.

M. Hewitt, G. L. Nemhauser, and M. W. Savelsbergh. Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem. *INFORMS Journal on Computing*, 22(2):314–325, 2010.

S. C. Ho and W. Szeto. Solving a static repositioning problem in bike-sharing systems using iterated tabu search. *Transportation Research Part E: Logistics and Transportation Review*, 69:180–198, 2014.

A. Hoff, A.-G. Lium, A. Løkketangen, and T. G. Crainic. A metaheuristic for stochastic service network design. *Journal of Heuristics*, 16(5):653–679, 2010.

Institute for Transportation and Development Policy. The bike-sharing planning guide, 2014. Accessed November 15, 2017, https://www.itdp.org/wp-content/uploads/2014/07/ITDP_Bike_Share_Planning_Guide.pdf.

Intelligent Energy Europe. Optimising bike sharing in european cities-a handbook., 2011. Accessed November 15, 2017, https://ec.europa.eu/energy/intelligent/projects/sites/iee-projects/files/projects/documents/obis_handbook_en.pdf.

C. Kloimüllner, P. Papazek, B. Hu, and G. R. Raidl. Balancing bicycle sharing systems: An approach for the dynamic case. In C. Blum and G. Ochoa, editors, *Evolutionary Computation in Combinatorial Optimisation*, pages 73–84, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

C. Kloimüllner, P. Papazek, B. Hu, and G. R. Raidl. A cluster-first route-second approach for balancing bicycle sharing systems. In R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia, editors, *Computer Aided Systems Theory – EUROCAST 2015*, pages 439–446, Cham, 2015. Springer International Publishing.

A. A. Kumar, J. E. Kang, C. Kwon, and A. Nikolaev. Inferring origin-destination pairs and utility-based travel preferences of shared mobility system users in a multi-modal environment. *Transportation Research Part B: Methodological*, 91:270–291, 2016.

G. Laporte, F. Meunier, and R. Wolfler Calvo. Shared mobility systems. *4OR*, 13(4): 341–360, Dec 2015.

L. M. Martinez, L. Caetano, T. Eir, and F. Cruz. An optimisation algorithm to establish the location of stations of a mixed fleet biking system: An application to the city of Lisbon. *Procedia - Social and Behavioral Sciences*, 54:513 – 524, 2012.

V. Mrkajić and I. Anguelovski. Planning for sustainable mobility in transition cities: Cycling losses and hopes of revival in Novi Sad, Serbia. *Cities*, 52:66–78, 2016.

B. A. Neumann-Saavedra, P. Vogel, and D. C. Mattfeld. Anticipatory service network design of bike sharing systems. *Transportation Research Procedia*, 10:355–363, 2015.

J. Niehaus and W. Banzhaf. Adaption of operator probabilities in genetic programming. In J. Miller, M. Tomassini, P. L. Lanzi, C. Ryan, A. G. B. Tettamanzi, and W. B. Langdon, editors, *Genetic Programming*, pages 325–336, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

O. O'Brien, J. Cheshire, and M. Batty. Mining bicycle sharing data for generating insights into sustainable transport systems. *Journal of Transport Geography*, 34:262 – 273, 2014.

M. B. Pedersen, T. G. Crainic, and O. B. Madsen. Models and tabu search metaheuristics for service network design with asset-balance requirements. *Transportation Science*, 43 (2):158–177, 2009.

J. Pfrommer, J. Warrington, G. Schildbach, and M. Morari. Dynamic vehicle redistribution and online price incentives in shared mobility systems. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1567–1578, 2014.

M. Rainer-Harbach, P. Papazek, G. R. Raidl, B. Hu, and C. Kloimüllner. Pilot, grasp, and vns approaches for the static balancing of bicycle sharing systems. *Journal of Global Optimization*, 63(3):597–629, 2015.

T. Raviv, M. Tzur, and I. A. Forma. Static repositioning in a bike-sharing system: models and solution approaches. *EURO Journal on Transportation and Logistics*, 2 (3):187–229, 2013.

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

S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006.

J. Shu, M. C. Chou, Q. Liu, C.-P. Teo, and I.-L. Wang. Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems. *Operations Research*, 61(6):1346–1359, 2013.

N. Teypaz, S. Schrenk, and V.-D. Cung. A decomposition scheme for large-scale service network design with asset management. *Transportation Research Part E: Logistics and Transportation Review*, 46(1):156–170, 2010.

P. Vogel, T. Greiser, and D. C. Mattfeld. Understanding bike-sharing systems using data mining: Exploring activity patterns. *Procedia-Social and Behavioral Sciences*, 20: 514–523, 2011.

P. Vogel, B. A. Neumann Saavedra, and D. C. Mattfeld. A hybrid metaheuristic to solve the resource allocation problem in bike sharing systems. In M. J. Blesa, C. Blum, and S. Voß, editors, *Hybrid Metaheuristics*, pages 16–29, Cham, 2014. Springer International Publishing.

D. M. Vu, T. G. Crainic, and M. Toulouse. A three-phase matheuristic for capacitated multi-commodity fixed-cost network design with design-balance constraints. *Journal of Heuristics*, 19(5):757–795, 2013.

D. Zhang, C. Yu, J. Desai, H. Lau, and S. Srivathsan. A time-space network flow approach to dynamic repositioning in bicycle sharing systems. *Transportation Research Part B: Methodological*, 103:188 – 207, 2017.