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

February 2019

CIRRELT-2019-06

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. Bike sharing systems (BSSs) rely on daytime redistribution to provide bikes for rentals and free rack slots for returns in order to warrant a reliable service level to users. Tactical planning of BSSs exploits mobility demand forecasts to answer service level expectations through a cost-efficient use of redistribution vehicles. We propose a novel formulation of service network design that coordinates redistribution and vehicle routing decisions in space and time to produce regular master tours. This formulation explicitly integrates resource-management decisions by considering limited redistribution budget to acquire and operate vehicles, as well as an accurate time representation of pickups and deliveries of bikes at stations. We propose a matheuristic relying on a neighborhood search scheme to find solutions of good quality for real-world sized problem instances in reasonable time. To produce starting solutions, we propose a construction heuristic decomposing the daytime redistribution process into three phases: determine pickup and delivery activities, link pickups and deliveries into transport requests, and assign transport requests to master tours. To evaluate the operational performance of master tours, we propose a simulation approach in which stochastic mobility demand is gradually revealed over time. In the simulation, master tours are carried out as planed by service network design, whereas pickups and deliveries are adjusted to place a suitable number of bikes according to the stations' needs. In our computational study, we show that master tours improve the level of service in BSSs with high and regular mobility patterns, e.g., commuting activities.

Keywords: Bike sharing systems, daytime bike redistribution, service network design, matheuristic, simulation.

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.

*Revised version of CIRRELT-2018-17

** Corresponding author: TeodorGabriel.Crainic@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2019

1 Introduction

Station-based bike sharing systems (BSSs) have emerged as a cheap and sustainable approach for providing first-and-last-mile connections between transport hubs and points of interest (Ricci, 2015). To perform rides, users rent bikes at stations, and return them later to an empty rack slot of any station. Operators of BSSs strive to offer a reliable service level so that users perform bike rides as demanded. To accomplish this objective, operators must deal with stations running full and empty several times during a day because of limited bike racks at stations and spatio-temporal dynamics of mobility demand (de Chardon et al., 2016). Bike redistribution is a fundamental activity to place bikes and empty rack slots whenever requested by users (Shu et al., 2013). To this end, operators require vehicles for station visits and driver-rebalancers for picking up and delivering bikes, also called *handling activities*.

Several studies analyzing recorded bike rides provide evidence that mobility demand in cities is usually stable over a season (O'Brien et al., 2014; Corcoran et al., 2014; Vogel et al., 2011). For example, commuters typically ride to work in the morning and back home in the afternoon, whereas in hilly cities users typically ride from top to bottom of a hill. Consequently, the times of day at which high-demanded stations run full or empty can be predicted with a high degree of confidence. These insights allow for meaningful forecasts to be used for planning daytime redistribution. Tactical planning for BSSs exploits these forecasts to design and assign resources to regular *master tours* that are operated day after day, defining when to visit stations and how long to perform handling activities.

Designing master tours requires an efficient management of resources to improve the level of service without damaging the economical viability of the operators. The monetary budget for bike redistribution is typically scarce since user-based business models lead to low revenues from bike rentals (Intelligent Energy Europe, 2011). In addition, next to the driving time of vehicles between stations, assigning sufficient handling time to station visits is of utmost importance as it covers a significant part of driver-rebalancer's working hours (de Chardon et al., 2016). This paper is the first attempt of considering these resource management issues explicitly at the tactical planning level. To this end, we follow advances in transport planning (e.g., Andersen et al. (2009); Crainic et al. (2014)) to appropriately coordinate vehicle routing and redistribution decisions in space and time by introducing a novel formulation of *service network design for bike sharing systems (SNDBSS)*. This formulation is based on a time-expanded network that explicitly considers driving and handling times as well as budget limitations for acquiring vehicles and designing master tours.

Obtaining solutions of good quality for SNDBSS instances is far from being trivial because of the high interdependence among master tours, handling activities, and mobility demand forecasts. Therefore, we design a construction heuristic for producing starting solutions in elaborate sequential steps. To improve this solution, we adopt the neighborhood search scheme of Hewitt et al. (2010) and introduce a matheuristic that carefully fixes variables associated with master tour decisions, and then finds a solution for the resulting reduced MIP problem using a solver. Computational experiments suggest the effectiveness of tailoring this neighborhood search scheme to SNDBSS.

When master tours are implemented, their operational performance is subject to a stochastic mobility demand which is gradually revealed over time. Because of day-to-day demand variations, the observed distribution of bikes when vehicles are on the road may differ from the determined one by SNDBSS. Although we stick to the execution of master tours, handling activities require adjustments with respect to the observed bike distribution and expected stations' needs. In this paper, we evaluate the operational performance of master tours with a discrete-event simulation mimicking the implementation phase.

This paper makes the following contributions. First, we introduce a new and comprehensible formulation of SNDBSS integrating resource management decisions into tactical planning. Second, we present a solution methodology combining exact and heuristic techniques to produce SNDBSS solutions of high quality in a reasonable runtime. Third, we evaluate master tour performance in an operational setting and discuss managerial implications. Our experiments show that although differences between mobility demand forecasts and observations exist, master tours obtained from tactical planning are able to improve the level of service of BSSs with high commuting activity.

The paper is organized as follows. Section 2 reviews the existing literature in BSSs and resource management. Section 3 provides a statement of the problem studied. Section 4 introduces the time-expanded network formulation. The solution methodology is presented in Section 5. Section 6 introduces the simulation approach. In Section 7, we report on the computational experiments. Finally, conclusions are given in Section 8.

2 Literature Review

BSSs have been a source of intensive research in the last decade. The literature about BSSs considers planning at strategic, tactical, and operational levels to provide a reliable service level in a cost-efficient manner, see for example Laporte et al. (2018).

Strategic planning determines the number, location, and capacity of stations over long-term horizons (Lin and Yang, 2011; García-Palomares et al., 2012; Lin et al., 2013). These strategic decisions are subject to high implementation costs as well as land-use regulation (Institute for Transportation and Development Policy, 2014). Approaches for bike redistribution assume these strategic decisions remain unalterable for the subordinate planning levels.

Operational planning determines vehicle routing and redistribution decisions over short-term horizons, for example, one day. We distinguish here between overnight and daytime redistribution. Approaches for overnight redistribution assume mobility demand is negligible in those hours. The goal is to achieve a suitable distribution of bikes among stations for the beginning of the daytime (Raviv et al., 2013; Rainer-Harbach et al., 2015; Ho and Szeto, 2014; Erdoğan et al., 2015; Freund et al., 2016; Forma et al., 2015; DellAmico et al., 2018; Espegren et al., 2016). Operational approaches for daytime redistribution consider a stochastic and dynamic environment in which decisions are made based on current observations and short-term demand forecasts. Pfrommer et al. (2014)

study the trade-off between the costs of offering incentives for returning shared vehicles at low-demanded stations and the costs of redistribution activities. Brinkmann et al. (2015) propose a single-vehicle inventory routing problem and myopic policies for determining decisions based on safety stocks of bikes and empty rack slots. In Brinkmann et al. (2018), look-ahead policies derived from approximate dynamic programming techniques are proposed to anticipate future shortages of bikes and empty racks slots. Ghosh et al. (2017) and Shui and Szeto (2018) adapt overnight redistribution approaches to address daytime redistribution in a rolling horizon framework.

Tactical planning aims to design master tours for daytime redistribution over mid-term horizons, for example, a season. Although tactical planning of BSSs has gained more attention in recent years, existing approaches make simplified assumptions with respect to resource management. Contardo et al. (2012) present a MIP formulation on a time-expanded network, in which each station and time period involves a demand of bikes or empty rack slots. Bikes are artificially added or removed at stations if redistribution does not suffice to fulfill demand; the objective is to minimize these artificial activities. Kloimüllner et al. (2014) adapt the formulation for overnight redistribution presented in Rainer-Harbach et al. (2015) to cope with tactical planning, but do not consider handling time accurately. Vogel et al. (2014) propose a MIP formulation determining time-dependent bike transports among stations to minimize shortages of bikes and empty rack slots. As the formulation does not assign bike transports to master tours, the lack of synchronization may lead to an excessive use of resources. Brinkmann et al. (2016) propose a multi-vehicle inventory routing problem with deterministic demands, aiming at minimizing the gap between the actual and desired distribution of bikes. Redistribution costs are not taken into account. In Angeloudis et al. (2014), a redistribution plan is produced in two steps. The first step determines master tours by solving a multi-vehicle traveling salesman problem. The second step determines redistribution quantities with a flow assignment model. This work does not consider an accurate representation of handling time. Kloimüllner et al. (2015) introduce a cluster-first route-second approach with the strong assumption that operators only address full and empty stations. Zhang et al. (2017) introduce a stochastic non-linear formulation to take redistribution decisions. They embed the formulation into a rolling horizon approach to determine vehicle routes within a certain time interval. Notice, that none of these papers evaluate the performance of tactical plans within their implementation.

We consider research advances in service network design for freight transportation as blueprints for integrating resource management to BSSs' tactical planning. Resource management has become relevant in the research community as it leads to more consistent transportation plans requiring less adjustments when they are implemented. Pedersen et al. (2009) introduce vehicle-balance constraints to ensure that the inflow and outflow of resources is balanced at each location and point in time. Crainic et al. (2014) introduce a service network design problem in which resources at terminals are costly, limited, and guided by operational rules. Zhu et al. (2014) introduce a service network design formulation for freight rail transportation which allows for double consolidation by grouping cars into blocks that are then grouped to make up trains. Service network design formulations are typically built on time-expanded networks to represent time-dependent decisions. These formulations are in most cases computationally intractable using exact methods because finding feasible solutions with respect to resource management issues is far from being trivial. The literature reports solution methodologies based on matheuristics to produce solutions of good quality in reasonable runtime (Vu et al., 2013; Chouman and Crainic, 2014; Teypaz et al., 2010).

3 Problem Statement

We adopt the standard problem setting used in the literature for daytime redistribution in station-based BSSs (Laporte et al., 2018). This setting considers a set of stations, one vehicle depot without bike storage facilities, users requesting bikes and empty rack slots, a homogeneous fleet of vehicles performing master tours, and driver-rebalancers picking up and delivering bikes. The goal of bike redistribution is to minimize user dissatisfaction by providing sufficient bikes and empty rack slots at each station. User dissatisfaction is directly related with full and empty stations as they lead to *failed rental requests* and *failed return requests*, respectively. A failed rental request occurs if a user cannot rent a bike at an empty stations. In turn, a failed return request takes place if a user cannot return a bike at a full station.

We build on the work of Datner et al. (2017) to model mobility demand and measure user dissatisfaction. Mobility demand is thus defined in terms of users' *demanded bike rides*. A demanded bike ride brings a user directly from its place of origin to its place of destination (for the sake of simplicity, these places coincide with the location of stations). A demanded ride can be performed if at least one bike is available at the station of origin and at least one empty rack is available at the station of destination. Notice that there exists an interdependence among stations because of failed rental requests and failed return requests. If no bike is available at the station of origin, a user may walk to another station to request a bike. If the user did not rent a bike and refrains from using the BSS to achieve its destination, the return request is then eliminated at the station of destination. If no empty rack slot is available at the station of destination, the user must roam among stations to end the ride. In these cases, a *journey* defines the route a user makes to achieve its destination.

Assuming that users aim to minimize their journeys' travel times, we adopt the journey dissatisfaction function (JDF) as a metric of service level. The JDF returns a non-negative value with respect to the user's performed journey and demanded bike ride (henceforth called demanded journey as well). The demanded journey returns a JDF value of zero, whereas any other journey returns a positive JDF value because of the dissatisfaction caused by roaming among stations to start or end a bike ride. The JDF value is computed based on the excess travel time, namely the difference of travel times involving the performed journey and the demanded bike ride. As in Datner et al. (2017), we assume that journeys' travel times are calculated based on users' walking and riding times. The authors emphasis that the JDF can be adapted to measure other sources of dissatisfaction, for example, the "cost" of using alternatives modes of transportation to achieve a destination.

A *station state* defines the number of bikes which are currently available at the station. As the capacity of a station is limited, there exists an interdependency between the number of bikes and empty racks slots in it; one placed bike for rentals blocks one rack slot for returns. The number of bikes in the BSS remains fixed so that a bike is either placed in a station, loaded on a vehicle, or used to perform a bike ride. The station state varies over time because of rentals and returns of bikes as well as handling activities. We define *station state alterations (SSA)* as a station visit and a bike quantity to pick up or deliver. A *transport service* defines the movement of a loaded vehicle between two stations.

At the tactical planning level, SNDBSS determines the SSAs to implement and the transport services to define time-ordered sequences of SSAs, which are assigned to master tours. A monetary budget limits the acquisition of vehicles and the design of master tours. The objective is to minimize the overall JDF value (OJDFV) based on a mobility demand forecast. Under the assumption that mobility demand displays highly regular patterns, one can rely on deterministic demand to determine with a high degree of confidence when stations require redistribution. For instance, Figures 1a and 1b show the station states for New York's Citibike BSS in the morning and evening, respectively, for a typical summer weekday. Each circle depicts a single station; the bigger the circle, the more bikes in racks at the respective station. We see that bikes are accumulated in working districts during the morning and residential districts during the evening. Because of the regularity of commuting rides, a similar distribution of bike is observed for different working days. Nevertheless, during the implementation phase, one needs to adjust handling activities based on day-to-day variations of the bike distribution. Tactical planning can thus at best provide guidelines to pick up or deliver bikes during the implementation phase. In this work, we show that this procedure can contribute to improve the service level in BSSs with high dominance of commuting rides.

4 Mathematical Formulation of SNDBSS

In this section, we introduce a mathematical formulation for SNDBSS. The mathematical formulation aims to produce master tours. We first introduce the general notation, then we present the mathematical formulation for SNDBSS, which takes the form of a MIP.

4.1 General Notation

Let $S_0 = \{0, 1..., S_{MAX}\}$ be the set of locations. Location 0 represents the vehicle depot and locations in $S = S_0 \setminus \{0\}$ represent stations. Each station $i \in S$ has a capacity of c_i bike racks; the depot has no bike racks. The daytime redistribution occurs within a workday length denoted by $T = \{0, 1, ..., T_{MAX}\}$. The workday length is divided into chronological and equal-length time periods $t \in T$. Let b define the number of bikes in the BSSs. We assume that all bikes are placed at a station in time periods 0 and T_{MAX}.

(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. Reference: http://bikes.oobrien.com/

We introduce two set of nodes to represent the interplay between stations and vehicles with respect to handling activities. Let $\mathcal{N}_S$ be the set of rack nodes. A rack node $(i,t) \in \mathcal{N}_S$ encodes the number of bikes in racks at station i and time period t. In addition, let $\mathcal{N}_V$ be the set of vehicle nodes. A vehicle node $(\widehat{i,t}) \in \mathcal{N}_V$ encodes the number of bikes in a vehicle if it parks at station i and time period t. In this way, bike flows from rack nodes to vehicle nodes represent pickups, whereas bike flows from vehicle nodes to rack nodes represent deliveries.

The mobility demand forecast is denoted by $\mathbf{d} = (U, \bigcup_{u \in U} Q_u)$. Let U be the set of users. Each $u \in U$ is associated with a set of journeys Q_u. Every journey $q \in Q_u$ brings user u from its station of origin i to its station of destination j. Each journey q involves a travel time T_q. Let $q^d \in W_u$ be the demanded journey of user u. The user's realized journey is denoted by $q^r \in Q_u$, with $T_{q^d} \leq T_{q^r}$.

In order to represent variations of station states due to user requests, we define Θ^+_{it} and Θ^-_{it} as the set of all user-journey tuples (u,q) which respectively end and start the bike ride at station i and time period t.

The vehicle link set $\mathcal{A}_V$ permits to model master tour decisions. 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 for the decision that one vehicle moves between two locations. 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 vehicle stays at a station to either pick up or deliver bikes during one time period.

We consider redistribution flows to encode either transport services or handling activities. We denote the redistribution link set by $\mathcal{A}_R$. This set consists of the transport service set $\mathcal{A}_T$, the pick-up link set $\mathcal{A}_P$, and the delivery link set $\mathcal{A}_D$. A transport service link $a \in \mathcal{A}_T$ connects two vehicle nodes from different stations to move a bike quantity from one station to another. To depict handling activities, we define the pick-up link set

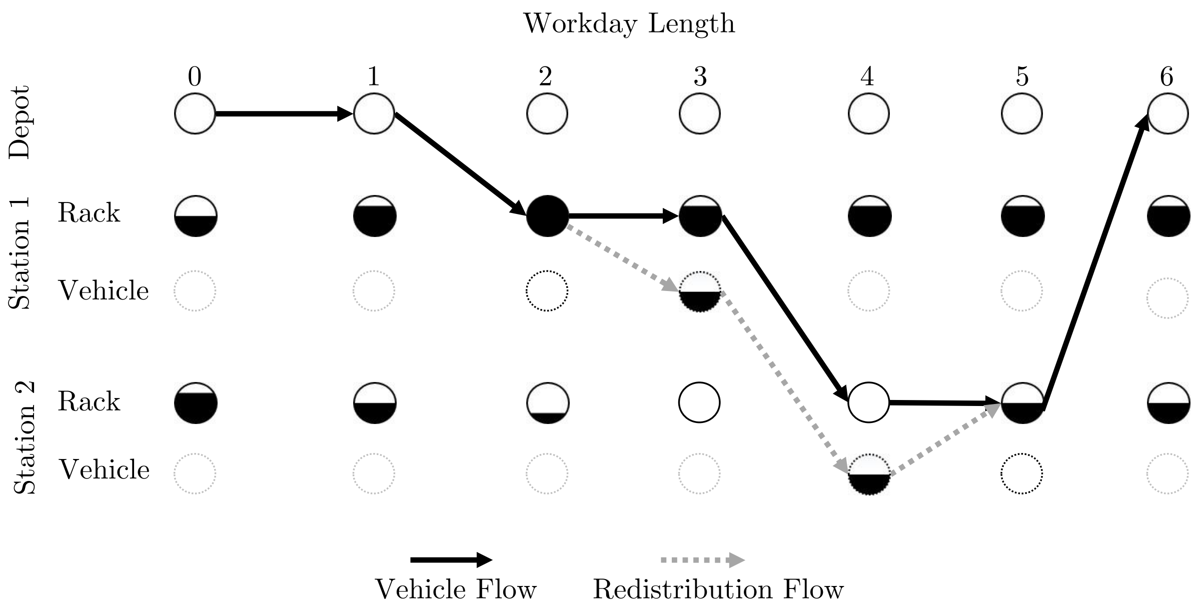

Figure 2: The time-expanded network of SNDBSS.

$\mathcal{A}_P$ and the delivery set $\mathcal{A}_D$. If a handling link $e \in \mathcal{A}_{HA}$ is used, either the corresponding pick-up link $a \in \mathcal{A}_P$ allows a bike flow 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 $e \in \mathcal{A}_{HA}$ by $\mathcal{P}(e)$ and $\mathcal{D}(e)$, respectively. Notice that pickup and delivery activities must not simultaneously occur in a station and time period. 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 $\alpha_e, e \in \mathcal{A}_{HA}$. 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 2, 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 become full and empty, respectively, due to mobility demand that is not shown for the sake of illustration. A master tour is depicted as a schedule of vehicle flows. First, the vehicle stays at the depot for one period, which is represented by a holding vehicle link. Then, the vehicle requires one time period to arrive at station 1. Notice that the vehicle parks at station 1 for one time period. Here, a pickup flow encodes that bikes are moved from the station to the vehicle, thus altering the fill of the respective rack and vehicle nodes. After that, the loaded vehicle drives to station 2. Again, the vehicle stays there for one time period to move bikes from the vehicle to the station. At the end of the work day, the vehicle must return to the depot.

Each vehicle has a load capacity of C bikes. If a handling link $e \in \mathcal{A}_{HA}$ is used, a maximum of μ_e bikes can be picked up or delivered through it. Let $\mathcal{R}(e)$ be the set of redistribution links that can be used whether vehicle link $e \in \mathcal{A}_V$ is set. Let $\mathcal{A}^+(i,t)$ and $\mathcal{A}^-(i,t)$ represent the outgoing and incoming link set at node (i,t), respectively.

Let β_{uq} be the JDF value returned by journey q realized by user u. The sum of all these values is defined as the overall JDF value (OJDFV). The objective is to minimize the OJDFV.

With respect to redistribution costs, let F be a fixed cost of using a vehicle unit. We define k_e as the fixed costs to set a link $e \in \mathcal{A}_{VM} \cup \mathcal{A}_{HA}$. Costs are limited by a given budget B.

4.2 The MIP

We now define the variables of the formulation:

- I_i^t: bikes at rack node $(i,t) \in \mathcal{N}_S$
- w_i^t: bikes at vehicle node $(\hat{i,t}) \in \mathcal{N}_V$
- v: vehicle fleet size
- z_{uq} : one whether user u performs journey q, zero otherwise
- y_e: quantity of vehicle flow assigned to link $e \in \mathcal{A}_V$
- x_a: quantity of redistribution flow assigned to link $a \in \mathcal{A}_R$
- $\alpha_e \in \mathcal{A}_{HA}$: one whether picking up bikes is permitted though link $a \in \mathcal{P}(e)$, zero otherwise

With the notation, the mathematical formulation $\mathcal{F}$ for redistribution planning given mobility demand forecast $\mathbf{d}$ reads as follows:

$$\min \{ \sum_{u \in U} \sum_{q \in Q_u} \beta_{uq} z_{uq} \} \tag{1}$$

s.t.

$$\sum_{q \in Q_u} z_{uq} = 1, \quad \forall u \in U \tag{2}$$

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

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

$$I_i^{t+1} = I_i^t - \sum_{a \in \mathcal{A}_P^+(i,t)} x_a + \sum_{a \in \mathcal{A}_D^-(i,t+1)} x_a - \sum_{(u,q) \in \Theta_{it}^+} z_{uq} + \sum_{(u,q) \in \Theta_{i,t+1}^-} z_{uq}, \quad \forall (i,t) \in \mathcal{N}_S : i \in S, t < \mathcal{T} \tag{5}$$

$$\sum_{a \in \mathcal{A}_P^+(i,t)} x_a + \sum_{(u,q) \in \Theta_{it}^+} z_{uq} \leq I_i^t, \quad \forall (i,t) \in \mathcal{N}_S : i \in S, t < \mathcal{T} \tag{6}$$

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

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

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

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

$$M\big(1 - \sum_{e \in \mathcal{A}_T^+(\hat{i,t})} y_e\big) \geq \omega_i^{t+1}, \quad \forall (\hat{i,t}) \in \mathcal{N}_0 : t \neq \{0, \mathcal{T}\} \tag{11}$$

$$\sum_{a \in \mathcal{R}(e)} x_a \leq C y_e, \quad \forall e \in \mathcal{A}_{VM} \cup \mathcal{A}_{HA} \tag{12}$$

$$x_a = 0, \quad \forall a \in \mathcal{A}_{VM} : i(a) = 0 \vee j(a) = 0 \tag{13}$$

$$x_a \leq \mu_e(1 - \alpha_e), \quad \forall e \in \mathcal{A}_{HA} : a \in \mathcal{P}(e) \tag{14}$$

$$x_a \leq \mu_e \alpha_e, \quad \forall e \in \mathcal{A}_{HA} : a \in \mathcal{D}(e) \tag{15}$$

$$\sum_{e \in \mathcal{A}_V^+(\hat{i,t})} y_e = \sum_{p \in \mathcal{A}_V^-(\hat{i,t})} y_e, \quad \forall i \in S, t \in T : t \neq \{0, \mathcal{T}\} \tag{16}$$

$$\sum_{e \in \mathcal{A}_V^+(\hat{i,t})} y_e \leq 1, \quad \forall (i,t) \in \mathcal{N}_S : t \neq \{0, \mathcal{T}\} \tag{17}$$

$$\sum_{e \in \mathcal{A}_V^+(0,0)} y_e = \sum_{p \in \mathcal{A}_V^-(0,\mathcal{T})} y_e = v \tag{18}$$

$$Fv + \sum_{e \in \mathcal{A}_{VM} \cup \mathcal{A}_{HA}} k_e y_e + \sum_{a \in \mathcal{A}_P \cup \mathcal{A}_D} q_a x_a \leq B \tag{19}$$

$$I_i^t \in \mathbb{Z}^+, \quad \forall (i,t) \in \mathcal{N}_S \tag{20}$$

$$\omega_i^t \in \mathbb{Z}^+, \quad \forall (\hat{i,t}) \in \mathcal{N}_V \tag{21}$$

$$x_a \in \mathbb{Z}^+, \quad \forall a \in \mathcal{A} \backslash \mathcal{A}_{HO} \tag{22}$$

$$z_{uq} \geq \{0,1\}, \quad \forall u \in U, \forall q \in Q_u \tag{23}$$

$$y_e \geq \mathbb{Z}^+, \quad \forall e \in \mathcal{A}_{HO} \tag{24}$$

$$y_e \geq \{0,1\}, \quad \forall e \in \mathcal{A}_V \backslash \mathcal{A}_{HO} \tag{25}$$

$$\alpha_e \geq \{0,1\}, \quad \forall e \in \mathcal{A}_{HA} \tag{26}$$

$$v \in \mathbb{Z}^+ \tag{27}$$

The objective function (1) minimizes the OJDFV associated to users's performed journeys. Equation (2) makes sure that each user performs one journey. 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. Equations (10-11) ensure that the vehicle node can have bikes only if it is visited by a vehicle. Equation (12) links master tours with bike flows. Bikes can be moved neither from nor to the depot (13). Equations (14) and (15) ensure that bikes are not simultaneously picked-up and delivered at a station and time period. The vehicle-balance constraints are established in (16). Equation (17) prohibits that two or more vehicles park simultaneously at one station. Equation (18) ensures that all vehicles start and end master tours at the depot. Equation (19) imposes budget limitations. Finally, the variable domain is established in (20 - 27).

5 Solution Methodology for SNDBSS

Producing high-quality solutions is challenging because of the management of interdependent resources. As stated in Andersen et al. (2009), vehicle-balance constraints (16) cause slow convergence of commercial solvers. In addition, Equation (19) that limits the acquisition and use of vehicles to a certain budget capacity is hard to handle because of its combinatorial nature.

In Section 5.1, we adopt the matheuristic proposed in Hewitt et al. (2010) by tailoring the neighborhood search scheme to the SNDBSS. In Section 5.2, we propose a construction heuristic to provide the matheuristic with a starting solution of reasonable quality.

5.1 Matheuristic

A high-level description of the matheuristic is depicted in Algorithm 1. We define an operator to be a selection method of vehicle flow variables. In each iteration, operators free a minor part of the vehicle flow variables resulting in a reduced MIP problem. A solution of the reduced MIP problem is then produced using a commercial solver. The operators choose vehicle flow variables describing feasible detours of existing master tours in the incumbent solution. A detour is defined by the time period where a vehicle leaves its assigned master tour, a time-ordered sequence of station visits within the detour (including the handling time at each station visit), and the time period in which the vehicle returns to the original master tour. So a reduced MIP problem contains *active* vehicle flow variables defining master tours in the incumbent solution as well as *candidate* vehicle flow variables defining detour opportunities. We propose three operators: one for linking stations with overflow of bikes to stations with shortages of bikes for quick improvements of the OJDFV, and two for providing diversification to the search. Over iterations, we update probabilities for the execution of each operator according to its contribution in finding solutions of better quality.

Algorithm 1 High-level description of the matheuristic

```
Require: SNDBSS problem, starting solution
Set incumbent solution = starting solution
while Runtime limit condition is not met do
    Free vehicle flow variables to be candidate using operators
    Solve reduced MIP problem using a solver
    if solution found by solving reduced MIP problem is better than incumbent solution then
        Set incumbent solution = solution found by solving reduced MIP problem
    end if
    Set probabilities for using operators
end while
```

The matheuristic builds and solves reduced MIP problems until a termination condition is met. To speed up the search, we follow the two-step approach introduced by Raviv et al. (2013). Within the iterative process, variables related to Equations (18)-(21) are addressed as continuous variables (the first step). Once the iterative process concludes, vehicle flow variables are fixed to their value in the incumbent solution, and we produce the final solution using a commercial solver, with all variables taking integer values this time (the second step). As shown in Raviv et al. (2013), a solver can produce an integer solution quickly in the second step with minor deterioration of solution quality.

In Section 5.1.1, we describe the matheuristic operators to be used for building reduced MIP problems. In Section 5.1.2, we introduce the procedure for assigning probabilities to apply these operators.

5.1.1 Operators for Defining Detours in Reduced MIP Problems.

Figure 3 illustrates possible detours from existing master tours. For the sake of simplicity, we only depict rack nodes and vehicle links. Solid lines depict active vehicle links for master tours, whereas dashed lines depict candidate vehicle links for detours. We observe that candidate vehicle links define two possible detours. The first detour arrives at station 2 in the second time period, stays one time period, and then returns to the original master tour to station 1 in the fourth time period. The second detour arrives at station 4 in the fourth time period, stays one period, and then returns to the original master tour to station 1 in the sixth time period. Notice, that a detour may involve several station visits as long as it returns to the original master tour.

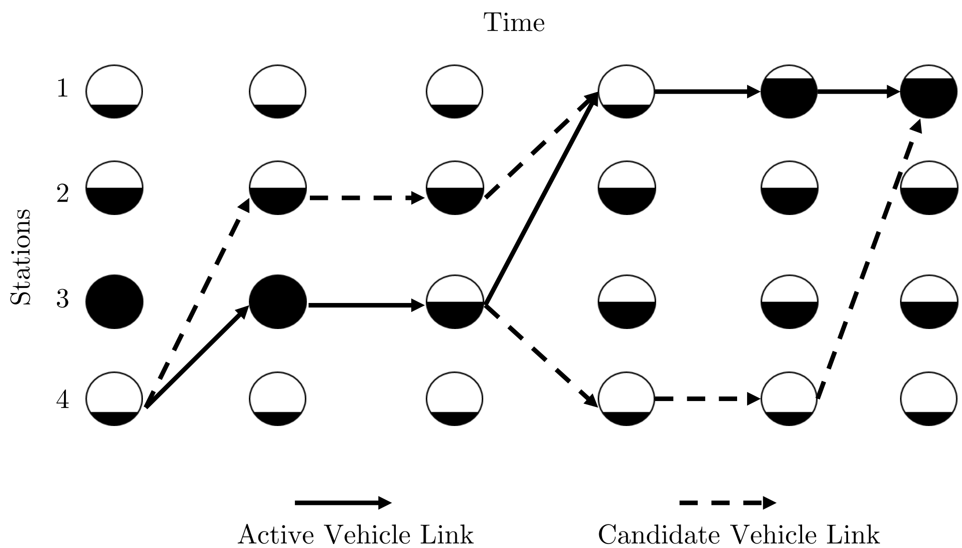

Figure 3: Feasible detours derived from an incumbent solution.

Each operator defines a detour in two generic steps. The first step determines a sub-path connecting non-active rack nodes. Here, the sub-path involves up to two station visits within the detour, but this can be easily extended to more station visits. The assigned handling time at each station visit is subject to the operator applied. Let (i_1, t_1) be a non-active rack node chosen by the operator. Let $\bar{t}_1$ be the integer number of time periods a vehicle stays at station i_1. So, the rack node sequence $(i_1, t_1), (i_1, t_1 + 1), ..., (i_1, t_1 + \bar{t}_1)$ depicts the first station visit. The first step concludes if the detour only involves one station visit. Otherwise, let (i_2, t_2) represent a non-active rack node where the second station visit begins. This rack node is selected so that $(i_1, t_1 + \bar{t})$ and (i_2, t_2) can be connected by a feasible vehicle link. Handling time $\bar{t}_2$ is then assigned for the second station visits as done for the first one before. Thus, the generated sub-path involving two station visits is given by $(i_1, t_1), (i_1, t_1 + 1), ..., (i_1, t_1 + \bar{t}_1), (i_2, t_2), (i_2, t_2 + 1), ..., (i_2, t_2 + \bar{t}_2)$.

The second step connects the generated sub-path with one master tour in the incumbent solution. More specifically, a feasible vehicle link must connect an active node of a master tour with (i_1, t_1), and a feasible vehicle link must connect $(i_2, t_2 + \bar{t}_2)$ with an active rack node of the master tour. Notice that connecting the generated sub-path with all master tours is possible will lead to a very large MIP problem. Instead, the sub-path is connected with only one master tour according to the costs k_e associated with leaving and returning to the original master tour. With a roulette wheel selection criterion, we ensure that detours of lower cost are heuristically chosen with higher probability.

We propose three operators to choose sub-paths:

- 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 redistributing a large number of bikes.

RPO permits vehicle detours through different stations in a randomly chosen time period t_{random}. Each vehicle detour consists of: 1) a candidate vehicle movement link connecting one active rack node with one non-active rack node $(i, t_{random}) \in \mathcal{N}_S$, 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 $(i_{random}, t) \in \mathcal{N}_S$. 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 maximum of ϱ vehicle detour possibilities. The FBO conducts a vehicle detour as follows. First, the 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 such 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 a feasible detour. 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, that is, $(1 - I_i^t)/c_i, \forall(i,t) \in \mathcal{N}_S$.

5.1.2 Defining a Suitable Mix of Operators

Given the characteristics of the three operators, FBO will outperform RPO and RSO in early iterations of the matheuristic since it can identify promising transport services involving large redistribution quantities. However, FBO loses its effectiveness in later iterations because it can hardly improves a solution made of fully loaded vehicles. Randomization provided by RPO and RSO helps escaping from local optima. However, there still exists a chance that FBO finds solutions of superior quality after escaping from local optima. We utilize these observations by changing the relative frequency of applying operators over iterations.

A suitable mix of operators can guide an effective exploration of the search space even with small reduced MIP problems, thus reducing the computational effort. Let n denote the total number of detours chosen to build a reduced MIP problem, that is, $n = \varpi + \varphi + \varrho$. Given a value of n, we dynamically adjust ϖ, φ, and ϱ according to the performance

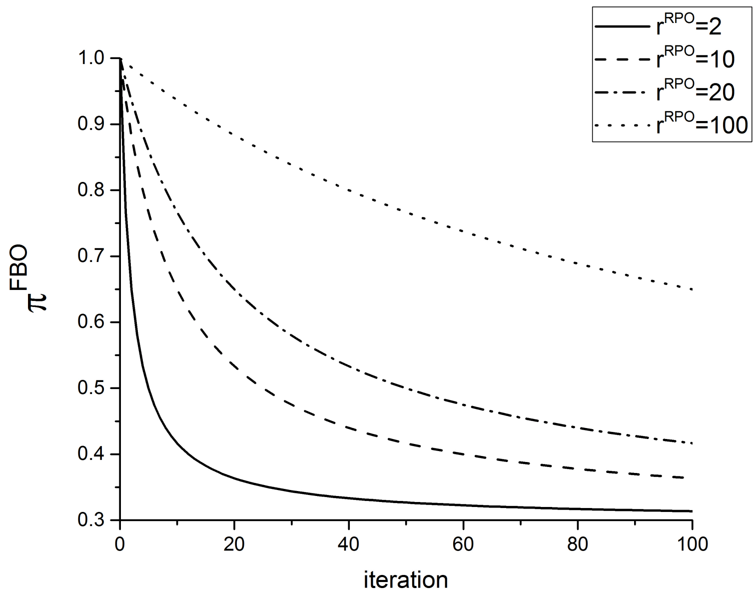

Figure 4: Behavior of π^{FBO} for different values r^{FBO}, with $(1 - \Theta^{FBO}) = 0.3$.

of operations in previous iterations. We derive probabilities that control the number of vehicle detours chosen by each operator. The probabilities depend on the success of the corresponding operator, that is, the number of candidate vehicle flow variables which becomes active. After defining the probabilities, a roulette wheel selection determines how many detours are produced by each 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 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 operators. We define the probability $\pi^o \in [0,1]$ for choosing a detour using the operator o. In iteration p, the operator o chooses a number of c^o candidate vehicle links to build the reduced MIP problem. 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 decreases too fast. In addition, let $\Theta^o \in [0,1]$ be a parameter that avoids that the probability of 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:

$$\pi^{FBO} := 1 - \frac{\Theta^{FBO} p}{r^{FBO} + p} \tag{28}$$

In Equation (28), 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 4 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):

$$\pi^o := (1 - \pi^{FBO})\big(\Theta^o + r^o \frac{1 - 2\Theta^o}{\Gamma}\big), \quad \forall o \in \{RPO, RSO\} \tag{29}$$

In Equation (29), 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 operator.

5.2 Construction Heuristic

The construction heuristic aims to determine a starting solution of reasonable quality. Finding feasible solutions of SNDBSS instances is far from being trivial due to the strong interactions among variables representing station states, bike redistribution, and master tours. The construction heuristic considers three phases in a top-down way:

- The first phase minimizes the number of SSAs satisfying the mobility demand forecast on a per-station basis.
- The second phase matches suitable pickup SSAs with delivery SSAs to obtain transport requests.
- The third phase assigns transport requests to vehicles by means of a pick up and delivery problem with time windows (PDPTW), thus determining the vehicle fleet size as well as master tours.

Figure 5 sketches the top-bottom information flow between phases. Notice, that the demand satisfied by SSAs determined in the first phase can be hardly met in the subordinate phases. In order to *quickly* obtain a starting solution for the matheuristic, we refrain from bottom-top information flows and thereby refinements of SSAs. In the following, we describe every phase and its outcome.

The outcomes of the first phase are SSAs in terms of bike quantities r_i^t to handle at node (i, t) representing station i and time period t. A positive value of r_i^t indicates a pickup. A negative value of r_i^t signals a delivery. In the first phase, SSAs satisfy bike rentals and returns neglecting station interdependencies. In Figure 5, we depict SSAs by black circles with a "P" for pickups or with a "D" for deliveries. We define a procedure to determine the set of r_i^t by station. This procedure consists of a triggering step and an adjusting step that are successively executed over iterations. Let I_i^0 be a given initial state of station i. The procedure starts in time period $t^{START} := 0$. The triggering phase determines a set of r_i^t in the time interval $[t^{START}, \mathcal{T}]$ using recursive equations. These recursive equations compute variations of station states I_i^t due to demand and, if necessary, triggers an SSA to avoid that a station i becomes full or empty in time

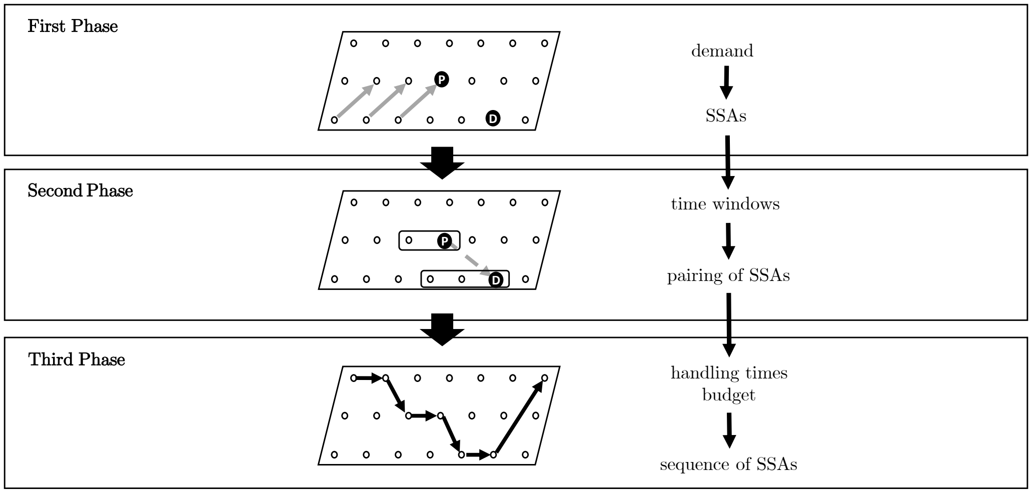

Figure 5: Framework of the construction heuristic.

period t. The procedure to triggers SSAs is formally stated in Appendix 9.1. Then, the adjusting phase proceeds as follows. Let t^{FIRST} be the time period in which the first SSA is determined within the triggering phase in $[t^{START}, \mathcal{T}]$. Then, the bike quantity $r_i^{t^{FIRST}}$ is increased or decreased so that future SSAs at this station are avoided. For example, if two pickup SSAs are triggered each time a station runs full, we may omit the second SSA by picking up more bikes in the first SSA. The procedure to adjust bike quantities of SSAs is described in Appendix 9.2. Notice that the adjusting phase is not executed if less than two SSAs triggered at station i in $[t^{START}, \mathcal{T}]$ as no more SSAs can be omitted. An iteration concludes by setting $t^{START} := t^{FIRST} + \bar{t}$, with $\bar{t}$ as a minimum time periods separating two SSAs. So in the next iteration, all SSAs in the time interval $[0, t^{START} - 1]$ remain fixed.

The second phase generates transport requests. The aim is to match pickup SSAs with delivery SSAs constrained by space, time, and bike quantities. There may be cases in which a pickup SSA cannot be matched with a delivery SSA as they occur in the same time period. We overcome this problem by designing time windows for which an SSA can be performed earlier or later than determined in the first phase. Given an node (i, t) in which an SSA has been determined, we replicate nodes on the corresponding station i, one at each time period between $t - \delta$ and t. Thus, δ defines the time window length. The SSA can be performed at one node in the set $\{(i, t - \delta), (i, t - \delta + 1), ..., (i, t)\}$.

For each SSA, we determine δ as follows. We first set $\delta := 1$. Then, we check if the station state $I_i^{t-\delta}$ of phase one permits that at least a certain subset of the desired bikes quantity 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 until: 1) the maximal time window length δ_{MAX} is achieved, 2) the minimal bike volume to pick up or deliver is not available in node $(i, t - \delta)$, or 3) if there already exist a pick-up or delivery SSA in node $(i, t - \delta)$ to avoid the time overlap of requests.

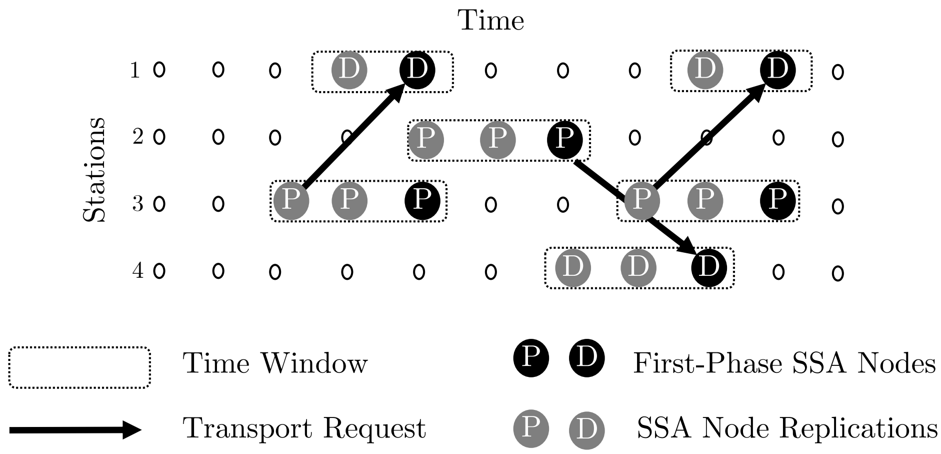

Figure 6: Illustrative solution of the matching problem.

Figure 6 shows a solution of an instance of the matching problem and the benefits of designing time windows. The y-axis represents stations, whereas the x-axis represents time periods. SSAs obtained from the first phase are depicted by black nodes (i, t) representing station i and time period t. We observe that the alternatives to pair these SSAs 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 first-phase SSAs in previous time periods. Every dashed oval wraps a certain number of nodes thus depicting what we now call a SSA request. Black links represent transport requests matching a bike quantity from a pick-up SSA request to a delivery SSA request. The matching problem is formally described in Appendix 10.

In the third phase, we rely on the algorithm for the pick up and delivery problem with time windows (PDPTW) presented in Ropke and Pisinger (2006) to assign vehicles to transport requests in order to obtain both the vehicle fleet size and the master tours. This phase considers handling time and resource limitations when determining master tours. We follow the large neighborhood search of Ropke and Pisinger (2006) to solve problem instances of PDPTW. 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.

To produce a solution of SNDBSS, we transform the obtained solution of a PDPTW into vehicle flow variables that depict master tours. Then, we just need to fix the respective vehicle flow variables and vehicle fleet sizes in the MIP problem so that a solver can quickly obtain the SNDBSS solution.

6 Simulation Approach

The SNDBSS formulation presented in this paper fully exploits a mobility demand forecast based on aggregated recorded data to determine master tours. When master tours are implemented, however, one must deal with differences between the observed mobility demand and the forecasted one. In the implementation phase, mobility demand is stochastic and revealed over time, while every user makes choices to perform journeys *independently* of the decisions made at the tactical planning level. These choices are related to the decision a user makes each time it approaches a station to start or end a ride. These user choices need to be explicitly considered for mimicking the implementation phase suitably.

We follow Datner et al. (2017) to modify the formulation $\mathcal{F}$ presented in Section 4 to a bi-level formulation in which redistribution activities are upper-level decisions, whereas the user choices are represented as lower-level decisions. Let Ω be the sample space of random events, with $\omega \in \Omega$ representing a random event. A mobility demand realization (a "day") for the random event $w \in \Omega$ is then represented by $\mathbf{d}^\omega = (U(\omega), \bigcup_{u \in U(\omega)} W_u(\omega))$. The set $U(\omega)$ defines a realization of users, while each user $u \in U(\omega)$ attempts to perform a journey of the set $W_u(\omega)$. Let $f_u(\mathbb{S})$ be a function representing the decision-making process of user u to minimize its travel time. The function $f_u(\mathbb{S})$ returns the journey performed by user u depending on the set of time-dependent station states $\mathbb{S}$ each time the user intends to rent or return a bike at a station.

The formulation $\mathcal{F}'$ that minimizes the OJDFV given mobility demand realization $\mathbf{d}^\omega$ is obtained by modifying Equations (1) and (2) of the formulation $\mathcal{F}$ such that:

$$\min \Big\{ \sum_{u \in U(\omega)} \beta_{u, f_u(\mathbb{S})} \Big\} \tag{30}$$

$$z_{u, f_u(\mathbb{S})} = 1, \quad \forall, u \in U(\omega) \tag{31}$$

Where $z_{u, f_u(\mathbb{S})}$ indicates that the user's performed journey depends on $f_u(\mathbb{S})$, while $\beta_{u, f_u(\mathbb{S})}$ is the returned JDF value of the respective journey.

Addressing the bi-level formualtion $\mathcal{F}'$ with a standard MIP solver is not suitable because, in reality, users make choices on their own and not on the basis of the redistribution plan. Therefore, we address the bi-level formulation using simulation. We simulate choices of every user with respect to walking and riding among stations with the aim of minimizing its travel time. As mentioned in Section 3, we can adapt the simulation to represent other users' sources of satisfaction like the incurred cost of using alternative transport modes. The reader is referred to Datner et al. (2017) for a formal definition of these user choices.

In the simulation, the sequence of SSAs determined by SNDBSS remain unalterable, that is, the values of vehicle flow variables y_e are given by the optimization. However, one requires guidelines to adjust bike flows with respect to the mobility demand realization $\mathbf{d}^\omega$.

To this end, we adopt the long-term redistribution strategy (LTR) introduced in Brinkmann et al. (2015) and derive target station states from the optimization. A target station state indicates the number of pursued bikes after a station visit. In the simulation, handling activities are determined so that the target states are achieved as close as possible. Achieving the target state depends on the current state, the available bikes for picking up, the available empty rack slots for delivery, the load and transport capacity of vehicles, as well as the available handling time at each station visit. Another concern to be considered when determining handling activities at a station is that rental and return requests alter its state within the visiting time of a vehicle. These user requests must be considered for each time period in order to produce accurate decisions regarding handling activities.

We denote by $[t_1, t_2]$ the time interval where vehicle v stays at station i. The variable $I_i^{t_1}$ tells the state of station i in time period $t \in [t_1, t_2]$. In this station, redistribution activity aims to achieve a target state $\tau_i^{t_2}$ in time period t_2. We set $\tau_i^{t_2} := \lfloor 0.25 * c_i \rfloor$ if a pick up is performed within $[t_1, t_2]$ in the optimization. In case a delivery is performed, we set $\tau_i^{t_2} := \lfloor 0.75 * c_i \rfloor$. As users may request bikes or empty rack slots within time interval $[t_1, t_2]$, we propose a procedure to determine handling decisions at each time period t of this interval. In the procedure, we assume that handling activity is determined once mobility demand is revealed in time period t.

We first set $t := t_1$. Then, the following equation determines the number of bikes to pick up or deliver at time period t:

$$\delta := \min\{\mu, \tau_i^{t_2} - I_i^t\} \tag{32}$$

In Equation (32), a positive δ indicates the need of delivering bikes at station i, while a negative δ suggests picking up bikes. The parameter μ indicates the maximal number of bikes that can be handled within t and $t+1$. In this equation, we assume that users have already requested bikes at time period t.

Handling the determined number of bikes δ depends on the load of vehicle v denoted by ω_v, and its capacity C. In this way, the achieved state δ^* is determined by:

$$\iota := \begin{cases} \min\{\delta, \omega_v, I_i^{t+1}\}, & \text{if } \delta^{ltr} > 0 \\ \max\{\delta, \omega_v - C, I_i^{t+1} - c_i\}, & \text{if } \delta^{ltr} < 0 \\ 0, & \text{otherwise} \end{cases} \tag{33}$$

Equation (33) indicates that the number of bikes to handle is given by the minimum between δ, ω_v, and I_i^{t+1} if a delivery of bikes is required; the maximum between δ^{ltr}, $\omega_v - C$, and $I_i^{t+1} - c_i$ if picking-up bikes is required; or zero whenever no handling operation is necessary. We assume that users already requested empty rack slots at time period $t+1$.

Then, we update $t := t + 1$ and repeat the procedure based on Equations (32) and (33) until $t = t_2$.

While the master tours always start without bike load on the vehicles, a vehicle may return to the depot with some bike load that they could not deliver within the workday length. We assume that these bikes can be placed at stations within the overnight redistribution process.

Table 1: Description of input data after filtering.

	Bay Area (BA)	Nice Ride (NR)	Hubway (HU)
City	San Francisco	Minneapolis	Boston
Number of stations	35	190	140
Min - Max - Avg. bike racks per station	15 - 27 - 19	15 - 35 - 18	11 - 46 -17
Year period	01 Mai - 31 Aug.	01 Mai - 31 Aug.	01 Mai - 31 Aug.
Avg. rides by day	1,167	2,305	4,551

7 Computational Experiments

The computational experiments aim at evaluating the effectiveness of the solution methods proposed to address SNDBSS, and at evaluating the performance of master tours by means of simulation. The methods are implemented in C++ using ILOG Concert Technology to access CPLEX 12.5 as the MIP solver. The simulation is implemented in Python 3.6. The experiments are performed on a computer with an AMD Ryzen Threadripper 1950X 3.4GHz processor and 64 GB of RAM that operate under Ubuntu Linux version 16.04.4.

7.1 Input Data

We generate problem instances using the 2015 free available data of three BSSs located in the USA: San Francisco's Bay Area (BA)[1], Minneapolis's Nice Ride (NR), and Boston's Hubway (HU). Available data include size and location of stations as well as recorded bike rides performed by users. We assume Euclidean distances between stations. We remove recorded rides with duration of less than 5 minutes as they typically result from technical deficiencies at bikes or station's racks. The recorded bike rides performed on weekends and holidays are also removed because they display different user patterns in contrast to workdays. In addition, we remove stations that have been operated less than one month. Recorded rides related to these stations are removed as well.

Table 1 provides a general description of the filtered data. BA is the smallest and NR the largest BSS in terms of number of stations. With respect to user demand, we note that HU almost doubles the average daily rides compared to NR, although both BSS have a similar number of stations. Furthermore, NR only generates around 12 rides per day, whereas all other BSS exceed 30 rides per day. The reader is referred to the respective website of each BSS for a more detailed description of the system.

We recognize that one major limitation of a recorded ride data is that it only informs stations and timestamps in which bikes are rented and returned at stations. In other words, a recorded ride data does not inform whether a user roams among stations because of failed rental and return requests. Although research efforts are necessary to derive a more trustworthy representation of mobility demand, this is beyond the scope of this paper.

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

7.2 Generating Problem Instances

For each BSS, we generate one instance for a-priori optimization and one instance of hindsight simulation. An instance of SNDBSS is defined in terms of the network infrastructure, mobility demand forecast, and all candidate redistribution and master tour decisions. Modeling all journeys a user could make to achieve its destination leads to a computationally intractable SNDBSS instances. We model journeys with which users either perform demanded rides or walk directly to their destinations. Other user choices to make journeys are represented in the simulation. An instance of the simulation is defined in terms of the network infrastructure, a set of mobility demand realizations, as well as master tours received from SNDSS. We consider a workday length of 16 hours starting at 6:00 AM. To obtain a detailed representation of time for the redistribution decisions, we split the workday length into 10-minutes time periods.

The network infrastructure includes the vehicle depot, which is assumed to be located in the center of the coverage area. The total number of bikes available in the BSS is around 50% of the total number of bike racks available. Thus, `BA`, `NR`, and `HU` disposes of 347, 1591, and 1192 bikes, respectively. We set the initial states using the guided local search introduced by Datner et al. (2017). We assume that these initial states can be achieved by overnight redistribution.

We divide the recorded ride data into two sets. The training data set is used to generate the mobility demand forecasts for SNDBSS instances. The test data set is used to generate 512 mobility demand realizations for each instance of the simulation approach.

Let (i, j, t) be a tuple defined by the station of origin i and station of destination j for user's demanded ride which start at time period t. We aggregate recorded ride data so that we obtain for each possible tuple combination a real-valued ride's demand rate r_{ijt}. To obtain a realization, these demand rates are discretized using Poisson distribution, obtaining integer demand values. The integer demand value associated with tuple (i, j, t) indicates the number of users which intend the respective demanded ride.

Recorded ride data displays the spatio-temporal dynamics of mobility demand, compare Vogel et al. (2011). As shown by O'Brien et al. (2014), `BA` and `HU` depicts a clear dominance of commuting activities, whereas `NR` is mostly used for leisure purposes. The reader is referred to O'Brien et al. (2014) for an extensive analysis of ride data in several cities.

We consider the following parameters to generate problem instances. The cost F of acquiring a vehicle is 250.00€. The cost k_e of using a vehicle movement link $e \in \mathcal{A}_{VM}$ is 2.00€ per kilometer driven. The cost k_e of using a handling link $e \in \mathcal{A}_{HA}$ is 0.50€. The handling time is one minute per bike. Each vehicle has a capacity C of 20 bikes. We assume a constant speed of $30\frac{km}{hr}$. Riding and walking speed of users are $15\frac{km}{hr}$ and $5\frac{km}{hr}$, respectively. The budget B is 1.50€ per bike in the BSS, that is, a total of 520.50€ for `BA`, 2386.50€ for `NR`, and 1788.00€ for `HU`.

7.3 Performance of the Construction Heuristic

We consider the following setting for the construction heuristic. In the first phase, the procedural approach considers a minimal time distance $\bar{t}$ of four time periods between two SSAs, that is, the the time to unload all the bikes from a full vehicle. In the second phase, we impose a maximal time windows length of three hours, that is, $\delta_{MAX} := 18$. The matching problems are solved using CPLEX with minor computational effort. In the third phase, the LNS algorithm used to address PDPTW instances stops after either 1000 iterations or 30 minutes runtime.

We consider the improvement ratio as measure for comparing solutions of SNDBSS. The improvement ratio is a metric taking values between 0 and 1 to measure the fraction of demanded bike rides satisfied. In other words, the improvement ratio is the percentage reduction of the OJDFV to the no-redistribution solution.

Table 2: Results of the construction heuristic.

BSS	No Redist.	Random Solutions		Construction Heuristic			
	OJDF	OJDF	Imp. Ratio	OJDF	Imp. Ratio	Vehicles	Runtime (s)
BA	28.84	25.62	0.11	19.02	0.34	1	1.41
NR	11.01	9.15	0.17	4.80	0.56	4	3.68
HU	63.01	55.16	0.12	40.13	0.36	3	21.88

For the sake of comparison, we generate for each BSS a pool of 50 solutions with randomly produced master tours. The vehicle fleet size v of these random solutions is an input given by the solution of the construction heuristic. Table 2 shows the results, where we report the BSS, the OJDFV (the excess travel time in hours) for a solution in which no redistribution is performed; the average OJDFV and average improvement ratio of 50 random solutions; the OJDFV, improvement ratio, and vehicle fleet size of solutions generated by the construction heuristic; and the runtime in seconds that the construction heuristic requires to reach the solution. With respect to the improvement ratio, we observe that the construction heuristic obtains solutions around three times as good as the ones obtained by the random procedure. We also note that the construction heuristic produces solutions in less than one minute for each BSS. The construction heuristic suggests the acquisition of one vehicle for BA, which makes sense as a second vehicle would consume a major part of the given budget, remaining less budget in order to operate master tours. NR and HU certainly require more vehicles than BA due to their higher dimensions in terms of stations and demanded rides.

7.4 Performance of the Matheuristic

We follow the guidelines of Section 5.1.2 to build reduced MIP problems. Regarding the size of these reduced MIP problems, we test on each BSS for $n = 50$; 100; 200; 400; and 800. Recall, that the parameter n indicates the number of vehicle detours considered in a reduced MIP problem. We set the maximal time for exploring a reduced MIP problem to 900 seconds. We set the runtime limit of the matheuristic to 3600 seconds.

Table 3: Results of the matheuristic.

BSS	n	OJDFV	Std-dev	Imp. Ratio	Iterations	% Budget Used
BA	50	11.76	0.75	0.59	2086.4	0.81
	100	11.57	1.40	0.60	1892.6	0.84
	200	10.75	0.33	0.60	1005.6	0.83
	400	9.57	0.99	0.67	116.2	0.83
	800	11.38	0.92	0.61	5.8	0.81
NR	50	0.00	0.00	1.00	246.2	0.98
	100	0.00	0.00	1.00	117.8	0.99
	200	0.00	0.00	1.00	131.4	0.99
	400	0.00	0.00	1.00	87.2	0.99
	800	0.06	0.13	0.99	28.0	0.98
HU	50	16.3	1.50	0.74	249.0	0.96
	100	13.5	2.17	0.79	233.4	0.97
	200	17.7	4.71	0.72	55.6	0.97
	400	30.4	1.55	0.52	4.0	0.89
	800	38.1	3.60	0.40	4.0	0.78

In Table 3, we report the average results of 5 runs for each BSS and value of n, where we report the OJDFV, the standard deviation (Std-dev) the improvement ratio, the average number of iterations performed until the runtime limit of 3600 seconds is reached, and percentage of budget consumed both to acquire vehicles and to operate master tours. Solutions of the construction heuristic are used as starting solutions for the matheuristic. This includes the vehicle fleet size which remains fixed within iterations. With respect to the improvement ratio, we observe that the solutions obtained by the matheuristic are around twice as good as the ones obtained by the construction heuristic. This shows that the construction heuristic is useful for quickly producing starting solutions of reasonable quality, but the matheuristic is key to obtaining high-quality solutions. As expected, the number of iterations within the time limit is inversely related to the value of n since increasing n yields larger reduced MIP problems to be explored. We also note that the sweet spot for n depends on the size of the BSS. For BA, which is the smallest BSS with respect to stations and mobility demand, we obtain the best matheuristic performance for $n = 400$. In the case of HU, $n = 100$ leads to the highest average improvement ratio.

Next, we observe that in all cases budget B is almost fully consumed and directly affects the number of vehicles to acquire. For example, for BA and $n = 400$, 48% of the budget is used for acquiring vehicles, and 35% for designing master tours. If a second vehicle would be acquired, the used budget for this item would jump to 96%, resulting in less remaining budget for designing master tours. This shows that the construction heuristic determines a reasonable vehicle fleet size for instances of SNDBSS. For NR, an average improvement ratio of 1.0 is nearly always reached. In this BSS, an improvement ratio of 1.0 could be reached with even less budget. For HU, higher improvement ratios could be achieved by increasing the budget. Below, we study this trade-off between redistribution resources and OJDFV in more detail.

Now, we study the quality of the solutions obtained by the matheuristic. To this end, we build a reduced MIP instance comprising all vehicle flow-related variables y_e ever explored within the matheuristic. All other y_e are fixed to zero. Then, we solve the MIP instance with CPLEX, using the best solution obtained by the matheuristic for a warm start. We run CPLEX for 180,000 seconds. In Table 4, we report the BSS, the best solution reached by the matheuristic as well as the solution and best bound obtained with CPLEX for the MIP instance. We exclude NR from this analysis as an optimal solution was obtained for the respective MIP problem, that is, the OJDFV equals 0.00. We observe that solutions of better quality are obtained with high computational efforts. For instance, CPLEX is only able to reach a solution of better quality after 118300 seconds for HU. We also observe that the optimality gap is very large. This is expected because solutions to the LP relaxation exhibits many fractional values close to zero. In other words, fractional vehicle flow variables can reach OJDFV closed to zero, although they are far from being close to an integer solution.

Notice that the reported MIP bound is not obtained from the full MIP problem, but from the restricted one. Unfortunately, performing the former analysis on the full MIP instance was not possible as the available memory space was insufficient to load these instances. However, as the matheuristic selects promising y_e within the search, we claim that the gap between the bounds of an original MIP instance and the this reduced MIP instance should be reasonably close.

Table 4: Quality of matheuristic solutions.

BSS	Best Math Sol.	Best red. MIP Sol.	Best red. MIP Bound	Opt. Gap
BA	7.77	6.43	1.35	78.98%
HU	10.24	9.67	0.84	91.34%

Next, we discuss the application of operator probabilities within the iterations of the matheuristic. Figure 7 shows the average probabilities over iterations for HU and $n = 100$. The x-axis represents iterations of the matheuristic. The y-axis represents the probabilities. The stacked area graphs indicate the application of each operator over iterations. 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. 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 others. However, variations of these applications over iterations indicate that the matheuristic finds solutions of superior quality even in late phases of the search process. This shows that the combination of operators is beneficial to escape from local optima.

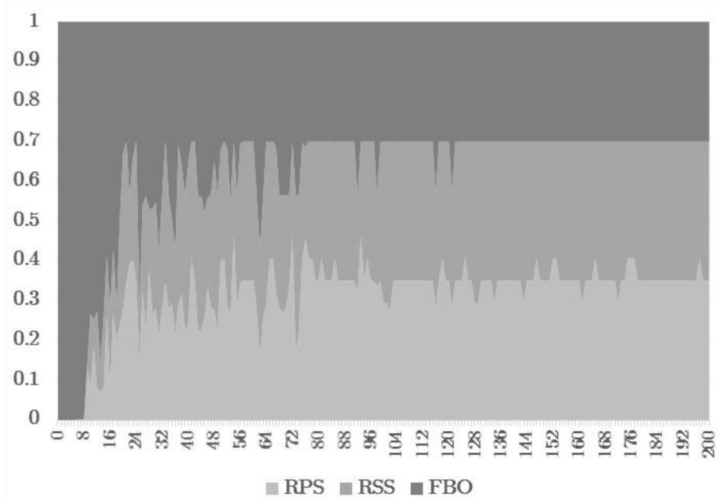

Figure 7: Probabilities of operators for HU and n equals 100.

7.5 Evaluating Master Tour Performance

In this section, we evaluate master tour performance in the optimization and the simulation. We study the effect of both varying the vehicle fleet size and increasing station capacities. To solving the SNDBSS instances, we run the matheuristic as before with the value of n yielding solutions of superior quality for the respective BSS. For each BSS and vehicle fleet size, we consider two instances: one with the original station capacities, and one in which we increase the capacity of the busiest 10% of the stations to 100. The budget constraint is excluded from this analysis as redistribution efforts are already limited by the fixed vehicle fleet size.

In Table 5, we report the BSS in the first column and the respective number of vehicles in the second column. We report the OJDF and the handled bikes by vehicle ("Redist. by Vehicle") for: the optimization with original station capacities (columns three and for), the optimization with increased station capacities (columns five and six), the simulation with original station capacities (columns seven and eight), and the simulation with increased station capacities (columns nine and ten).

First, we focus on the results for the optimization with original station capacities. As expected, we observe that adding vehicles to the fleet improves the solution quality. This is expected as the matheuristic exploits knowledge of the mobility demand forecast over the workday length. Hence, adding vehicles to the fleet can not deteriorate solution quality. However, the marginal contribution of the additional vehicle tends to decrease. In HU, for example, we clearly observe that adding a third vehicle is beneficial as the OJDF decreases around 65%. Despite the OJDFV decreases around 50% by adding the fourth vehicle, the marginal decrease just halves. For BA, an OJDFV equals zero is obtained with three vehicles. However, the major improvement of the solution quality is achieved with just one vehicle. For NR, we observe that an OJDV equals zero is achieved for all vehicle fleet sizes evaluated. With respect to redistribution efforts, we observe that for every BSS the number of handled bikes by vehicle reduces with respect to the

Table 5: Results for different vehicle fleet sizes and station capacities.

		Optimization				Simulation			
		Original Station Cap.		Increased Station Cap.		Original Station Cap.		Increased Station Cap.	
BSS	Veh.	OJDFV	Redist/Vehicle	OJDFV	Redist/Vehicle	OJDFV	Redist/Vehicle	OJDFV	Redist/Vehicle
BA	0	28.84	0.00	17.11	0.00	17.96	0.00	12.51	0.00
	1	7.77	264.00	4.13	160.00	14.13	147.91	10.95	100.77
	2	4.16	175.00	1.57	105.00	14.77	106.90	10.52	35.73
	3	0.00	130.67	0.00	80.00	12.50	82.60	10.62	36.81
NR	0	11.01	0.00	9.01	0.00	7.24	0.00	5.18	0.00
	3	0.00	79.33	0.00	53.33	7.38	39.44	4.88	8.88
	4	0.00	46.50	0.00	45.00	7.72	33.26	4.95	10.24
	5	0.00	40.80	0.00	39.60	7.27	20.38	5.12	9.33
HU	0	63.01	0.00	48.64	0.00	80.29	0.00	56.07	0.00
	2	29.32	220.00	16.00	213.00	76.19	146.65	53.88	112.17
	3	10.24	212.00	4.85	193.33	72.45	136.66	55.12	119.80
	4	5.57	181.00	3.01	161.50	67.43	106.80	50.75	76.27
	5	1.75	160.00	0.84	136.40	65.64	105.13	51.72	80.49

vehicle fleet size. In other words, adding further vehicles leads to a successively smaller utilization of them.

Next, we focus on the results for the optimization with increased station capacity. We observe that the development of the OJDF and handled bikes by vehicles with respect to the vehicle fleet size is similar to the results for the instances with original station capacity. However, we note that increasing the capacity of high-demanded stations generally improves the solution quality. In addition, less redistribution effort is required as a station with more bike racks is able to serve additional rental and return requests before redistribution becomes necessary. Despite these results seem promising to be implemented in practice, increasing station capacity is not always feasible due to land-use regulations and high implementation costs.

Now, we compare the optimization and simulation results. For a BSS and a given number of vehicles, there exists clear differences between the OJDFV obtained for the optimization and the simulation. For BA and HU, although daytime redistribution contributes to reduce OJDFV, the stochasticity of mobility demand affects the performance of master tours within the simulation. Regarding NR, we observe that in the simulation approach redistribution worsens the solution quality even with a large fleet of vehicles. The results suggest that since NR is mostly used for leisure, large variations of mobility demand over days negatively impact master tour performance. These observations are also valid for the results of the simulation with increased station capacities. This shows that considering deterministic mobility demand in tactical planning is not suitably for BSSs with high day-to-day variations in mobility demand. For these instances, a combination of overnight redistribution and intra-day control of bike redistribution should provide better results.

8 Conclusions

We introduce 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 limited resources for acquiring and operating vehicles. 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.

We propose two solution methods to address large problem instances of the MIP formulation. 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 SSAs, 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. The matheuristic relies on a neighborhood-search scheme in order to find qualitative solutions within iterations. The matheuristic builds and solves reduced MIP problems which involve a small number of vehicle flow variables of the original problem instance. We provided evidence the proposed solution methods are able to produce solutions of good quality in a reasonable runtime.

We show that the marginal benefit 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. In practice, a trade-off between redistribution costs and service level is subject to negations between the municipality and the operator. We claim that the developed approach may be used as a tool to support such negotiations.

To evaluate the master tour performance, we proposed a simulation approach in which the master tours are given by the service network design formulation, whereas handling activities are adjusted online to cope with day-to-day variations of mobility demand. The results show that our deterministic MIP model can produce master tours of reasonable performance for bike sharing systems dominated by communing rides despite the stochastic and dynamic nature of mobility demand. However, the results when users mainly perform rides for irregular demand indicate that the proposed methodology for designing master tours is not adequate. Therefore, future work will consider integrating demand stochasticity to service network design in order to reduce the gap between the optimization and simulation results. Another interesting avenue for future research is to temporarily increase the capacity of some stations in order to both reduce redistribution efforts and increase the service level. In practice, operators offer valet services, that is, additional staff who collects the overflow of bikes at stations. Service network design can contribute to determine when and how long these valet services need to be offered.

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, 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

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. 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.

J. Brinkmann, M. W. Ulmer, and D. C. Mattfeld. Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems. *Computers & Operations Research*, 2018.

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

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.

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

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.

M. DellAmico, M. Iori, S. Novellani, and A. Subramanian. The bike sharing rebalancing problem with stochastic demands. *Transportation Research Part B: Methodological*, 118:362 – 380, 2018. ISSN 0191-2615.

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.

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.

D. Freund, A. Norouzi-Fard, A. Paul, S. G. Henderson, and D. B. Shmoys. Data-driven rebalancing methods for bike-share systems. Technical report, working paper, 2016.

J. C. García-Palomares, J. Gutiérrez, and M. Latorre. Optimizing the location of stations in bike-sharing programs: A gis approach. *Applied Geography*, 35(1-2):235–246, 2012.

S. Ghosh, P. Varakantham, Y. Adulyasak, and P. Jaillet. Dynamic repositioning to reduce lost demand in bike sharing systems. *Journal of Artificial Intelligence Research*, 58: 387–430, 2017.

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.

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.

G. Laporte, F. Meunier, and R. W. Calvo. Shared mobility systems: an updated survey. *Annals of Operations Research*, 271(1):105–126, 2018.

J.-R. Lin and T.-H. Yang. Strategic design of public bicycle sharing systems with service level constraints. *Transportation research part E: logistics and transportation review*, 47(2):284–294, 2011.

J.-R. Lin, T.-H. Yang, and Y.-C. Chang. A hub location inventory model for bicycle sharing system design: Formulation and solution. *Computers & Industrial Engineering*, 65(1):77–86, 2013.

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.

C. Shui and W. Szeto. Dynamic green bike repositioning problem a hybrid rolling horizon artificial bee colony algorithm approach. *Transportation Research Part D: Transport and Environment*, 60:119 – 136, 2018. ISSN 1361-9209. Special Issue on Traffic Modeling for Low-Emission Transport.

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.

E. Zhu, T. G. Crainic, and M. Gendreau. Scheduled service network design for freight rail transportation. *Operations research*, 62(2):383–400, 2014.

9 Annex 1 - Phases of the Procedural Approach

We introduce the recursive equations to obtain SSAs during the triggering phase. Then, we describe the adjusting phase.

9.1 Triggering Phase

Let I_i^t be the bikes in racks at station i and time period t. We assume a given number of bikes I_i^0 at station i and time period 0. Demanded rides w^d are used as input. Let Ω_{it}^+ and Ω_{it}^- be the set of demanded rides w^d which respectively end and start at station i and time period t. Then, we denote f_i^{t+1} as the difference between the number of return requests in time period $t+1$ and the number of rental requests in time period t. The value of f_i^{t+1} is computed using Equation (34).

$$f_i^{t+1} = -\sum_{w^d \in \Omega_{it}^+} w^d + \sum_{w^d \in \Omega_{i,t+1}^-} w^d, \quad \forall i \in S, t < \mathcal{T} \tag{34}$$

A positive value of f_i^t means that demanded rides lead to an increase of the 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 bikes in racks because of bike rides. In time period 0, the value of f_i^0 is zero.

Let c_i the capacity of station i in terms of bike racks. If the number of bike in racks at station i is out of the interval $[0, c_i]$, we trigger an SSA to pick up or deliver a certain bike volume r_i^t in time period t. A negative value of r_i^t means that a pickup of bikes is required, whereas a positive value indicates the volume of bikes to deliver. If no SSA is necessary, the bike volume r_i^t is zero. Let θ_i^t be the pursued number of bikes in racks on station i and time period t if a SSA is performed. Here, $\theta_i^t = \lceil c_i/2 \rceil$. The bike quantity to pick up or deliver by an SSA is limited by the vehicle load capacity C.

9.2 Adjusting Phase

We describe the procedure to adjust one SSA based on the next SSA performed at the same station. The aim is to omit further SSAs at one station by varying the bike quantity of a preceding SSA.

Let r_i^t and $r_i^{t'}$ bike quantities for SSAs in time period t andt', respectively, at station i, with $t < t'$. No SSA occurs in the time interval $[t+1, t'-1]$. In addition, we define the minimum and maximum number of bikes in racks between $[t+1, t'-1]$ by I^{MIN} and I^{MAX}, respectively. We define four cases to adjust r_i^t:

1. This case involves two pickup SSAs. The bike volume to pick up $|r_i^t|$ may increase up to I^{MIN} units.

2. This is the case if both SSAs 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 SSA and then a delivery SSA. 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 SSA and then a pickup SSA 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.

10 Annex 2 - Matching Problem

We state the matching problem to pair pickup and delivery SSAs. For each SSA, we define a node (i, t) representing station i and time period t. A node involves either a pickup of bikes if $r_i^t < 0$, or a delivery of bikes if $r_i^t > 0$.

We denote the set of pick-up and delivery nodes by N_P and N_D, respectively. The available bike volume to be picked up at node $n_P \in N_P$ is denoted by a_{n_P}, whereas the bike volume to be delivered at node $n_D \in N_D$ is b_{n_D}. We refer to these nodes as pickup and delivery requests, denoted 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 denoted by B_{r_d}.

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 it is possible that the total bike volume to pick up is not equal to the bike volume to deliver, i.e., $\sum_{r_p \in R_P} A_{r_p} \neq \sum_{r_d \in R_D} B_{r_d}$, 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.

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 $C_{kl}, k \in N_P, l \in N_D$. The volume allowed 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:

$$\min \quad z = \sum_{k \in N_P} \sum_{l \in N_D} f_{kl} y_{kl} + \theta\Big(\sum_{r_P \in R_P} \alpha_{r_P} + \sum_{r_D \in R_D} \beta_{r_D} \Big) \tag{35}$$

s.t.

$$\sum_{k \in N_P(r_P)} \sum_{l \in N_D} x_{kl} + \alpha_{r_P} = A_{r_P}, \quad \forall r_P \in R_P \tag{36}$$

$$\sum_{k \in N_P} \sum_{l \in N_D(r_D)} x_{kl} + \beta_d = B_{r_D}, \quad \forall r_D \in R_D \tag{37}$$

$$\sum_{k \in N_P(r_P)} \sum_{l \in N_D} y_{kl} \leq 1, \quad \forall r_P \in R_P \tag{38}$$

$$\sum_{k \in N_P} \sum_{l \in N_D(r_D)} y_{kl} \leq 1, \quad \forall r_D \in R_D \tag{39}$$

$$x_{kl} \leq C_{kl} y_{kl}, \quad \forall k \in N_P, l \in N_D \tag{40}$$

$$x_{kl} \geq 0, \quad \forall k \in N_P, l \in N_D \tag{41}$$

$$y_{kl} \geq \{0, 1\}, \quad \forall k \in N_P, l \in N_D \tag{42}$$

$$\alpha_{r_P} \geq 0, \quad \forall r_P \in R_P \tag{43}$$

$$\beta_{r_D} \geq 0, \quad \forall r_D \in R_D \tag{44}$$

The objective function (35) 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 (37) 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 (38) 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 (39-40) enforce that only one redistribution decision can be established from or to a request. Finally, Equations (41-44) state the domain of the variables.