Inventory Routing for Bike Sharing Systems

Jan Brinkmann · Marlin W. Ulmer · Dirk C. Mattfeld

January 12th, 2015

Abstract

In the last decade, bike sharing systems have been launched in many cities. Such systems contain of stations allowing users to rent and return bikes. Due to spatio-temporal variation of demands, stations tend to run out or full of bikes. Customer demands may be not fulfilled. To rebalance the system, a fleet of vehicles transport bikes between stations during the day. Objective is to maintain suitable fill-levels for stations. These fill-levels are derived from target intervals provided by tactical management.

We present an inventory routing problem on operational management level of bike sharing systems taking into account both time-depended target intervals and user activities. We define the multi-periodic problem setting as an integer program. Due to the large number of variables, instances are solved by a decomposition approach. Optimization is conducted over a rolling horizon. In each period, the set of stations is divided into partitions. For routing, each vehicle is assigned to one subset of stations. Appropriate partitions are achieved via variable neighborhood search by local search algorithms. Partitions are evaluated by a cost-benefit inventory routing algorithm. For computational studies, we use real world data of Vienna's bike sharing system "CityBike Wien". Our results depict that appropriate partitions allow efficient routing and repositioning using two vehicles achieving an overall service level of 91.36%.

Keywords inventory routing, vehicle routing, metaheuristics, variable neighborhood search, shared mobility, bike sharing

Contact: Decision Support Group, Technische Universität Braunschweig, Mühlenpfordtstr. 23, 38106 Braunschweig, Germany, j.brinkmann@tu-braunschweig.de.

This working paper is currently under review.

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 drawbacks is the use of public bike sharing systems (BSS, Büttner et al. 2011). In BSS, users ad hoc rent and return bikes at certain stations. Usually, trips are not charged, if they not exceed a certain length of time. The revenue generated by the trips cannot compensate operational costs. Therefore, BSS are subsidized on condition of ensuring a certain service level. So, objective is the cost efficient provision of sufficient numbers of bikes and bike racks at all stations throughout the day. Providers face several challenges maintaining this service level. Due to the short time rental activities, customer demands are uncertain. Further, bikes are not returned at the station they have been rented. Some stations may tend to run out of bikes or congest. Moreover, mobility demand is subjected to spatio-temporal variation, so the request behavior for every station differs during the day. To ensure that customers are able to use the service anytime, the BSS has to be designed and maintained to match this challenges.

This can be achieved on strategical (long-term), tactical (mid-term) and operational (short-term) management levels (Vogel et al., 2015). The strategic management decides about location and size of stations. On tactical level, optimal fill-levels for each station and for (hourly) periods throughout the day are determined by analyzing user activities. These fill-levels anticipate typical user behavior and serve as input for operational planning. In our article, we analyze the problem setting on the operational management level. Here, the actual fill-levels have to be adjusted to the target fill-levels, provided by the tactical management. Therefore, capacitated vehicles are routed between stations to reallocate bikes during the day. The provided target fill-levels are real-valued and, therefore, not suitable on the operation level. Hence, fill-levels are assigned to fill-intervals. Generally, not every target interval can be achieved due to capacity and time restrictions. So, routing aims on reducing the number of imbalances over all stations. These imbalances are represented by the the deviation of actual fill-levels and target intervals. The higher the deviation, the more grows the risk of an unfulfilled demand due to stochastic customer behavior. Hence, objective is to minimize the squared deviations over all stations.

In this article, we model this problem as a multi-vehicle multi-periodic inventory routing problem (IRP). Within each period, decisions consider 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. A small interval violation in one period might allow skipping the station in the following. Due to the interval representation of fill-levels, the model allows stations to be source and sink and even balanced stations to be used to reduce imbalances of adjacent stations. Since this rich IRP cannot be optimally solved within reasonable time, we apply a temporal and spatial decomposition approach. Temporal decomposition is applied using a rolling horizon to the periods provided by the tactical level. Within each time period, for routing, a set partitioning problem is solved by optimizing algorithms using variable neighborhood search (VNS). The partitions are assigned to routes and evaluated by a routing heuristic. Therefore, a routing approach is developed comparing the costs and the benefit of serving a station. We test the approach with real-world data from Vienna's BSS "CityBike Wien" operating on 59 stations with two vehicles. The required target fill-levels are provided by Vogel et al. (2015). We examine the results regarding the number of vehicles, the neighborhood operators and the interval sizes. For two vehicles, our approaches achieve an overall service level of 91.36%. The service level shows significant differences during the day. In the peak hours, the vehicles are not able to rebalance the high number of trips.

This article is structured as follows: In section 2, we give a literature review of BSS related literature highlighting the work considering operational decision making. In section 3, the problem of relocations in BSS is formally defined as an integer program (IP). The decomposition approach is introduced in section 4. Computational studies and its results are presented in section 5. Finally, a conclusion is drawn in section 6.

2 Literature Review

Research on BSS has become vast during the last decade. In this section, we first give a short overview of different research areas concerning BSS. Then, we focus on the operational level of decision making.

Generally, objective is to install and maintain the functionality of a BSS in a cost efficient way, first time mentioned by Benchimol et al. (2011). This challenge can be approached on several management levels. A detailed definition and literature review of the different management levels can be found by Vogel et al. (2015). On the strategic level, decision about network design are made. Decisions contain the locations and sizes of stations and adaptions of the according infrastructure. Lin and Yang (2011) propose hub models to determine location and size of stations and identify the need for exclusive bike lanes. García-Palomares et al. (2012) implement location allocation models considering station capacity and user demands.

The tactical management addresses service network design. It deals with user activity patterns and suitable fill-levels for each station and point in time. Activity analysis has been done by Borgnat et al. (2011) and Vogel and Mattfeld (2011). Additionally, Vogel and Mattfeld (2011) stated, that stations can be clustered according to similar rental and return behavior. E. g., stations in residential areas show a different demand structure compared to stations in working areas. In Vogel et al. (2015), a resource allocation problem (RAP) is presented. Here, decisions about repositionings are made using a transportation model. So, routing is simplified by single transportation flows. The

Table 1: Inventory Routing Problems in Bike Sharing Systems

characteristic	Chemla et al. (2013)	Raviv et al. (2013)	Di Gaspero et al. (2013)	Rainer-Harbach et al. (2013)	Papazek et al. (2014)	Schuijbroek et al. (2013)	Contardo et al. (2012)	Kloimüllner et al. (2014)	presented problem setting
fleet of vehicles		✓	✓	✓	✓	✓	✓	✓	✓
limited time		✓	✓	✓	✓		✓	✓	✓
multiple visits	✓			✓	✓	✓	✓	✓	✓
service time		✓	(✓)	(✓)	(✓)			(✓)	✓
open tours						✓	✓		(✓)
target intervals						✓			✓
user activities							✓	✓	✓
multi-periodic							✓	✓	✓
min. tour length / time / costs	✓	✓	✓	✓	✓			✓	
min. makespan						✓			
min. gaps		✓	✓	✓	✓			✓	
min. gaps, squared									✓
min. repositioning operations			✓	✓	✓			✓	
min. unsatisfied user demands							✓	✓	

RAPs objective is to minimize expected operational costs, i.e., costs regarding realized tours, repositionings and accepted unserved user demands. Based on mentioned activity patterns, the RAP is solved by a hybrid metaheuristic. Regarding initial fill-levels, expected user activities and claimed relocation flows, suitable fill-levels for upcoming time periods are derived. These fill-levels serve as anticipating input for operational planning, i.e., if target fill-levels are realized, we suppose to satisfy the next period's demands.

The operational management faces vehicle routing problems with inventory decisions. These problems are well known as IRP (Dror et al., 1985). The work on IRPs regarding BSS differs in objective function and constraints. An overview regarding characteristics and problem settings is given in Table 1. The very right column gives a definition of the problem setting presented in this article (see section 3). In general, the approaches can be differentiated in single and multi-periodic rebalancing.

2.1 Single Periodic Rebalancing

At first, single periodic problem settings are investigated. In these problems, bikes are relocated over night, when the system is closed. User activities and time dependent fill-levels are not considered. Therefore, the target fill-levels represent optimal initial fill-levels of a day. Usually, closed tours are generated. A closed tour starts and ends at a given location, e.g., a depot. Chemla et al. (2013) present a problem setting, in which a single vehicle has to realize target fill-levels. Objective is to minimize the travel distance. The problem is defined by an exact IP and solved using several relaxions. Raviv et al. (2013) consider a fleet of vehicles and a limited time horizon. Service time is variable and depends on the number of repositioned bikes. Because of the time limit, realization of all target fill-levels may not be possible. The deviation of realized fill-levels and target fill-levels is called gap. Objective is to minimize the overall gap and the total travel and service times. Therefore, two models using arc-indexed 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 multi-criteria objectives are minimization of gaps, the total time and the number of repositioning operations. The authors emphasize the sum of gaps as most important impact to the objective function. As solution approaches, Di Gaspero et al. (2013) use constraint programming and ant colony optimization. Rainer-Harbach et al. (2013) apply a variable neighborhood search algorithm and Papazek et al. (2014) use a hybrid GRASP approach combining path relinking.

Schuijbroek et al. (2013) defined a single periodical mixed integer program allowing open tours. An open tour does not necessarily have identical start and end points. Hence, this approach can be extended for rolling horizon optimization of the multi-periodical case. Schuijbroek et al. (2013) compute target intervals for each station. The objective is to realize a specific service level considering target intervals within a minimal makespan, i.e., minimizing the maximum tour over all vehicles. Optimization is done via a cluster-first route-second algorithm and constraint programming.

In essence, all single-periodic approaches neglect user activities and time-dependent target fill-levels. So, for most BSS, single-periodic approaches are not applicable, because rebalancing is done during the day.

2.2 Multi-Periodic Rebalancing

Contardo et al. (2012) introduce a multi-periodic problem with time-depended pickup or delivery requests for each station. A pickup request stands for a surplus of bikes, whereas, a delivery request represents a shortage. If a request is not served or served partially, user demands are assumed to be unsatisfied at certain station. The objective is to minimize unsatisfied user demands. For modeling, an arc-flow formulation is applied. Optimization is carried out with Dantzig-Wolf and Benders decomposition approaches.

Kloimüllner et al. (2014) defines a multi-periodic problem setting considering time-depended user activities. Demands are modeled as expected values. A demand may be unserved, if a station's fill-level cannot comply the demand, e.g., if the sum of fill-level and demand would overrun the station's capacity. The objective includes the minimization of unserved user demands. Additionally, tour length, repositioning operations and gaps regarding target fill-levels at the end of the time horizon are considered. Optimization is done by Greedy Randomized Adaptive Search (GRASP) using VNS.

Many operational approaches include tactical and strategical decisions in the objective function. A multi-periodical approach only considering operational routing regarding the tactical fill-intervals as proposed by Vogel et al. (2015) is not defined yet. From an operational view, resources are given, i.e., fix costs for workers and vehicles are already paid. The according costs have to be considered by the operational management only as constraints. Further, anticipation of future customer behavior is already included in the tactical fill-levels. In essence, the operational approach mainly focus on routing fulfilling the tactical demands by given resources.

3 A Multi-Period Inventory Routing Problem for Bike Sharing Systems

In the following, we formulate a multi-periodic approach to balance a BSS over a whole working day with respect to changing target fill-intervals and user activities. Further, we define the necessary instance variables and model the problem as an integer program.

3.1 Problem Definition

Consider a BSS consisting of stations, bikes, capacitated relocation vehicles and a depot. Each station has a certain starting fill-level and a maximum capacity. The working day is divided in periods. Thus, we present a multi-periodic problem setting. For each period, stations are assigned target fill-intervals. Regarding the working day, all vehicles start and end their tours at the depot allowing open tours within the periods. Vehicles start empty at the depot in the first period and have to return empty in the last period. Stations may be visited multiple times by one vehicle and different vehicles. We consider a service time per reallocated bike. User activities (derived from the expected trips) are deterministic and are considered in the system's fill-levels at the begin of each period. Here, uncertainty in user activity is already incorporated on tactical level in the determination of appropriate intervals.

We consider spatio-temporal dependent target intervals given by tactical information systems, e.g., Vogel et al. (2015). If a station's fill-level differs from the target interval, a gap arises. A gap is the deviation between fill-level and target interval. The objective is to minimize the squared gaps over all stations at the end of all periods. Thus, large

Table 2: Instance Parameters

symbol	description
$G = (N, E)$	bike-sharing system
$N = \{n_0, n_1, \dots\}$	set of stations incl. depot n_0
$N^* = N \setminus \{n_0\}$	set of stations
$E = \{e_{ij} \colon n_i, n_j \in N\}$	set of edges
$V = \{v_1, v_2, \dots\}$	set of vehicles
$T = \{0, \dots, \bar{t}\}$	set of decision points, i.e., time horizon
$P \subseteq T$	set of period's last decision points
$\overline{c} \colon N \to \mathbb{N}_0$	upper station capacity
$\underline{c} \colon N \to \mathbb{Z}_0^-$	lower station capacity
$\overline{a} \colon N \times T \to \mathbb{Z}$	upper target interval limit
$\underline{a} \colon N \times T \to \mathbb{Z}$	lower target interval limit
$d \colon N \times T \to \mathbb{Z}$	user activities
$w \colon E \to \mathbb{R}^+$	edge weights, i.e., travel times
$l \colon V \to \mathbb{N}$	vehicle capacity
ρ	service time for repositioning one bike
M	sufficient large number

gaps are mostly avoided while small gaps are accepted. Gaps are measured at the end of each period.

3.2 Instances

An instance is defined by parameters listed in Table 2. The BSS is modeled as a complete graph with sets of stations and edges. The edges represent shortest paths between stations. The time horizon consists of discrete decision points. We want to point out, that station's capacities and fill-levels are modeled indirectly. I.e., $\overline{c}$ is the number of free racks in the beginning of the time horizon. $\underline{c}$ is the negative fill-level representing the maximum number of available bikes that can be removed. Thus, a station's capacity is equal to $\overline{c} - \underline{c}$.

The target intervals are modeled identically. If $\overline{a} > 0$, the current fill-level is below the upper target interval limit. $\overline{a}$ indicates how many bikes can be placed without exceeding the upper target interval limit. If $\overline{a} < 0$, the current fill-level is beyond the upper target interval limit. Then, $|\overline{a}|$ indicates how many bikes have to be removed to reach the target interval. $\underline{a}$ is defines analogously. If $\underline{a} < 0$, the current fill-level is beyond the lower target interval limit. Then, $|\underline{a}|$ indicates how many bikes can be removed without deceeding the lower target interval limit. If $\underline{a} > 0$, the current fill-level is below the lower target interval limit. Then, $\underline{a}$ indicates how many bikes have to be placed to reach the target interval.

If and only if

$$\overline{a} \geq 0 \wedge 0 \geq \underline{a} \quad \Rightarrow \quad \text{gap} = 0, \tag{1}$$

Table 3: Decision Variables

symbol	description
$\mathcal{X} = \left\{x_{ijwt} \colon x_{ijwt} \in \{0,1\}, e_{ij} \in E, v_w \in V, t \in T\right\}$	tours
$\mathcal{R} = \{r_{iwt} \colon r_{iwt} \in \mathbb{Z}, n_i \in N^*, v_w \in V, t \in T\}$	repositionings
$\mathcal{A} = \{ar_{iwt} \colon ar_{iwt} \in \mathbb{N}_0, r_{iwt} \in \mathcal{R}\}$	absolute repositionings
$\mathcal{G} = \{g_{it} \colon g_{it} \in \mathbb{N}_0, n_i \in N^*, t \in T\}$	gaps

i.e., a station's fill-level within the target interval leads to a gap equal to zero.

Since we only want to determine gaps at the end of a period, we define functions

$$\overline{a}(n_i, t) \begin{cases} \leq \overline{c}(n_i, t), \text{ if } t \in P \\ = \overline{c}(n_i, t), \text{ else} \end{cases} \tag{2}$$

and

$$\underline{a}(n_i, t) \begin{cases} \geq \underline{c}(n_i, t), \text{ if } t \in P \\ = \underline{c}(n_i, t), \text{ else} \end{cases} \tag{3}$$

appropriately.

3.3 Integer Program

The decision variables are shown in Table 3. Decision variables are gathered in sets for tours, repositionings, absolute repositionings and gaps. Tours are determined by variables

$$x_{ijwt} = \begin{cases} 1, \text{ if } v_w \text{ starts traveling from } n_i \text{ to } n_j \text{ in } t. \\ 0, \text{ else.} \end{cases} \tag{4}$$

Repositionings are denoted by

$$r_{iwt} \begin{cases} > 0, \text{ if } v_w \text{ places } r_{iwt} \text{ bikes into } n_i \text{ in } t. \\ < 0, \text{ if } v_w \text{ removed } |r_{iwt}| \text{ bikes from } n_i \text{ in } t. \\ = 0, \text{ else.} \end{cases} \tag{5}$$

The absolute number of repositionings are stated by

$$ar_{iwt} = |r_{iwt}|. \tag{6}$$

These variables are necessary to determine service times. Gaps are presented as

$$g_{it} \begin{cases} > 0, \text{ if } n_i\text{s fill-level is not within the target interval in } t \\ = 0, \text{ else.} \end{cases} \tag{7}$$

The integer program reads as follows:

$$\min \sum_{i=0}^{|N^*|} \sum_{t=0}^{\bar{t}} g_{it}^2 \tag{8}$$

subject to

$$\sum_{j=1}^{|N^*|} \sum_{t=0}^{\bar{t}} x_{j0wt} \cdot t \geq \sum_{j=1}^{|N^*|} \sum_{t=0}^{\bar{t}} x_{0jwt} \cdot t \quad \forall v_w \in V \tag{9}$$

$$\sum_{j=1}^{|N^*|} \sum_{t=0}^{\bar{t}-w(e_{j0})} x_{j0wt} = 1 \quad \forall v_w \in V \tag{10}$$

$$\sum_{j=0}^{|N^*|} \sum_{t=0}^{\bar{t}} x_{ijwt} = \sum_{j=0}^{|N^*|} \sum_{t=0}^{\bar{t}} x_{jiwt} \quad \forall n_i \in N, v_w \in V \tag{11}$$

$$\sum_{j=0}^{|N^*|} \sum_{t=0}^{\hat{t}} x_{jiwt} - \sum_{j=0}^{|N^*|} \sum_{t=0}^{\hat{t}} x_{ijwt} \geq 0 \quad \forall n_i \in N^*, v_w \in V, \hat{t} \in T \tag{12}$$

$$\sum_{j=0}^{|N^*|} \sum_{t=0}^{\hat{t}} x_{jiwt} - \sum_{j=0}^{|N^*|} \sum_{t=0}^{\hat{t}} x_{ijwt} \leq 1 \quad \forall n_i \in N^*, v_w \in V, \hat{t} \in T \tag{13}$$

$$x_{jiw\hat{t}} \cdot M + \sum_{k=0}^{|N^*|} \sum_{t=\hat{t}}^{\hat{t}+ar_{jw\hat{t}}\cdot\rho+w(e_{ji})} x_{ikwt} \leq M$$

$$\forall n_i, n_j \in N, v_w \in V, \hat{t} \in \left\{0, \ldots, \bar{t} - r_{jw\hat{t}} \cdot \rho - w(e_{ji})\right\} \tag{14}$$

$$\sum_{j=0}^{|N^*|} \sum_{t=0}^{\bar{t}} x_{ijwt} \cdot \big(t + w(e_{ij}) + w(e_{j0})\big) \leq \bar{t} \quad \forall n_i \in N, v_w \in V \tag{15}$$

$$\sum_{i=0}^{|N^*|} \sum_{w=0}^{|V|} \sum_{t=0}^{\bar{t}} x_{iiwt} = 0 \tag{16}$$

$$r_{0wt} = 0 \quad \forall v_w \in V, t \in T \tag{17}$$

$$\sum_{j=0}^{|N^*|} x_{ijwt} \cdot M \geq r_{iwt} \quad \forall n_i \in N, v_w \in V, t \in T \tag{18}$$

$$\sum_{j=0}^{|N^*|} x_{ijwt} \cdot M \geq \text{-}r_{iwt} \quad \forall n_i \in N, v_w \in V, t \in T \tag{19}$$

$$\sum_{i=0}^{|N^*|} \sum_{t=0}^{\hat{t}} \text{-}r_{iwt} \geq 0 \quad \forall v_w \in V, \hat{t} \in T \tag{20}$$

$$\sum_{i=0}^{|N^*|}\sum_{t=0}^{\hat{t}} \text{-}r_{iwt} \leq l(v_w) \quad \forall v_w \in V, \hat{t} \in T \tag{21}$$

$$\sum_{i=0}^{|N^*|}\sum_{t=0}^{\bar{t}} r_{iwt} = 0 \quad \forall v_w \in V \tag{22}$$

$$\sum_{w=1}^{|V|}\sum_{t=0}^{\hat{t}} r_{iwt} + \sum_{t=0}^{\hat{t}} d(n_i,t) \leq \overline{c}(n_i) \quad \forall n_i \in N^*, \hat{t} \in T \tag{23}$$

$$\sum_{w=1}^{|V|}\sum_{t=0}^{\hat{t}} r_{iwt} + \sum_{t=0}^{\hat{t}} d(n_i,t) \geq \underline{c}(n_i) \forall n_i \in N^*, \hat{t} \in T \tag{24}$$

$$ar_{iwt} \geq r_{iwt} \quad \forall n_i \in N, v_w \in V, t \in T \tag{25}$$

$$ar_{iwt} \geq \text{-}r_{iwt} \quad \forall n_i \in N, v_w \in V, t \in T \tag{26}$$

$$\left(\sum_{w=1}^{|V|}\sum_{t=0}^{\hat{t}} r_{iwt} + \sum_{t=0}^{\hat{t}-1} d(n_i,t)\right) - \overline{a}(n_i,t) \leq g_{it} \quad \forall n_i \in N, \hat{t} \in T \tag{27}$$

$$\underline{a}(n_i,t) - \left(\sum_{w=1}^{|V|}\sum_{t=0}^{\hat{t}} r_{iwt} + \sum_{t=0}^{\hat{t}-1} d(n_i,t)\right) \leq g_{it} \quad \forall n_i \in N, \hat{t} \in T \tag{28}$$

The objective function (8) minimizes the sum of squared gaps over all stations and periods. Each vehicle leaves the depot before it returns (9) within the time horizon (10). Each node is visited and left by a single vehicle the same number of times (11). Visits and returns have to be done alternatingly (12, 13). If a vehicle leaves a station, it cannot leave the destination before it has arrived there (14). A station can be visited, if returning to the depot is possible in time (15). To avoid short cycles, a vehicle cannot visit a station where it currently stays (16). At the depot, no repositionings can be done (17). Repositionings are possible, if the vehicle stays at related station (18, 19). One vehicle's cumulated repositionings have to match the vehicle's capacity at all time (20, 21). In sum, each vehicle's repositioning operations have to be balanced (22). Repositionings and user activities must not violate the station's capacities (23, 24). Absolute repositionings are determined as the repositionings absolute value (25, 26). Finally, the gaps are calculated (27, 28).

4 A Decomposition Approach with Variable Neighborhood Search

Due to the large number of integer decision variables, problem instances cannot be solved exactly within reasonable time. Thus, we propose a decomposition approach as shown in Figure 1. The decomposition is applied in temporal and spatial dimension. Temporal, we use a rolling horizon. Spatial, we combine a set partitioning with a (variable) neighborhood search. To solve the routing problem, we define a cost-benefit heuristic.

4.1 Decomposition

Anticipation of future customer demands is included in the design of the target interval. Hence, even though we use a temporal decomposition, the achieved solutions are anticipatory. The temporal decomposition is realized by optimization over a rolling horizon (Baker, 1977). Thus, the periods as described in section 3 are optimized sequentially. I.e., we start the first period with its target fill-intervals. Afterwards, user activities of the associated period are added. Then, the optimization is done regarding the next period's target fill-intervals realizing fill-levels. The procedure continuous until the last period is finished.

We use spatial decomposition to reduce the number of routing decisions. The set of stations N^* is partitioned into subsets. The stations in each subset are assigned to a vehicle. Efficient subsets consider both short distances between stations and a balance in bike surplusses and shortages. A generalization of this problem is given by Balas and Padberg (1976). A set partitioning problems solution consists subsets N_i^* that satisfy both

$$N_1^* \cup N_2^* \cup \dots \cup N_{|V|}^* = N^* \tag{29}$$

and

$$N_i^* \cap N_j^* = \emptyset \quad \forall N_i^*, N_j^* \subseteq N^*, i \neq j. \tag{30}$$

4.2 Variable Neighborhood Search

The set partitioning problem is tackled by different local search algorithms iteratively. For that, we define two operators for altering partitions, respectively solution.

- INSERT changes one station's assignment. I.e., one station is removed from its subset and is assigned to another.
- EXCHANGE refers to two stations of different seubsets. The first station is assigned to the second station's subsets and vice versa.

The procedure described is depicted in Figure 1. The application of an operator spans one solutions neighborhood. In each iteration, the current solutions neighborhood is evaluated by a routing algorithm (section 4.3). According to the search algorithm, a new solution from the neighborhood is selected for the next iteration until a stop criterion is reached. To combine Insertion and Exchange, a variable neighborhood is examined. I.e., first the small neighborhood (Insert) is examined. When a stop criterion is reached, the neighborhood is enlarged (Exchange).

Three local search algorithms are applied. Hill Climbing always chooses the best solution from a neighborhood until a local optimum is found (Mattfeld, 1996). Tabu

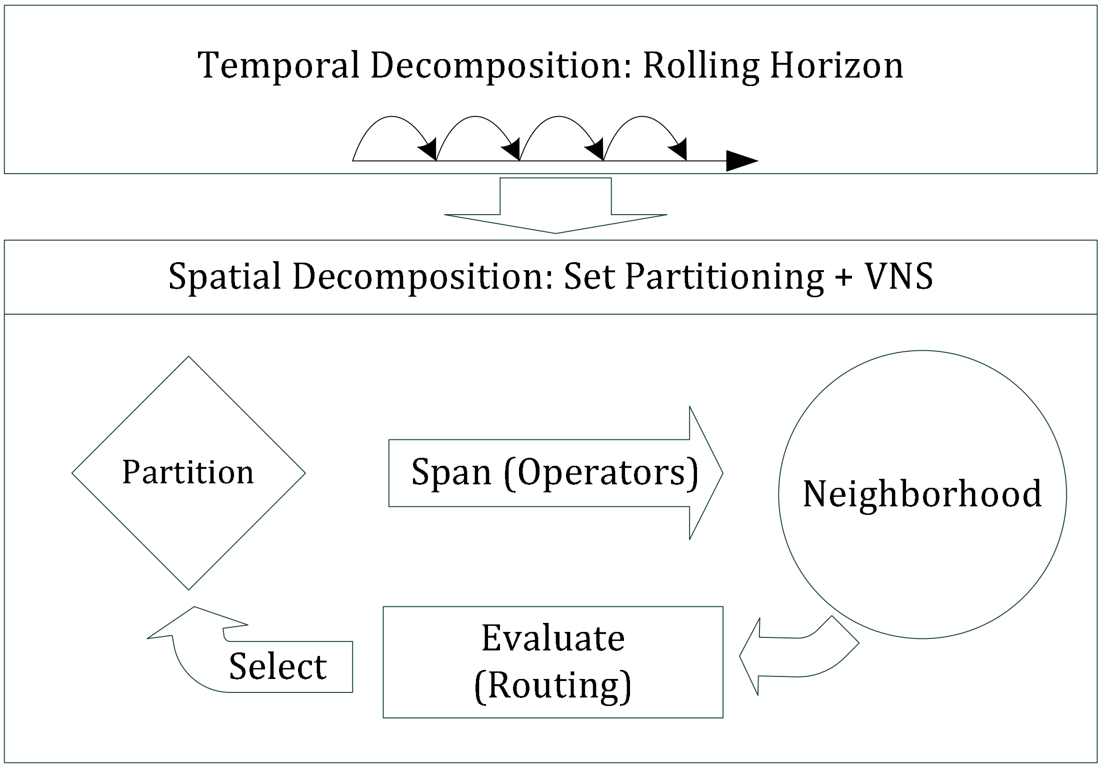

Figure 1: Temporal and Spatial Decomposition Approach with Variable Neighborhood Search

Search prohibits a set of solutions on a tabu list to overcome local optimality. Here, the tabu list contains the last solutions (Rothlauf, 2011). Simulated Annealing randomly chooses a solution from the neighborhood. Depending on the objective value and a cool down parameter, the solution choosen is accepted (Aarts and Korst, 1989).

4.3 Cost-Benefit Routing

For evaluation of the partition's subsets, we define an inventory routing algorithm comparing costs and benefit of visiting a station. The goal is to generate a tour regarding repositioning operations for each subset. Routes are generated here by an adapted nearest neighbor. Usually, according to Schneider and Kirkpatrick (2006), the next vehicle's destination is the nearest station that has not been visited yet. In our approach, we also take possible repositionings into account. Depending on the number of bikes loaded and the vehicle's capacity, we consider the possible gap reduction of a station. We call the squared gaps maximum reduction $\triangle g^2$. The time for conducting such reduction depends on the travel time w, the absolute number of repositionings ar, and on the service time ρ as well. Now, a

$$score = \frac{\triangle g^2}{w + ar \cdot \rho} \tag{31}$$

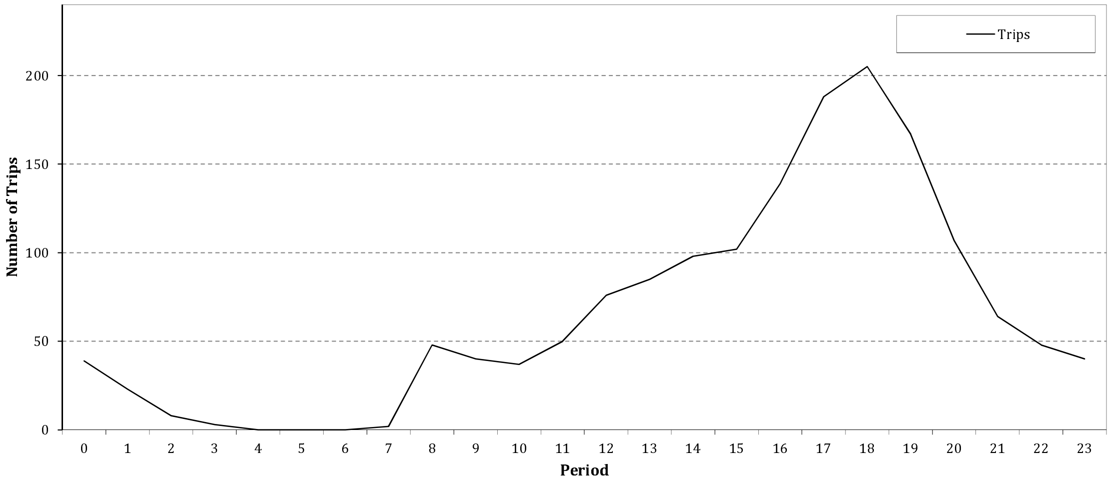

Figure 2: Number of Trips per Period for a Typical Working Day

is calculated for each station within the given subset. I.e., we balance costs and benefit. The station with highest *score* is the vehicle's next destination. Destinations and repositioning operations are iteratively generated, as long as the time limit of the period is not exceeded. The sum of resulting squared gaps over all stations is the partition's objective value. Hence, the sum over all partitions and periods is the overall objective value.

5 Computational 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 algorithms impact to routing and fill-levels. Finally, we compare the results with the transportation problem provided by Vogel et al. (2015).

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. The BSS consists of 59 stations and 627 bikes. 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 $25\frac{km}{h}$. The service time required for the reposition of a bike is $2min$. We examine the results for $2, 3, 4$, and 8 homogeneous vehicles with capacity of 20 bikes. Station capacities differ between 10 and 40. A detailed analysis of the spatial-temporal customer behavior can be found in Vogel and Mattfeld (2011). Typical flows of bikes for a work day have been extracted by Vogel et al. (2015). These flows contain 1,569 single trips per day. The amount of trips over the day is shown in Figure 2.

User activities start at 8 a.m. and increase up to an overall peak at 6 p.m. A majority of customers use the BSS regularly to commute between home and work. In the morning, stations in the city center congest, while stations in residential areas run out of bikes.

For the typical flows, optimal fill-levels for station and hour of the day have been generated. The preprocessed data serves as input for our computational studies. The according intervals have a size of 5 bikes. For two stations, Figure 3 depicts exemplarily the development of target intervals during the day. The graphs also contain fill-level realizations provided by the solution approach. The target intervals change hourly. The graph on the left side shows intervals for a station in a residential area. As assumed, residential stations require a lot of bikes in the beginning of the day for users commuting to work. In the evening, target intervals are low to allow commuters to return their bikes. In the graph on the right, intervals for a working station in the city center can be seen. Target intervals show a opposing evolution over the day compared to the residential station.

5.2 Algorithm Tuning

To apply VNS, an initial partition is randomly generated. Except Hill Climbing, the different local search algorithms requires parameter settings. In Tabu Search, the length of the tabu list is set to 100. Iterations are performed until the algorithm reaches a setting for the second time. I.e., the same solution is found twice regarding an identical tabu list. In Simulated Annealing, parameters are chosen according to Rothlauf (2011). The cool down parameter differs between 0.99 and 0.999 and the initial temperature is initialized as the objective values standard deviation on a sample of randomly generated solutions. Since the objectives differs with a different number of vehicles, the initial temperature τ differs between 64.0 and 256.0. The procedure stops when the probability equal to $e^{\frac{-1}{\tau}}$ of accepting an inferior solutions falls below 0.01. The according minimal temperature is 0.2.

5.3 Test Results

First, we depict the advantages of a variable neighborhood search. Second, we compare the different neighborhood approaches regarding solution quality. For Simulated Annealing, we exemplary analyze the solutions structure in detail considering routing and fill levels for different stations types during the day.

To examine the solution quality of the different operator settings, we run 100 tests for the single operators and the operator combination considering different numbers of vehicles. Solution quality and variance for Hill Climbing are shown in Table 4. VNS, i. e., the combination of the two operators, outperforms the single operators by far for every instance. Further, VNS reduces the objective value significantly up to 20.5%

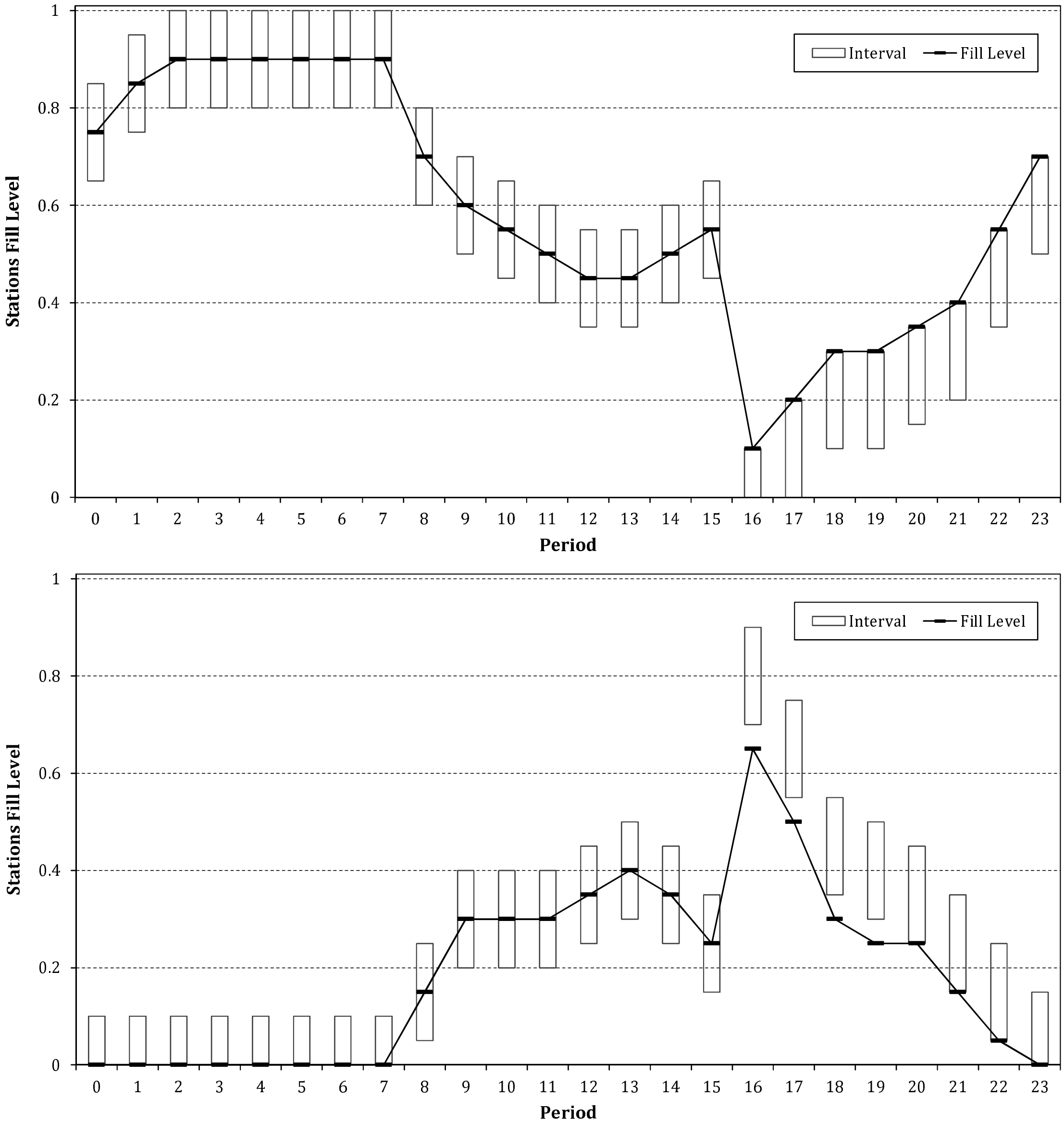

Figure 3: Target Intervals and Realized Fill Levels for a Residential Station

Table 4: operator comparison: average squared gaps / sample variance

operators \ vehicles	2	3	4	8
Insert	174.66/764.33	129.24/238.63	126.04/327.33	116.10/315.91
Exchange	165.19/355.97	123.20/133.48	122.50/240.11	134.36/463.99
Insert/Exchange	155.68/274.56	116.70/115.80	112.90/142.46	106.77/154.50

Table 5: algorithm comparison: average / sample variance

algorithm \ vehicles	2	3	4	8
Squared Gaps				
Hill Climbing	155.88/325.28	116.02/134.16	113.76/210.53	108.63/148.34
Tabu Search	139.28/108.20	105.07/ 83.54	104.19/101.69	103.19/111.15
Simulated Annealing	135.63/125.23	104.81/ 73.49	100.77/100.32	91.87/ 88.62
Runtime (in seconds)				
Hill Climbing	3.28	2.90	2.81	3.20
Tabu Search	131.53	81.78	51.52	171.46
Simulated Annealing	245.09	45.16	37.67	14.03
Service Level (in %)				
Hill Climbing	90.07	92.61	92.75	93.08
Tabu Search	91.12	93.30	93.36	93.42
Simulated Annealing	91.36	93.32	93.58	94.14
Gaps				
Hill Climbing	91.16/24.50	75.94/25.55	71.34/34.07	67.37/43.12
Tabu Search	86.48/ 9.59	74.51/23.26	71.57/40.45	69.63/36.80
Simulated Annealing	86.11/13.23	75.01/23.34	69.03/44.37	63.15/34.61

compared to the single approaches. The variance is reduced up to 66.7%. So, in the following, we use VNS to compare the different search algorithms.

The results for the different local search algorithms combined with VNS and the according runtimes shown in Table 5 (Experiments are performed on a Intel Core i5-3470 @ 3.2GHz with 32GB RAM).

Simulated Annealing outperforms the two other approaches followed by Tabu Search. As expected, Hill Climbing is not able to reach similar results as the metaheuristics. The average increase in objective value compared to Simulated Annealing is 14.1%, in variance even 111.1%. The high variance indicates that for Hill Climbing the quality of the initial partition is essential for finding good solutions. Simulated Annealing provides high quality solutions regardless the design of the initial partition. The more vehicles, i.e., the higher the number of subsets in a partition, the more advantageous is the application of Simulated Annealing. Especially for 8 vehicles, the stations in each subset of the initial partition may be spatially divided and the sum of bike shortages and surpluses may be highly imbalanced. Here, Simulated Annealing outperforms both approaches by more than 12.3%. The runtimes for Simulated Annealing decreases significantly regarding the number of vehicles while, as comparison, Hill Climbing requires nearly a contant amount of time regardless the number of vehicles. Hence, especially for a high number of subsets, Simulated Annealing achieves fast high quality solution.

Because the algorithms are applied to typical flows, the trips are deterministic and smoothed (Vogel et al., 2015). Hence, a direct calculation of the number of unfulfilled

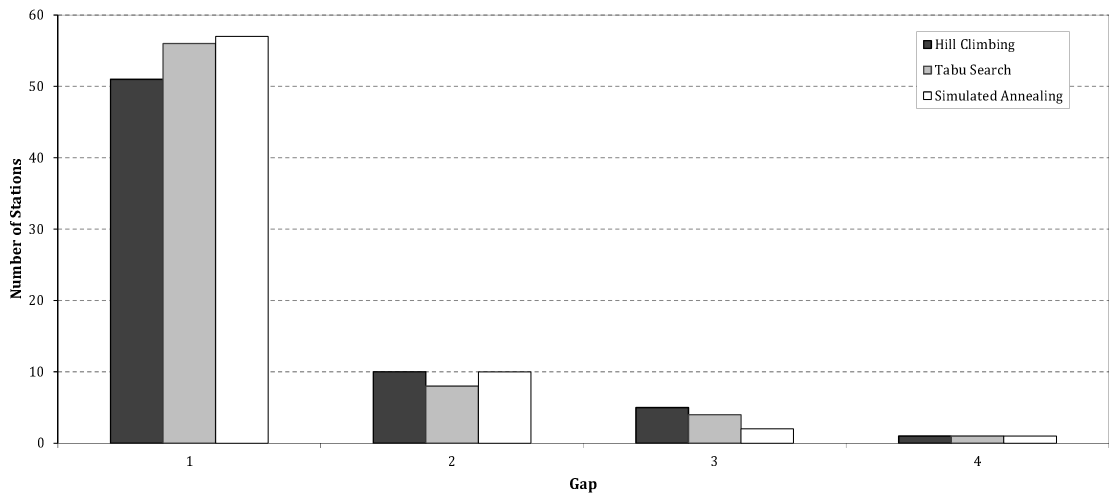

Figure 4: Gap Sizes of Imbalanced Stations for 2 Vehicles

demands and, therefore, the service level is not possible. We have to estimate the number of unfulfilled demands. We assume to satisfy all demands, if a station's fill-level is within the associated target interval. If an interval is not reached, the possibility of an unfulfilled demand increases significantly with the respective gap. Hence, we estimate the service level as the relation between the sum of squared gaps $\sum g^2$ compared to the overall number of trips $\sum d$.

$$servicelevel = 1 - \frac{\sum g^2}{\sum d} \tag{32}$$

The results can be seen in Table 5. The overall service levels differ from 90.07% up to 94.14% regarding solution approach and number of vehicles. So, in general, more than 9 out of 10 demands can be fulfilled. Adding another vehicle to the two existing increases the service level about 2.15%, whereas the addition of a fourth vehicle only leads to a slight increase of 0.27%. For this small instance, more than three vehicles may not be necessary to maintain the system.

To examine the solution structure in more detail, we additionally analyze the average gaps and associated sample variances displayed in Table 5. Here, the gaps differ less in relation to the objective values. For some instances, Hill Climbing even achieves lower gaps than Tabu Search. Figure 4 shows the distribution of gaps for the different algorithms and two vehicles. As we can we see, the number of imbalanced stations for Hill Climbing is lower than for the other approaches whereas the gap sizes are larger. While Tabu Search and Simulated Annealing allows slight imbalances in many stations, Hill Climbing is not able to avoid a significant amount of stations with higher gaps resulting in relatively poor solution quality.

To further analyze the solution structure, in the following, we exemplary show the results for Simulated Annealing and two vehicles. First, we analyze the solution results during the day. Figure 5 depicts the number of imbalanced stations, the gaps in bikes

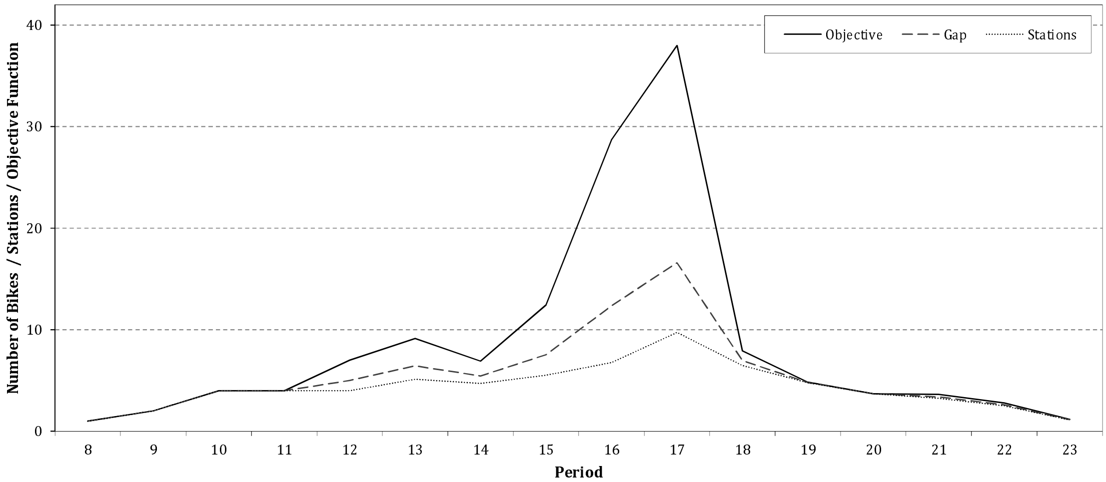

Figure 5: Number of Imbalanced Stations, Bike/Rack Gaps and Objective Function over the Periods for Simulated Annealing, 2 Vehicles

and racks, and the objective function over the periods. The structure matches the customer behavior shown in Figure 2. Between periods 0 and 7 only a small number of trips occur and no interval violation is discovered. The first violations can be experienced in period 8, when the morning peak occurs. During the day, the violations increase until the main peak between period 16 and 18. Here, in average, 10 stations are imbalanced. These imbalances are subsequently reduced during the rest of the day.

In Figure 3, the fill levels per period for a residential and a working station are shown. There is no repositioning before period 17. Then, a vehicle picks up 11 bikes from the residential station seen on the left and delivers 10 to the working station seen on the right. The 11th bike is unloaded on a station in between. So, while the target interval for the residential station can be reached, for the working station, the missing bike is not unloaded until period 20.

As expected, two vehicles are not able to maintain the service level in this time periods, the service level drops to 79.3%. Even with eight vehicles, the service level in period 17 remains low with only 81.8%. Here, the provided target fill intervals might be impracticable. The transportation problem applied by Vogel et al. (2015) to estimate the repositioning tours has no capacity contraints per period. The resulting flows are compared to the realized repositionings in Figure 6. The correlation between flows and repositionings is significant. But, while there is no transportation flow later than period 17, repositionings have to be carried out until the last period. Here, the number even increases, due to the constraint that the vehicles have to return to the depot empty. This indicates that the matching between transportation problem and routing might be weak within peak periods and temporal capacity constraints applied to the transportation problem could improve the matching and the solution quality significantly.

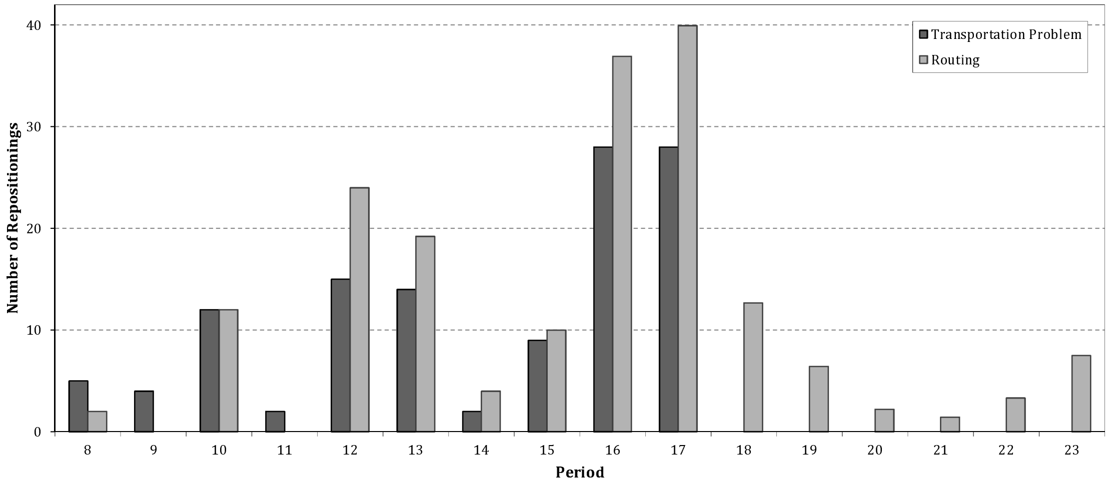

Figure 6: Repositionings per Period: Comparison of Transportation Problem by Vogel et al. (2015) and Routing Approach

6 Conclusion

We have introduced a multi-periodic problem setting for rebalancing bike sharing systems. Time-dependent target intervals anticipating future demands for stations are provided by tactical information systems. These target intervals have to be realized by a fleet of capacitated transportation vehicles. Thus, inventory and routing decision have to be made. We defined the problem using an integer program. To solve problem instances, a decomposition approach is presented. The decomposition is implemented both temporal and spatial. I.e., a given time horizon is divided into periods. Optimization is performed over a rolling horizon. Subsequently, the set of stations is divided into partitions. In each partition, a vehicle realizes reposition operations. The set partitioning problem is tackled via (variable) neighborhood search by metaheuristics, i.e., Tabu Search and Simulated Annealing. Therefore, initial partition are generated randomly and improved iteratively. To evaluate partitions, an myopic inventory routing algorithm comparing costs and benefits of visiting a station is introduced.

We present a computational case study based on real world data of Vienna's bike sharing system "CityBike Wien". Results show that the selection of partitions has a huge impact on inventory routing. For this problem, variable neighborhood search is advantageous. Compared to a greedy procedure, the applied metaheuristics lead to significant better solutions regarding solution quality. Using randomly created initial partitions, the metaheuristics are reliable allowing good solutions and a low sample variance. Increasing the number of vehicles, Tabu Search cannot achieve better results. In that case, Simulated Annealing leads to significant better results. Thus, we assume Simulated Annealing to be suitable for large instances. Investigations on realized fill-level and target intervals reveal that significant deviations occur mainly during peak hours. The consideration of routing on tactical level using a uncapacitated transportation problem may be impracticable. Including temporal and spatial capacity constraints may lead here to more suitable fill levels.

In future work, the approach might be validated for larger BSS and extended. E. g., optimization regarding target intervals of the following periods could lead to explicit anticipatory inventory routing. Additionally, simulation of stochastic customer requests may allow insights over the reliability of fill levels and routing approach. Generally, further research should concentrate on a close connection between operational and tactical management. E. g., initial partitions could be created in advanced on tactical level to support operational optimization.

Acknowledgements The authors thank Patrick Vogel for the provision of the target intervals, and his helpful suggestions and comments in preparation of this work. We also thank Gewista Werbegesellschaft mbH for providing the required data from their project "Citybike Wien".

References

Aarts E, Korst J (1989) Simulated Annealing and Boltzmann Machines. John Wiley & Sons Ltd.

Baker KR (1977) An experimental study of the effectiveness of rolling schedules in production planning. Decision Sciences 8(1):19–27

Balas E, Padberg MW (1976) Set Partitioning: A survey. SIAM Review 18(4):710–760

Benchimol M, Benchimol P, Chappert B, De La Taille A, Laroche F, Meunier F, Robinet L, et al. (2011) Balancing the stations of a self-service bike hire system. RAIRO-Operations Research 45(1):37–61

Borgnat P, Abry P, Flandrin P, Robardet C, Rouquier JB, Fleury E (2011) Shared bicycles in a city: A signal processing and data analysis perspective. Advances in Complex Systems 14(03):415–438

Büttner J, Mlasowsky H, Birkholz T, et al. (2011) Optimising Bike Sharing in European Cities - A Handbook. OBIS project, `http://www.obisproject.com` (2014-05-29)

Chemla D, Meunier F, Wolfler Calvo R (2013) Bike sharing systems: Solving the static rebalancing problem. Discrete Optimization 10(2):120–146

Contardo C, Morency C, Rousseau LM (2012) Balancing a Dynamic Public Bike-Sharing System. CIRRELT-2012-09, `https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2012-09.pdf` (2014-12-08)

Di Gaspero L, Rendl A, Urli T (2013) A Hybrid ACO+CP for Balancing Bicycle Sharing Systems. In: Hybrid Metaheuristics, Lecture Notes in Computer Science, Springer, pp 7919:198–212

Dror M, Ball M, Golden B (1985) A computational comparison of algorithms for the inventory routing problem. Annals of Operations Research 4(3–23)

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

Gewista Werbegesellschaft mbH (2014) Citybike Wien. `http://www.citybikewien.at` (2014-06-12)

Kloimüllner C, Papazek P, Hu B, Raidl GR (2014) Balancing Bicycle Sharing Systems: An Approach for the Dynamic Case. In: Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science, Springer, pp 8600:73–84

Lin JR, Yang TH (2011) Strategic design of public bicycle sharing systems with service level constraints. Transportation Research Part E: Logistics and Transportation Review 47(2):284–294

Mattfeld DC (1996) Evolutionary Search and the Job Shop: Investigations on Genetic Algorithms for Production Scheduling. PhD thesis, Universität Bremen

Papazek P, Kloimüllner C, Hu B, Raidl GR (2014) Balancing Bicycle Sharing Systems: An Analysis of Path Relinking and Recombination within a GRASP Hybrid. In: Parallel Problem Solving from Nature PPSN XIII, Lecture Notes in Computer Science, Springer, pp 8672:792–801

Rainer-Harbach M, Papazek P, Hu B, Raidl GR (2013) Balancing bicycle sharing systems: a variable neighborhood search approach. In: Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science, Springer, pp 7832:121–132

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

Rothlauf F (2011) Design of Modern Heuristics: Priciples and Applications. Springer

Schneider JJ, Kirkpatrick S (2006) Stochastic Optimization, 1st edn. Springer

Schuijbroek J, Hampshire R, van Hoeve WJ (2013) Inventory Rebalancing and Vehicle Routing in Bike Sharing Systems. Tepper School of Business, Paper 1491, `http://repository.cmu.edu/tepper/1491` (2014-05-20)

Vogel P, Mattfeld DC (2011) Strategic and Operational Planning of Bike-Sharing Systems by Data Mining A Case Study. In: Computational Logistics, Lecture Notes in Computer Science, Springer, pp 6971:127–141

Vogel P, Ehmke JF, Mattfeld DC (2015) Decision Support for Service Network Design of Bike Sharing Systems. under review