Inventory Routing for Bike Sharing Systems

Jan Brinkmann *, Marlin W. Ulmer, Dirk C. Mattfeld

Technische Universität Braunschweig, Decision Support Group, Mühlenpfordtstr. 23, 38106 Braunschweig, Germany

Abstract

Bike sharing systems have been launched in many cities. Such systems consist of stations allowing users to rent and return bikes. Due to spatio-temporal variation of requests, stations tend to either be empty or full of bikes. At empty stations, no bikes can be rented while at full stations bikes cannot be returned. The global goal is to satisfy all rental and return requests anytime at any station. To meet the goal, a fleet of transport vehicles relocates bikes between stations to realize suitable fill levels.

We present a multi-periodic inventory routing problem on the operational management level of bike sharing systems. Here, we take into account both time-dependent requests and target fill levels. The request structure is offered by data analysis. Target fill levels are given by optimization models on the tactical management level. The objective is to minimize the deviation of realized fill levels and target fill levels subjected to capacity and time constraints. Due to the large number of possible solutions, instances are solved by a two-dimensional decomposition approach. First, the given periods are solved independently. Second, in each period the set of stations is divided into disjoint subsets. Each subset is assigned to one vehicle. Appropriate subsets are generated by local search algorithms. Subsets are evaluated by a cost-benefit routing algorithm.

For computational studies, we use real-world data of Vienna’s bike sharing system "CityBike Wien". Our results depict that appropriate subsets allow efficient routing and relocation of bikes and thus, target fill levels can be realized.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Keywords: inventory routing; local search; shared mobility; bike sharing

* Corresponding author. Tel.: +49 531 391-3112; fax: +49 531 391-8144.

E-mail address: j.brinkmann@tu-braunschweig.de
1. Introduction

Cities deal with a large volume of traffic resulting in traffic jams and environmental pollution, e.g., noise and carbon dioxide emission. One approach to tackle these inconveniences is the use of public bike sharing systems (BSS, Büttnner et al., 2011). Most BSS consist of stations, where users can rent and return bikes ad hoc. Further, station-less systems exist, where bikes are parked anywhere in the city center while they are not rented (Shaheen et al., 2010). Regarding rental requests, station-based systems seem to be advantageous since users can always rent bikes at the same stations according to their habits. In station-less systems, every time a user wants to rent a bike, the nearest bike has to be located, e.g., via smart phone app. Regarding return requests, station-less systems seem to be advantageous since users can return bikes almost anywhere. In station-based systems, bikes have to be returned at stations.

To ensure that users can rent and return bikes anytime at any place, providers aim to provide sufficient numbers of bikes and free bike racks at all stations throughout the day in a cost efficient manner. Here, providers face several challenges. Due to the short trip duration, requests are uncertain. Further, bikes are not returned at the station they have been rented at and so stations may tend to either run full or out of bikes. Moreover, the structure of user request is subject to spatio-temporal variation. Thus, the request behavior for every station differs in the course of the day. To satisfy as many requests as possible, the BSS has to be designed and maintained to match these challenges.

For station-based BSS, the challenges can be proceed on the strategic (long-term), tactical (mid-term), and operational (short-term) management levels (Vogel, 2016). The strategic management decides about locations and capacities of stations. On the tactical level, optimal fill levels for each station and for (hourly) periods throughout the day are determined by analyzing recorded data on single trips. Each trip consists of a rental and a return request. For each request, the associated station and the time it takes place are given. The determined fill levels anticipate typical user behavior and serve as input for operational planning. To realize these target fill levels, we distinguish two options. An indirect option focuses on user-based relocation of bikes initiated by financial incentives. Here, users are granted a price reduction if they rent or return a bike in a defined area or at a defined station respectively (Ruch et al., 2014). A direct option draws on vehicles which are routed between stations to pick up and deliver bikes (Shaheen et al., 2010). We state the direct option suits station-based BSS well since stations allow relocating bulks of bikes with only one stop. In a station-less system, vehicles would have to stop for every bike to pick up. Thus, we assume vehicles in a station-based system to allow most efficient relocations to realize target fill levels. Nevertheless, 30% of the running costs (between 1,500 and 2,500€ per bike and year) are caused by relocations (Büttnner et al., 2011). Thus, providers aim for efficient relocations. If not stated otherwise, from here on we focus on station-based BSS and direct relocations.

The target fill levels are real-valued and, therefore, not suitable for operation planning. Hence, fill levels are enlarged to fill intervals with integer bounds. Generally, not every target interval can be realized due to capacity and time constraints. The resulting deviations of realized fill levels and given target intervals are called gaps. We assume a high probability of unsatisfied rental or return requests in the presence of large gaps. Thus, the objective is to minimize the squared gaps over all stations. Then, large gaps are mostly avoided while small gaps are allowed.

In this article, we model the problem on realizing target intervals as a multi-periodic inventory routing problem (IRP). Within each period, decisions considering the served stations, the amount of transported bikes, and the according vehicle routing have to be made. For this problem, decisions impact the current and following periods. Due to the interval representation of fill levels, the model allows stations to be both source and sink, and even balanced stations to be used to reduce gaps of adjacent stations. Since this rich IRP cannot be solved to optimality within reasonable time, we apply a temporal and spatial decomposition approach. Temporal decomposition is applied using a rolling horizon, i.e., time periods are solved independently. Within each period, a spatial decomposition is applied. To this end, the set of stations is partitioned into disjoint subsets. Efficient subsets consider both short distances between stations and a balance in bike surpluses and shortages. The subsets are assigned to vehicles and evaluated by a cost-benefit routing algorithm. We test the approach with real-world data from Vienna's BSS "CityBike Wien" operating on 59 stations. The required target fill levels are provided by Vogel et al. (2014). We examine the results regarding the search operators, the local search algorithms, and the number of vehicles. Our results depict, that our approach offers suitable subsets where the target intervals can mostly be realized. Only in the rush hour a few stations remain imbalanced.
This article is structured as follows: In Section 2, we give a literature review of BSS related literature highlighting the work considering IPR. In Section 3, the multi-periodic rebalancing problem in BSS is defined. The decomposition approach is introduced in Section 4. Computational studies and their results are presented in Section 5. Finally, a conclusion is drawn in Section 6.

2. Literature Review

Generally, the goal is to install and maintain the functionality of a BSS in a cost efficient way (Benchimol et al., 2011). This challenge can be handled on several management levels (Vogel, 2016). We first give a brief overview of literature on the strategic and tactical management level. Then, we focus on the operational level where we distinguish single-periodic and multi-periodic rebalancing.

On the strategic level, decisions about network design are made. Decisions contain the locations and capacities of stations, and adaptations of the according infrastructure as well. Lin and Yang (2011) propose hub models to determine locations and sizes of stations, and to identify the need for exclusive bike lanes. García-Palomares et al. (2012) implement location allocation models to decide about station locations.

The tactical management addresses both data analysis and optimization. Request structure analysis has been done by Borgnat et al. (2011) and Vogel et al. (2011). Additionally, Vogel et al. (2011) state that stations can be clustered according to the request structure. E.g., stations in residential areas are indicated by a relatively high number of rental requests and a relatively low number of return requests in the morning. Further, stations in working areas are indicated by a relatively low number of rentals and a relatively high number of returns in the morning. In the course of the day, the request structure inverts. This behavior exists since BSS are used by commuters. Vogel et al. (2014) present a resource allocation problem (RAP) to determine suitable fill levels per station and hour. Here, decisions about relocations are made using a transportation model. Thus, routing is simplified by single transportation flows. The RAP's objective is to minimize costs for tours, relocations, and unserved requests. It is solved by a hybrid metaheuristic. Initial fill levels, expected requests, and claimed relocation transports lead to suitable target fill levels for upcoming time periods. These fill levels serve as anticipating input for operational planning. We state that request can be served if target fill levels are realized.

The operational management faces vehicle routing problems with inventory decisions. These problems are well known as IRPs (Dror et al., 1985). A comprehensive literature overview on IRPs is presented by Coelho et al. (2014). Since in BSS two competing resources have to be balanced, namely bikes and free bike racks, IRPs for BSS have to be studied separately. IRPs for BSS differ in objective function and constraints. Here, we further differentiate between single-periodic and multi-periodic rebalancing.

2.1. Single-Periodic Rebalancing

In single-periodic rebalancing problems, bikes are relocated over night when the system is closed. Requests and time-dependent fill levels are not considered. Therefore, the target fill levels represent suitable initial fill levels for the next day. Usually, closed tours are generated, i.e., tours start and end at a depot. Chemla et al. (2013) and Erdoğan et al. (2015) present problem settings in which a single vehicle has to realize target fill levels. Instead of target fill levels, Erdoğan et al. (2014) aim on realizing target intervals. The objectives are to minimize the travel distance or time respectively. Raviv et al. (2013) consider a fleet of vehicles and a limited time horizon. A service time, i.e., the time a vehicle spends at a station to perform relocation operations, depends on the exact number of relocated bikes. Due to the time limit, the realization of all target fill levels may not be possible. The deviation of realized fill levels and target fill levels is called gap. The objective is to minimize the overall gap, the total travel time, and service times. Two models using arc-index and time-index formulation are defined. Similar problem settings are introduced by Di Gaspero et al. (2013), Rainer-Harbach et al. (2013), and Papazek et al. (2014). The objectives are the minimization of gaps, the total time for traveling, and the number of relocation operations. The authors consider static service times which represent the expected time to perform relocation operations. Di Gaspero et al. (2013) implement constraint programming and ant colony optimization. Rainer-Harbach et al. (2013) apply variable neighborhood search (VNS) while Papazek et al. (2014) use a hybrid greedy randomized adaptive search approach (GRASP) combined with path relinking.
Kloimüllner et al. (2015) introduce a problem where the stations are marked for pick-ups or deliveries beforehand. The objective is to maximize the number of visited stations within limited time. Service times are static and pick-up and delivery stations have to be visited alternately. A cluster-first route-second heuristic is implemented to assign stations to vehicles.

Schuijbroek et al. (2013) define a single-periodic mixed integer program allowing open tours. An open tour does not necessarily have identical start and end points. In a preprocessing stage, target intervals for every station are determined via Markov chains. Then, in the inventory routing stage, the target intervals are realized without explicit consideration of user activities. The objective is to minimize the maximum tour length over all vehicles. Optimization is done via a cluster-first route-second heuristic and constraint programming.

2.2. Multi-Periodic Rebalancing

Multi-periodic rebalancing problems consider a time-dependent request structure and/or time-dependent target fill levels. The models are suitable if relocations should be performed during the day.

Contardo et al. (2012) and Kloimüllner et al. (2014) introduce multi-periodic problems with time-dependent requests for each station. The objective is to minimize the number of unsatisfied requests. Contardo et al. (2012) apply arc-flow formulation for modeling. Optimization is carried out with Dantzig-Wolfe and Bender's decomposition approaches. Kloimüllner et al. (2014) optimize via GRASP in combination with VNS.

The referred literature and characteristics are summarized in Table 1. To the best of our knowledge, a problem comprising both time-dependent requests as well as target fill levels has not been studied yet. Since the request structure in BSS is subject to spatio-temporal variation, we state that time-dependent request should be considered. Further, time-dependent targets allow implicit anticipation since they point out future requests. Hence, relocations can be applied before certain requests occur causing the BSS to become imbalanced. To provide integrated decision support, we present a problem setting considering these two aspects in the next section.

Table 1. Inventory Routing Problems for Bike Sharing Systems.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>fleet of vehicles</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>limited time</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>service time</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>target intervals</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>time-dependent targets</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>time-dependent requests</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>max. number of visited stations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>min. tour length / time / costs</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>min. make span</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>min. gaps</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>min. relocation operations</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>min. unsatisfied requests</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
3. A Multi-Periodic Inventory Routing Problem for Bike Sharing Systems

In this section, we formulate an IRP on rebalancing a station-based BSS over a whole working day with respect to both time-dependent target intervals and time-dependent requests. Relocation operations are carried out by a fleet of vehicles within limited time.

We consider a BSS consisting of a set \( N = \{n_0, n_1, \ldots, n_{\text{max}}\} \) representing depot \( n_0 \) and the stations. Stations have a capacity given by \( r: N \rightarrow \mathbb{N}_0 \) and a fill level indicated by \( f: N \rightarrow \mathbb{N}_0 \). A fleet of vehicles \( V = \{v_1, \ldots, v_{\text{max}}\} \) moves between stations to perform relocations. Each vehicle's capacity is denoted by \( q: V \rightarrow \mathbb{N} \). The numbers of bikes currently loaded by the vehicles are pointed out by \( l: V \rightarrow \mathbb{N}_0 \). Travel times are given by \( d: N \times N \rightarrow \mathbb{R}^+ \). The service time for loading or unloading one bike is indicated by \( \lambda \in \mathbb{R}^+ \). Optimization is applied over time periods in \( P = \{p_0, \ldots, p_{\text{max}}\} \). Each period \( p \) has a time horizon \( T_p = \{t_{p,0}, \ldots, t_{p,\text{max}}\} \). Target intervals are given by \( \tau: N \times P \rightarrow \mathbb{N}_0 \) and \( \overline{\tau}: N \times P \rightarrow \mathbb{N}_0 \), where \( \tau(n,p) \leq \overline{\tau}(n,p) \) holds for given \( n \) and \( p \). A station \( n \) is called balanced if and only if the fill level is in the target interval, i.e., \( \tau(n,p) \leq (n) \leq \overline{\tau}(n,p) \). We call the BSS balanced if each station \( n \in N \) is balanced. The expected rental and return requests are consolidated in \( \varepsilon: N \times P \rightarrow \mathbb{Z} \). For a given station \( n \) and a given period \( p \), the number of rentals prevails the number of returns if \( \varepsilon(n,p) < 0 \). Analogously, returns prevail rentals if \( \varepsilon(n,p) > 0 \). We want to emphasize that a period is characterized by static target intervals and expected requests for each station. In the first period, all stations are balanced and vehicles start empty at the depot. In the last period, they return at the depot empty as well.

In the first period \( p_0 \), all stations are balanced. Then, in the beginning of each period \( p \), the expected requests are added to the stations’ fill levels. We assume to satisfy all requests at a given station in a given period if the station is balanced in the beginning of the period. To estimate the probability of unserved requests, we observe the deviation of a station’s fill level and its target interval. This deviation is called gap. The gap \( g: N \times P \rightarrow \mathbb{N}_0 \) is defined in Formula (1).

\[
g(n,p) := \begin{cases} 
\tau(n,p) - (n) & : \tau(n,p) > (n) \\
(n) - \overline{\tau}(n,p) & : \overline{\tau}(n,p) < (n) \\
0 & : \text{else}
\end{cases}
\]  

(1)

We assume a high probability of unserved requests in the presence of large gaps. Due to capacity and time constraints, usually not all stations can be served. To mostly avoid large gaps, the objective (2) is to minimize the squared gaps over all stations and periods.

\[
\min \sum_{i=1}^{N} \sum_{j=0}^{P} g(n_i,p_j)^2
\]  

(2)

4. A Two-Dimensional Decomposition Approach

Due to the large number of possible tours and relocation operations, problem instances cannot be solved exactly within reasonable time. Thus, we propose a two-dimensional decomposition approach where the IRP is divided into subproblems. In the temporal dimension, the time periods are solved independently. In the spatial dimension, we partition the set of stations into subsets and assign each subset to one vehicle. The subproblems are solved independently.

4.1. Temporal Decomposition

To apply the temporal decomposition, the periods are optimized independently. This concept is well known as rolling horizon (Baker, 1977). For a given period, the target fill levels are static. According to the problem definition in Section 3, all stations are balanced in the first period. In every period, rental and return requests as well as pick-up and deliveries influence the stations’ fill levels. In a succeeding period, the fill levels are equal to the realized fill levels of the preceding period. At the end of a period, the sum of gaps over all stations denotes the period’s objective value. Hence, the sum of objective values over all periods depicts the objective value as defined in Formula (2).
4.2. Spatial Decomposition

In each period, we use spatial decomposition to reduce the number of routing decisions. Therefore, the set of stations is partitioned into subsets (Section 4.2.1). Stations within efficient subsets consider short distances between each other. Further, the shortages and surpluses of bikes need to be in balance. Thus, the single stations can be balanced. A generalization of this problem is given by Balas and Padberg (1976). A set partitioning problem's solution is a partition defining disjoint subset. In other words, it assigns each station to one vehicle. For each vehicle, a routing problem arises. To solve the routing problem, we define a cost-benefit routing algorithm (Section 4.2.2).

4.2.1. Local Search

The set partitioning problem is tackled by a local search approach. Starting with a given solution, local search iteratively searches the current solution's neighborhood to find better ones. A solution's neighborhood is defined by an operator and includes similar solutions. In each iteration, a search algorithm chooses a new solution from the neighborhood. The procedure is repeated until a termination criterion is met. (Rothlauf, 2011)

We consider two operators. The neighborhood defined by Insert contains all solutions differing in exactly one assignment. To generate the neighborhood, one station is removed from its subset and is assigned to another. Exchange refers to two stations of different subsets. To generate the neighborhood, the first station is assigned to the second station's subset and vice versa. Thus, the neighborhood generated by Insert is smaller than Exchange's neighborhood. However, Insert may change the subsets' sizes. To evaluate partition's subsets, a cost-benefit routing algorithm is applied (Section 4.2.2). It offers an objective value for each subset. Consequentially, a partition's objective value (and therefore the associated period's objective value as well) is the sum of objective values over all subsets. A subset's objective value is the sum of squared gaps over all stations within the subset.

Starting with randomly generated solutions, two search algorithms are applied. Hill Climbing (HC) iteratively chooses the best solution from the current solution's neighborhood until a local optimum is found (Mattfeld, 1996). A local optimum is the best solution in its neighborhood. For further exploitation of the search space, we consider Simulated Annealing (SA). The name comes from metallurgy, where a convergence process is controlled by the temperature. In each iteration, SA randomly chooses a new solution from the current solution's neighborhood. Depending on the objective values and a temperature parameter T, the chosen solution is accepted. The temperature T explicitly controls exploitation and intensification. Assume 0_c to be the current solution's objective value, and 0_n to be the new solution's objective value. The probability \( \phi \) to accept the new solution is defined according to Formula (3).

\[
\phi := \min \left\{ 1, \exp \frac{0_c - 0_n}{T} \right\}
\]  

A superior or equally evaluated solution is accepted always since \( \phi = 1 \) holds if \( 0_c \geq 0_n \). To escape from local optima, inferior solutions can be accepted with a probability \( \phi \in (0,1) \). Given \( 0_c < 0_n \), a high temperature provides a high probability to accept the new inferior solution. A low temperature leads to a low probability of accepting inferior solution. If the temperature is too high, SA cannot perform better than a random walk within the search space since no intensification is possible. If the temperature is too low, SA cannot perform better than HC since no exploitation is possible. At the end of each iteration, regardless of whether the new solution has been accepted or not, the temperature is lowered. The procedure stops if it meets a termination criterion. After the last iteration, the best solution found is returned. (Rothlauf, 2011)

4.2.2. Cost-benefit Routing

For evaluation of the partition's subsets, we define an inventory routing algorithm comparing costs and benefits of visiting a station. The goal is to generate a tour including relocation operations for each subset. Routes are generated here by an adapted Nearest Neighbor routing algorithm. In Nearest Neighbor usually the next vehicle's destination is the nearest station that has not been visited yet (Rosenkrantz, 1977). In our approach, we also take possible relocations into account. Due to the limited time horizon, we cannot visit every imbalanced station. So, we have to balance out costs (time) and benefits (as a reduction of gaps). To determine the next station to visit, we
determine a score $\rho$ for each station within the given subset. Therefore, in Formula (4), we determine the maximum reduction $\Delta g: N \times V \times P \to N_0$ of the given gap for each station first.

$$\Delta g(n, v, p) := \begin{cases} \min \{g(n, p), l(v)\} : \tau(n, p) > f(n) \\ \min \{g(n, p), q(v) - l(v)\} : \tau(n, p) < f(n) \\ 0 : \text{else} \end{cases}$$

(4)

Given that the associated stations $n$ has a lack of bikes, $\Delta g$ is the minimum of the gap and the number of bikes $l(v)$ currently loaded by vehicle $v$. Given that the station has a surplus of bikes, $\Delta g$ is the minimum of the gap and the number of bikes that can be loaded additionally by the vehicle. Given the vehicle currently stays at station $m$, the time for conducting the reduction depends on the travel time $d(m, n)$, on the service time $\lambda$, and on the absolute number of bikes to relocate as well. Here, the number of bikes to relocate is equal to $\Delta g$. Thus, the algorithm relocates the minimum number of bikes to balance the station. Now, for each station $n$ within the vehicle's subset a score $\rho: N \times V \times P \to R_0$ is determined as shown in Formula (5).

$$\rho(n, v, p) := \frac{\Delta g(n, v, p)}{d(m, n) + \lambda}$$

(5)

Hence, $\rho$ balances reduction of gaps (benefits) and spent time (costs). The station with the highest score will be served next. Destinations and relocation operations are iteratively generated as long as the time limit of the period is not exceeded.

5. Case Studies

In this section, we define the test instances based on the case study on "CityBike Wien" and tune the applied algorithm. Then, we analyze the results regarding solution quality and show the decomposition approach's impact on objective values, relocations, and fill levels.

5.1. Instances

For computational studies, we use real-world data from Vienna's BSS "CityBike Wien" (Gewista Werbegesellschaft m.b.H., 2014). The data set contains approximately 750,000 single trips observed in the years 2008 and 2009. At that time, the BSS consists of 59 stations and 627 bikes. Station capacities differ between 10 and 40 bike racks. The travel time between a pair of stations depends on the Euclidean distance and on the vehicle's speed. We assume to have a constant speed of 15km/h. The service time required for the relocation of one bike is 2 minutes. We examine the results for 2, 3, 4, and 8 homogeneous vehicles with capacity of 10 bikes.

![Fig. 1. Number of Trips per Period for a Typical Working Day.](image-url)
Expected trips (and therefore requests as well) for a working day have been extracted by Vogel et al. (2016). These set of trips contains 1,569 single trips per day. The temporal distribution of trips over the day is shown in Figure 1. Considerable activities start in period 8 and indicating a small morning peak. Between periods 10 and 18, the number of trips increases steadily. The global peak takes place in period 18. A significant number of customers uses the BSS regularly to commute between home and work. In the morning, stations in working areas congest, while stations in residential areas run out of bikes. This behavior inverts in the course of the day. A detailed analysis of the spatio-temporal request structure can be found in Vogel et al. (2011). Using this data set, target intervals for each station and each hour of the day have been generated by Vogel et al. (2014). This preprocessed data serves as input for our computational studies. For two exemplary stations, Figures 3 and 4 (Section 5.3) depict the development of target intervals during the day reflecting commuter requests. In the temporal decomposition, we apply 24 periods since target intervals have been generated for each hour of the day.

5.2. Algorithm Tuning

To apply local search, an initial subset is randomly generated. For SA, parameter settings are chosen according to Rothlauf (2011). So, we define a cool down factor \( c \in [0.99, 0.999] \) to lower temperature \( T \). The initial temperature \( T_0 \) is initialized based on the objective value's standard deviation \( \sigma \) of a set of randomly generated solution. Rothlauf (2011) recommends initial temperatures \( T_0 \in [\alpha, \beta \sigma] \). Therefore, we start the first iteration with \( T_0 \in [384, 768] \). At the end of each iteration \( i \), the temperature of the next iteration \( i + 1 \) is defined by \( T_{i+1} := c \cdot T_i \). The procedure continuous until \( T < 0.2 \) (752 – 8,249 iterations). From then on the probability \( \phi \) of accepting a minimal inferior solution falls below 1% since \( \exp(-1/T) < 0.01 \) for \( T < 0.2 \). We define this termination criterion, since a chance less than 1% to accept inferior solutions is negligible and further exploitation is hardly possible.

5.3. Results

First, we depict the advantages of local search with different search operators. Second, we compare the different local search algorithms regarding solution quality. For SA, we exemplarily analyze the solution's structure in detail. Experiments base on a Java implementation and have been performed on an Intel Core i5-3470 at 3.2 GHz with 32GB RAM.

To examine the solution quality of the different operator settings, we simulate 500 non-concatenated working days. Here, we apply HC with single operators and the combination of operators as well as different numbers of vehicles. To combine Insert and Exchange, first a small neighborhood (Insert) is examined. When the termination criterion is met, the neighborhood is enlarged (Exchange). The average objective values are shown in Table 2.

<table>
<thead>
<tr>
<th>Operator \ Number of Vehicles</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>no optimization via local search</td>
<td>842.07</td>
<td>754.40</td>
<td>779.96</td>
<td>1,088.18</td>
</tr>
<tr>
<td>Insert</td>
<td>242.10</td>
<td>97.86</td>
<td>71.66</td>
<td>60.34</td>
</tr>
<tr>
<td>Exchange</td>
<td>248.79</td>
<td>113.87</td>
<td>96.61</td>
<td>106.22</td>
</tr>
<tr>
<td>Insert / Exchange</td>
<td>211.45</td>
<td>86.09</td>
<td>65.24</td>
<td>57.74</td>
</tr>
</tbody>
</table>

The first row refers to randomly generated subsets without optimization. As we can see, the decomposition approach leads to huge improvements. Further, the operator combination outperforms the single operators by far for every number of vehicles. In the following, we use the combination of Insert and Exchange to compare HC and SA. Again, the simulation comprises 500 non-concatenated working days for each combination of algorithms and numbers of vehicles. The results and the according runtimes are shown in Table 3. The more vehicles are considered, both algorithms are able to lower the sums of squared gaps, gaps, and imbalanced stations as well. As expected, SA outperforms HC regarding each of these observations. Additionally, SA provides lower standard deviations of squared gaps. The numbers of served stations and relocation operations increase only in a small range if more vehicles are applied. Therefore, we assume these subsets of stations to allow more efficient relocations and further to
realize the target intervals. However, SA with four vehicles leads to superior solutions than HC with eight vehicles. I.e., for the given instances, 50% of the costs for rebalancing can be saved by using SA. However, SA needs a multiple of the runtime spent by HC. The average runtimes for simulating one working day decrease if the number of vehicles is raised. This points out, that suitable subsets can be found fast if more resources are provided.

Table 3. Results by Hill Climbing and Simulated Annealing

<table>
<thead>
<tr>
<th>Algorithm \ Number of Vehicles</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Squared Gaps</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hill Climbing</td>
<td>211.45</td>
<td>86.09</td>
<td>65.24</td>
<td>57.74</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>171.99</td>
<td>69.98</td>
<td>52.77</td>
<td>49.83</td>
</tr>
<tr>
<td><strong>Standard Deviation of Squared Gaps</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hill Climbing</td>
<td>47.72</td>
<td>21.04</td>
<td>17.27</td>
<td>14.40</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>30.84</td>
<td>7.32</td>
<td>11.65</td>
<td>11.71</td>
</tr>
<tr>
<td><strong>Gaps</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hill Climbing</td>
<td>76.76</td>
<td>42.53</td>
<td>31.19</td>
<td>25.48</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>67.32</td>
<td>39.32</td>
<td>25.83</td>
<td>22.75</td>
</tr>
<tr>
<td><strong>Imbalanced Stations</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hill Climbing</td>
<td>42.58</td>
<td>29.25</td>
<td>21.20</td>
<td>15.69</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>38.56</td>
<td>28.88</td>
<td>17.38</td>
<td>14.03</td>
</tr>
<tr>
<td><strong>Relocation Operations</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hill Climbing</td>
<td>178.92</td>
<td>179.96</td>
<td>181.51</td>
<td>182.13</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>179.13</td>
<td>179.93</td>
<td>182.53</td>
<td>182.40</td>
</tr>
<tr>
<td><strong>Served Stations</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hill Climbing</td>
<td>71.55</td>
<td>72.73</td>
<td>74.30</td>
<td>75.65</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>71.87</td>
<td>73.06</td>
<td>75.31</td>
<td>75.70</td>
</tr>
<tr>
<td><strong>Runtime (in Seconds)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hill Climbing</td>
<td>5.18</td>
<td>1.38</td>
<td>1.35</td>
<td>1.82</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>158.81</td>
<td>44.26</td>
<td>25.36</td>
<td>11.66</td>
</tr>
</tbody>
</table>

To further analyze the solution structure, we exemplarily study the results of SA and four vehicles. Figure 2 depicts for each period the sums of squared gaps, imbalanced stations, and relocation operations. Between periods 0 and 7, only a small number of trips take place and neither interval violations (indicated by squared gaps) nor relocation operations occur. The first very small interval violations can be experienced in period 8 at scattered stations. Up to and including period 14, the relocation operations are able to ensure a nearly perfectly balanced system. During the rush hours between periods 15 and 18, we observe some imbalanced stations and the associated squared gaps. Thus, we face a significant probability of failed requests in that time. In the evening, the relocation operations are able to balance the system.

We exemplary study the occurrences at two exemplary stations. The target intervals and realized fill levels per period for a residential and a working station are shown by Figures 3 and 4. We want to emphasize that the target intervals reflect the commuter request behavior. In the morning at the residential station (Figure 3), the target intervals demand high fill levels since a high number of rentals and no returns are expected. In the course of the day, the number of rentals decreases and also return requests occur. Thus, the target intervals demand lower fill levels. For the working station (Figure 4), the characteristics are vice versa. In the morning, the target intervals are low due to a high number of returns take place and no bikes are rented. Later the day, also rentals occur. So, the target intervals are high. Regarding the realized fill level, we observe some self-balancing effect. I.e., the realized fill levels will match the target intervals due to rentals and returns even without relocations. In period 16 the target interval drops at the residential station while the working station's target interval jumps to the top. This information depicts a
need of relocations. Therefore, a vehicle picks up bikes at the residential station and delivers them to the working station. At the residential station, the realized fill level meets the target interval again while a small gap appears at the working station. Since rentals prevail return requests, the gap is closed in the evening without further relocation operations. In period 23, the stations' fill levels meet both the period's target intervals as well as the next day's first period's target intervals.

Fig. 2. Results per Period by Simulated Annealing and four Vehicles.

Fig. 3. Target Intervals and Realized Fill Levels for a Residential Station.

Fig. 4. Target Intervals and Realized Fill Levels for a Working Station.
6. Conclusion

We have introduced a multi-periodic inventory routing problem for rebalancing station-based bike sharing systems. In bike sharing systems, providers have to satisfy as many rental and return requests as possible. Therefore, vehicles are applied to pick up and deliver bikes. Time-dependent target intervals for stations' fill levels anticipate future requests. We assume to fulfill all requests at a given if the fill level is within the target interval. Therefore, the objective is to route a fleet of vehicles to minimize the deviation of fill levels and target intervals. Thus, inventory and routing decisions have to be made. To this end, a two-dimensional decomposition approach is presented. In the temporal dimension, time periods are solved independently. In the spatial dimension, the set of stations is partitioned into subsets. Each subset is assigned to one vehicle. The set partitioning problem is tackled via local search by Hill Climbing and Simulated Annealing. To evaluate subsets, a cost-benefit routing algorithm is introduced.

To evaluate the decomposition approach, we have presented case studies based on real-world data of Vienna's bike sharing system "CityBike Wien". Results depict that the selection of subsets has a significant impact on inventory routing. For this problem, local search with different search operators is advantageous. Compared to a greedy Hill Climbing procedure, Simulated Annealing leads to superior solutions regarding objective value and standard deviation. Further, even with fewer vehicles Simulated Annealing outperforms Hill Climbing. In detail, routing for subsets generated by Simulated Annealing allows the realization of target intervals expect for the rush hour in the late afternoon. During this time, only a few stations remain imbalanced.

Future work might comprise several features. To generalize our findings, the decomposition approach can be evaluated on larger instances. To constitute a more realistic simulation, the request structure can be modeled as a stochastic function. Further, trips can be modeled as a tuple of one rental and one return request. This would allow observing a number of key performance indicators, e.g., failed requests and user detours. For a vehicle routing problem with stochastic user requests, Ulmer et al. (2015) make use of stochastic information allowing explicit anticipation. We claim that this might also be beneficial for stochastic inventory routing and dynamic solution approaches in bike sharing systems.

Acknowledgements

The authors thank Patrick Vogel for both the provision of the target intervals as well as his helpful suggestions and comments. We also thank Gewista Werbegesellschaft mbH for providing the required data from their project "CityBike Wien".

References


