Hierarchical Clustering and Multilevel Refinement for the Bike-Sharing Station Planning Problem

Christian Kloimüllner and Günther R. Raidl

Institute of Computer Graphics and Algorithms, TU Wien
Favoritenstraße 9–11/1861, 1040 Vienna, Austria
{kloimuellner|raidl}@ac.tuwien.ac.at

Abstract. We investigate the Bike-Sharing Station Planning Problem (BSSPP). A bike-sharing system consists of a set of rental stations, each with a certain number of parking slots, distributed over a geographical region. Customers can rent available bikes at any station and return them at any other station with free parking slots. The initial decision process where to build stations of which size or how to extend an existing system by new stations and/or changing existing station configurations is crucial as it actually determines the satisfiable customer demand, costs, as well as the rebalancing effort arising by the need to regularly move bikes from some stations tending to run full to stations tending to run empty. We consider as objective the maximization of the satisfied customer demand under budget constraints for fixed and variable costs, including the costs for rebalancing. As bike-sharing stations are usually implemented within larger cities and the potential station locations are manifold, the size of practical instances of the underlying optimization problem is rather large, which makes a manual decision process a hardly comprehensible and understandable task but also a computational optimization very challenging. We therefore propose to state the BSSPP on the basis of a hierarchical clustering of the considered underlying geographical cells with potential customers and possible stations. In this way the estimated existing demand can be more compactly expressed by a relatively sparse weighted graph instead of a complete matrix with mostly small non-zero entries. For this advanced problem formulation we describe an efficient linear programming approach for evaluating candidate solutions, and for solving the problem a first multilevel refinement heuristic based on mixed integer linear programming. Our experiments show that it is possible to approach instances with up to 2000 geographical cells in reasonable computation times.

Keywords: Bike-Sharing Station Planning Problem · Hierarchical Clustering · Multilevel Refinement · Facility Location Problem

1 Introduction

Many large cities around the world have already built bike sharing systems (BSS), and many more are considering to introduce one or extend an existing one. These systems consist of rental stations around the city or a certain part of it where customers can rent and return bikes. A rental station has a specific number of parking slots where a bike can be taken from or returned to. On the contrary to bike-rental systems, BSSs encourage a short-term usage of bikes. As bikes are typically returned at a different station than they have been taken from, a need for active rebalancing arises as the demand for bikes to rent and parking slots to return bikes is not equally distributed among the stations.

Finding a good combination of station locations and building these stations in the right size is crucial when planning a BSS as these stations obviously directly determine the satisfied customer demand in terms of bike trips, the arising rebalancing effort, and the resulting fixed and variable costs. Stations close to public transport, business parks, or large housing developments will likely face a high demand whereas stations in sparser inhabited areas will probably face a lower demand. However, also the station density and connectedness of the actual regions to be covered play crucial roles. Some solitary station that is far from any other station will most likely not fulfill much demand. Moreover, a clever choice of station locations might also exploit the natural demands and customer flows in order to keep the rebalancing effort and associated costs reasonable.

As BSSs are usually implemented in rather large cities the problem of finding optimal locations for rental stations and sizing these stations appropriately is challenging and manually hardly comprehensible. Thus, there is the need for computational techniques supporting this decision-making. Besides fixed costs for building the system, an integrated approach should also estimate maintenance and rebalancing costs over a certain time horizon such that overall costs for the operator can be approximated more precisely. It is further important to consider the customer demands in a time-dependent way because there usually exists a morning peak and an afternoon peak which is due to commuters, people going to work, and students. Between these peaks, the demand of the system is usually a bit lower. We refer to this problem as *Bike Sharing Station Planning Problem* (BSSPP). The objective we consider here is to determine for a specified total-cost budget and a separate fixed-cost budget a selection of locations where rental stations of an also to be determined size should be erected in order to maximize the actually fulfilled customer demand.

In this work, we first concentrate on how to efficiently model the BSSPP such that we can also deal with very large instances with thousands of considered geographical cells for customers and potential station locations. To this end we propose to utilize a hierarchical clustering to express the estimated potential customer demand on it. We will then describe a *linear programming* (LP) based method to evaluate candidate solutions, and finally present a first novel *multilevel refinement heuristic* (MLR), based on mixed integer linear programming (MIP), to approach the optimization problem.

In Section 2 we discuss related work. Section 3 defines the BSSPP formally, also introducing the hierarchical clustering. Sections 3.3 and 3.4 describe LP models for determining the actually fulfilled customer demands for a candidate solution and estimating the required rebalancing effort, respectively. The MLR is then described in Section 4. First computational results on randomly generated instances are shown in Section 5, and finally, conclusions are drawn in Section 6.

2 Related Work

There already exists some work which tries to find optimal station locations for BSSs, although mostly considering different aspects. To the best of our knowledge, Yang et al. [12] were the first who considered the problem in 2010. They relate the problem to *hub location problems*, a special variant of the well-known *facility location problem*, and propose a mathematical model for it. The considered objective is to minimize the walking distance by prospective customers, fixed costs, and, a penalty for uncovered demands. The authors solve the problem by a heuristic approach in which a first part of the algorithm tries to identify the location of rental stations and a second, inner part tries to find shortest paths between origin and destination pairs. The authors illustrate their approach by a small example consisting of 11 candidate cells for bike stations.

Lin et al. [6] propose a mixed integer non-linear programming model and solve a small example instance with 11 candidate stations by the commercial solver LINGO, and furthermore provide a sensitivity analysis. Martinez et al. [8] develop approaches for a case study within Lisbon having 565 prospective candidate stations. They propose a hybrid approach consisting of a heuristic part utilizing a *mixed integer linear programming* (MIP) formulation. Locations as well as the fleet dimension are optimized, e-bikes are also considered, and rebalancing requirements are estimated.

Lin et al. [7] propose a heuristic algorithm for solving the hub location inventory problem arising in BSSPP. They do not only optimize station locations but their algorithm also identifies where to build bike lanes. As a subproblem they have to determine the travel patterns of the customers, i.e., solve a flow problem for a given configuration. They illustrate their approach on a small example consisting of 11 candidate locations for stations. Saharidis et al. [9] propose a MIP formulation which minimizes unmet demands and walking distance for prospective customers. They test their approach in a case study for the city center of Athens having 50 candidate cells for stations. Chen et al. [1] provide a mathematical non-linear programming model and solve the problem utilizing an improved immune algorithm. They define three different types of rental stations depending on their location (e.g., near a metro station, supermarkets). Their aim is that stations in the residential area have enough bikes available such that the morning peak can be managed and that stations near metro lines or important places have enough free parking slots available to manage incoming bikes during the morning peak. They provide a case study for a particular metro line of Nianjing city including 10 district stations and 31 residential stations. In [2] Chen and Sun aim at satisfying a given demand and minimizing travel times of the users. The authors propose an integer programming model which they solve with the LINGO solver. A computational analysis is provided on a small example. Frade et al. [3] describe an approach for a case study of the city of

Coimbra, Portugal. They present a compact MIP model which they solve using the XPRESS solver. Their objective is to maximize the demand covered by the BSS under budget constraints. They also include the net revenue in their mathematical model which reduces the costs incurred by building the BSS. Their single test instance consists only of 29 cells or *traffic zones*, how they call it. Hu et al. [5] also present a case study for a BSS along a metro line. They aim at minimizing total costs incurred by building particular BSS stations. In their computational study they consider three scenarios, each consisting of ten possible station candidates. They solve the proposed MIP model by the LINGO solver. Last but not least, Gavalas et al. [4] summarized diverse algorithmic approaches for the design and management of vehicle-sharing systems.

We conclude that all previous works on computational optimization approaches for designing BSS only consider rather small scenarios. Most previous work accomplishes the optimization with compact mathematical models that are directly solved by a MIP solver. Such methods, however, are clearly unsuited for tackling large realistic scenarios of cities with up to 2000 cells or more. In the following, we therefore propose a novel multilevel refinement heuristic based on a hierarchical clustering of the demand data.

3 Problem Formalization

The considered geographical area is partitioned into cells. Let S be the set of cells where a BSS station may potentially be located (*station cells*), and let V be the set of cells where some positive travel demand (outgoing, ingoing, or both) from prospective customers of the BSS exists (*customer cells*).

To handle such a large number of cells effectively, we consider a hierarchical abstraction as crucial in order to represent and model the further data in a meaningful and relatively compact form. To this end, we are expecting a hierarchical clustering of all customer cells V as input.

This hierarchical clustering is given in the form of a rooted tree with the inner nodes corresponding to clusters and the leafs corresponding to the cells. All cells have the same depth which is equal to the height of the tree, denoted by h. Let $C = C_0 \cup \ldots \cup C_h$ be the set of all tree nodes, with C_d corresponding to the subset of nodes at depth $d = 0, \ldots, h$. $C_0 = \{0\}$ contains only the root node 0 representing the single cluster with all cells, while $C_h = V$. Let $\mathrm{super}(p) \in C$ be the immediate predecessor (parent cluster) of some node $p \in C \setminus C_0$ and $\mathrm{sub}(p) \subset C$ be the set of immediate successors (children) of a cluster $p \in C \setminus C_h$.

As the travel demand of potential users varies over time we are given a (small) set of periods $T = \{1, \ldots, \tau\}$ for a "typical" day for which the planning shall be done. The estimated existing travel demand occurring in each period $t \in T$ from/to any cell $v \in V$ is given by a weighted directed graph $G^t = (C^t, A^t)$. All relevant outgoing travel demand at a cell v is represented by outgoing arcs $(v, p) \in A^t$ with $p \in C$ and corresponding values (weights) $d^t_{v,p} > 0$, i.e., (v, p) represents all expected trips from v to any cell represented by p in period t that might ideally be satisfied, and $d^t_{v,p}$ indicates the expected number of these trips.

Moreover, for each time period $t \in T$ we are given its duration denoted by δ_t^{period} and we are given a global parameter δ^{rent} which defines the average duration of a single trip performed by some user of the BSS.

The following conditions must hold to keep this graph as compact and meaningful as possible: the target node p of an arc (v, p) must not be a predecessor of v in the cluster tree. Self-loops (v, v), however, are allowed and important to model demand where the destination corresponds to the origin, arcs representing a neglectable demand, i.e., below a certain threshold, shall be avoided. Consequently, if there is an arc (v, p) no further arc (v, q) is allowed to any node q being a successor or a predecessor of p.

All estimated ingoing travel demand for each cell $v \in V$ is given correspondingly by arcs $(p, v) \in A^t$ with $p \in C$ with demand values $d_{p,v}^t \geq 0$, and corresponding conditions must hold.

Furthermore, it is an important property, that ingoing and outgoing demands have to be consistent: Let us denote by $V(p)$ the subset of all cells from V contained in cluster $p \in C$, i.e., the leafs of the subtree rooted in p, and by $C(p)$ the subset of all the nodes $q \in C$ that are part of the subtree rooted in p, including p and $V(p)$. For any $p \in C \setminus V$ it must hold that

$$\sum_{(v,q)\in A^t | v\in V(p), q\notin C(p)} d_{v,q}^t \geq \sum_{(q,v)\in A^t | q\in C(p), v\notin V(p)} d_{q,v}^t \tag{1}$$

and

$$\sum_{(q,v)\in A^t | q\notin C(p), v\in V(p)} d_{q,v}^t \geq \sum_{(v,q)\in A^t | v\notin V(p), q\in C(p)} d_{v,q}^t. \tag{2}$$

Condition (1) ensures that the total demand originating at the leafs of the subtree rooted at p and leading to a destination outside of the tree is never less than the total ingoing demand at all the cells outside the tree originating from some cluster inside the tree. Condition (2) provides a symmetric condition for the total ingoing demand at all the leafs of the tree. Furthermore, for the root node $p = 0$, inequalities (1) and (2) must hold with equality.

For each customer cell $v \in V$, we are given a (typically small) set $S(v) \subseteq S$ of station cells in the vicinity by which v's demand may be (partly) fulfilled. Furthermore, let $a_{v,s} \in (0, 1]$, $\forall v \in V,\ s \in S(v)$, be an attractiveness value indicating the expected proportion of demand from v (ingoing as well as outgoing) that can at most be fulfilled with a sufficiently sized station at s. These attractiveness values will be determined primarily based on the walking distances among the stations (the value will typically roughly exponentially decrease with the distance), but can be in general an arbitrary distance decay model. If there is a one-to-one correspondence of cells in V and S, for each $v \in V$, $v \in S(v)$, $a_{v,v} = 1$ will typically hold.

For the costs of building a station we consider here only a (strongly) simplified linear model, but we distinguish fixed costs for building the station and initially buying the bikes, variable costs for maintaining the station and the respective bikes, and costs for performing the rebalancing. Let b^{fix} and b^{var} be the average fixed and variable costs per bike slot, and let b^{reb} be the average costs for rebalancing one bike per day over the whole planning horizon. The fixed costs for a station in cell $s \in S$ with x_s slots are then $\mathrm{fixcost}(s) = b^{\mathrm{fix}} \cdot x_s$ and the total costs are $\mathrm{totalcost}(s) = b^{\mathrm{fix}} \cdot x_s + b^{\mathrm{var}} \cdot x_s + b^{\mathrm{reb}} \cdot Q_x(s)$, where $Q_x(s)$ denotes an estimation for the number of bikes that need to be redistributed from station s to some other station. We assume here that the size of each station, i.e., the number of its slots, can be freely chosen from 0 (i.e., no station is built) up to some maximum cell-dependent capacity $z_s \in \mathbb{N}$. The determination of the rebalancing effort for a given candidate solution will be described in Section 3.4. We remark that this cost model only is a first very rough estimate. Considering location dependent costs, costs for a station to be built that are independent of the number of slots, and a more restricted selection of station sizes is left for future research. We assume that a total budget $B^{\mathrm{tot}}_{\max}$ is given as well as a budget for only the sum of all fixed costs $B^{\mathrm{fix}}_{\max} < B^{\mathrm{tot}}_{\max}$, and both must not be exceeded in a feasible solution.

3.1 Solution Representation

A solution $x = \{x_s \in \mathbb{N} \mid s \in S\}$ assigns each station cell $s \in S$ an amount of parking slots to be built, possibly also 0 which would mean that no station is going to be built in cell s.

3.2 Objective

The goal is to maximize the expected total number of journeys in the system, i.e., the total demand that actually can be fulfilled at each day over all time periods, considering the available budgets $B^{\mathrm{tot}}_{\max}$ and $B^{\mathrm{fix}}_{\max}$.

Let $D(x,t)$ be the total demand fulfilled by solution x in time period $t \in T$, and let $Q_x(s)$ be the required rebalancing effort arising at each station $s \in S \mid x_s \neq 0$ in terms of the number of bikes to be moved to some other station. The calculation of these values will be considered separately in Sections 3.3 and 3.4. The BSSPP can then be stated as the following MIP.

$$\max \sum_{t \in T} D(x,t) \tag{3}$$

$$\sum_{s \in S} (b^{\mathrm{fix}} \cdot x_s + b^{\mathrm{var}} \cdot x_s + b^{\mathrm{reb}} \cdot Q_x(s)) \leq B^{\mathrm{tot}}_{\max} \tag{4}$$

$$\sum_{s \in S} b^{\mathrm{fix}} \cdot x_s \leq B^{\mathrm{fix}}_{\max} \tag{5}$$

$$x_s \in \{0, \ldots, z_s\} \qquad s \in S \tag{6}$$

Inequality (4) calculates the total costs over all stations and ensures that the total budget is not exceeded, while inequality (5) restricts only the fixed costs over all stations by the respective budget.

3.3 Calculation of Fulfilled Customer Demand

To determine the overall fulfilled demand for a specific, given solution x and a certain time slot $t \in T$, we first make the following local definitions. Let

$S' = \{s \in S \mid x_s \neq 0\}$ correspond to the set of cells where a station actually is located, $V' = \{v \in V \mid S(v) \cap S' \neq \emptyset\}$ be the set of customer cells whose demand can possibly (partly) be fulfilled as at least one station exists in the neighborhood. Moreover, let $C' = \{p \in C \mid V(p) \cap V' \neq \emptyset\}$ be the set of all nodes in the hierarchical clustering representing relevant customer cells, i.e., cells whose demand can possibly be fulfilled. The set $S'(v) = S(v) \cap V',\ \forall v \in V'$ refers to the existing stations that might fulfill part of v's demand, and $V'(p) = V(p) \cap V',\ \forall p \in C'$ denotes the existing customer cells contained in cluster p. $C'(p)$ refers to the subset of all the nodes $q \in C'$ that are part of the subtree rooted at p, including p and $V'(p)$, and $G' = (C', A')$ with $A' = \{(p,q) \in A^t \mid p,q \in C'\}$ is then the correspondingly reduced demand graph.

In the following we use variables u, v, w for referencing customer cells in V', variables p, q for referencing cluster nodes in C' (which might possibly also be customer cells), variable s for station cells in S', and α, β for arbitrary nodes in $C' \cup S$.

We further define for each arc in A' corresponding to a specific demand an individual flow network depending on the kind of the arc:

- Arcs $(u,v) \in A'$ with $u, v \in V'$, including the case $u = v$: $G_{\mathrm{f}}^{u,v} = (V_{\mathrm{f}}^{u,v}, A_{\mathrm{f}}^{u,v})$ with node set $V_{\mathrm{f}}^{u,v} = \{u\} \cup S'(u) \cup S'(v) \cup \{v\}$ and arc set $A_{\mathrm{f}}^{u,v} = (\{u\} \times S'(u)) \cup (S'(u) \times S'(v)) \cup (S'(v) \times \{v\})$.
- Arcs $(v,p) \in A'$ with $v \in V', p \in C' \setminus V'$: $G_{\mathrm{f}}^{v,p} = (V_{\mathrm{f}}^{v,p}, A_{\mathrm{f}}^{v,p})$ with node set $V_{\mathrm{f}}^{v,p} = \{v\} \cup S'(v) \cup \{p\}$ and arc set $A_{\mathrm{f}}^{v,p} = (\{v\} \times S'(v)) \cup (S'(v) \times \{p\})$.
- Arcs $(p,v) \in A'$ with $p \in C' \setminus V', v \in V'$: $G_{\mathrm{f}}^{p,v} = (V_{\mathrm{f}}^{p,v}, A_{\mathrm{f}}^{p,v})$ with node set $V_{\mathrm{f}}^{p,v} = \{p\} \cup S'(v) \cup \{v\}$ and arc set $A_{\mathrm{f}}^{p,v} = (\{p\} \times S'(v)) \cup (S'(v) \times \{v\})$.

All arcs $(\alpha, \beta) \in A_{\mathrm{f}}^{p,q}$ of all flow networks have associated corresponding flow variables $0 \leq f_{\alpha,\beta}^{p,q} \leq d_{p,q}^t$. The fulfilled demands can be modeled within these networks as maximum flows. Furthermore, we utilize variables $H_p^{\mathrm{in}},\ H_p^{\mathrm{out}}\ \forall p \in C' \setminus V'$, for the total inflow/outflow at all customer cells $V'(p)$ originating at/targeted to cluster nodes from outside cluster p, i.e., $C' \setminus C'(p) \setminus V'$. Variables $F_p^{\mathrm{in}}, F_p^{\mathrm{out}},\ \forall p \in C' \setminus V'$, represent the total ingoing/outgoing flows at all cluster nodes q within cluster p originating at/targeted to customer cells outside cluster p, i.e., $V' \setminus V'(p)$, respectively. The flow variables, however, depend on each other and the stations' capacities. A weighting factor ω is used to adjust the number of trips which can be performed in time period t by using only a single bike. The following LP is used to compute the total satisfied demand $D(x,t) =$

$$\max \quad \sum_{(v,p) \in A' \mid v \in V'} \ \sum_{(v,s) \in A_{\mathrm{f}}^{v,p}} f_{v,s}^{v,p} \tag{7}$$

$$\text{s.t.} \quad \sum_{(v,s) \in A_{\mathrm{f}}^{v,p}} f_{v,s}^{v,p} \leq d_{v,p}^t \qquad (v,p) \in A' \mid v \in V' \tag{8}$$

$$\sum_{(s,v) \in A_{\mathrm{f}}^{p,v}} f_{s,v}^{p,v} \leq d_{p,v}^t \qquad (p,v) \in A' \mid v \in V' \tag{9}$$

$$f_{u,s}^{u,v} = \sum_{s' \in S'(v)} f_{s,s'}^{u,v} \qquad (u,v) \in A' \mid u,v \in V',\ s \in S'(u) \quad (10)$$

$$\sum_{s' \in S'(u)} f_{s',s}^{u,v} = f_{s,v}^{u,v} \qquad (u,v) \in A' \mid u,v \in V',\ s \in S'(v) \quad (11)$$

$$f_{v,s}^{v,p} = f_{s,p}^{v,p} \qquad (v,p) \in A' \mid v \in V',\ p \in C' \setminus V', s \in S'(v) \quad (12)$$

$$f_{p,s}^{p,v} = f_{s,v}^{p,v} \qquad (p,v) \in A' \mid v \in V',\ p \in C' \setminus V', s \in S'(v) \quad (13)$$

$$\begin{aligned} -x_s \leq & \sum_{(p,q) \in A'} \sum_{(\alpha,s) \in A_{\mathrm{f}}^{p,q}} f_{\alpha,s}^{p,q} \\ & - \sum_{(p,q) \in A'} \sum_{(s,\alpha) \in A_{\mathrm{f}}^{p,q}} f_{s,\alpha}^{p,q} \\ & - \omega \cdot \frac{\delta^{\mathrm{rent}} \cdot \sum_{(p,q) \in A'} \sum_{(\alpha,s) \in A_{\mathrm{f}}^{p,q}} f_{\alpha,s}^{p,q}}{\delta_t^{\mathrm{period}}} \end{aligned} \qquad s \in S' \quad (14)$$

$$\begin{aligned} x_s \geq & \sum_{(p,q) \in A'} \sum_{(\alpha,s) \in A_{\mathrm{f}}^{p,q}} f_{\alpha,s}^{p,q} \\ & - \sum_{(p,q) \in A'} \sum_{(s,\alpha) \in A_{\mathrm{f}}^{p,q}} f_{s,\alpha}^{p,q} \\ & + \omega \cdot \frac{\delta^{\mathrm{rent}} \cdot \sum_{(p,q) \in A'} \sum_{(s,\alpha) \in A_{\mathrm{f}}^{p,q}} f_{s,\alpha}^{p,q}}{\delta_t^{\mathrm{period}}} \end{aligned} \qquad s \in S' \quad (15)$$

$$H_p^{\mathrm{in}} = \sum_{(q,v) \in A' \mid q \notin C'(p) \cup V', v \in V'(p)} \sum_{(s,q) \in A_{\mathrm{f}}^{q,v}} f_{s,q}^{q,v} \qquad p \in C' \setminus V' \quad (16)$$

$$F_p^{\mathrm{in}} = \sum_{(v,q) \in A' \mid v \notin V'(p), q \in C'(p) \setminus V'} \sum_{(s,q) \in A_{\mathrm{f}}^{v,q}} f_{s,q}^{v,p} \qquad p \in C' \setminus V' \quad (17)$$

$$H_p^{\mathrm{in}} \geq F_p^{\mathrm{in}} \qquad p \in C' \setminus V' \setminus \{0\} \quad (18)$$

$$H_0^{\mathrm{in}} = F_0^{\mathrm{in}} \quad (19)$$

$$H_p^{\mathrm{out}} = \sum_{(v,q) \in A' \mid v \in V'(p), q \notin C'(p) \cup V'} \sum_{(q,s) \in A_{\mathrm{f}}^{q,v}} f_{q,s}^{q,v} \qquad p \in C' \setminus V' \quad (20)$$

$$F_p^{\mathrm{out}} = \sum_{(q,v) \in A' \mid q \in C'(p) \setminus V', v \notin V'(p)} \sum_{(p,s) \in A_{\mathrm{f}}^{q,v}} f_{q,s}^{q,v} \qquad p \in C' \setminus V' \quad (21)$$

$$H_p^{\text{out}} \geq F_p^{\text{out}} \qquad p \in C' \setminus V' \setminus \{0\} \quad (22)$$

$$H_0^{\text{out}} = F_0^{\text{out}} \quad (23)$$

$$0 \leq f_{v,s}^{v,p} \leq a_{v,s} \cdot d_{v,p}^t \qquad (v,p) \in A' \mid v \in V',\ (v,s) \in A_{\mathrm{f}}^{v,p} \quad (24)$$

$$0 \leq f_{s,v}^{p,v} \leq a_{s,v} \cdot d_{p,v}^t \qquad (p,v) \in A' \mid v \in V',\ (s,v) \in A_{\mathrm{f}}^{p,v} \quad (25)$$

$$0 \leq f_{\alpha,\beta}^{p,q} \leq d_{p,q}^t \qquad (p,q) \in A',\ (\alpha,\beta) \in A_{\mathrm{f}}^{p,q} \mid \alpha,\beta \notin V' \quad (26)$$

$$F_p^{\text{in}}, F_p^{\text{out}} \geq 0 \qquad p \in C' \setminus V' \quad (27)$$

$$H_p^{\text{in}}, H_p^{\text{out}} \geq 0 \qquad p \in C' \setminus V' \quad (28)$$

Objective function (7) maximizes the total outgoing flow over all $v \in V'$, i.e., the fulfilled demand. Note that this also corresponds to the total ingoing flow over all v. Inequalities (8) limit the total flow leaving $v \in V'$, for each demand $(v,p) \in A' \mid v \in V'$ to $d_{v,p}^t$. Inequalities (9) do the same w.r.t. ingoing demands. Equalities (10) and (11) provide the flow conservation at source and destination stations s for $(u,v) \in A'$ with $u, v \in V'$. Equalities (12) provide the flow conservation at the source station in case of an arc $(v,p) \in A'$ towards a cluster node p, while (13) provide the flow conservation at the destination station in case of an arc $(p,v) \in A'$ originating at a cluster node p. Inequalities (14) and (15) provide the capacity limitations at each station $v \in V'$. It is the accumulated demand occurring at the particular station including a "compensation term" for large values of ingoing as well as outgoing demand. The fraction $\delta_t^{\text{period}}/\delta^{\text{rent}}$ represents the number of trips which can ideally be performed in period t using a single bike. The weighting factor ω is used to adjust this value such that it better reflects reality as the bike trips are not likely to be performed "optimally" with respect to the distribution over the whole time period in real world. Equalities (16) compute the total outgoing flow for the leafs of the subtree rooted at p to any cluster which is not part of the subtree rooted at p. Equalities (17) compute the total ingoing flow for each cluster node p by considering the ingoing flow from any $v \in V$ for which p is not a predecessor to every cluster of the subtree rooted at p. Inequalities (18) ensure that there must not be more ingoing flow to clusters of the subtree rooted at p as there is outgoing flow from the leafs contained in the subtree rooted at p. Equality (19) ensures that at the top level, i.e., at the root node 0, the outgoing flow from leaf nodes to cluster nodes and the ingoing flow from cluster nodes to leaf nodes is balanced, i.e, the same amount. Inequalities (21)–(23) state the corresponding constraints for the outgoing flow instead of the ingoing flow. Equations (24) and (25) provide the domain definitions for the flow variables from/to a cell v to/from a neighboring station s by considering the demand weighted by factor $a_{v,s}$. For all remaining flow variables, (26) provide the domain definitions based on the demands. The remaining variables are just restricted to be non-negative in (27) and (28).

3.4 Calculation of Rebalancing Costs

We state an LP for minimizing the total rebalancing effort over all time periods T at each station $s \in S'$ by choosing an appropriate initial fill level for each period, ensuring that the whole prospective customer demand is fulfilled. We estimate the rebalancing effort by considering the necessary changes in the fill levels inbetween the time periods. The LP uses the following decision variables. By $y_{t,s}$ we refer to the initial fill level of station $s \in S'$ at the beginning of time period $t \in T$, and by $r^+_{t,s}$ and $r^-_{t,s}$ we denote the number of bikes which need to be delivered to, respectively picked up from, station $s \in S'$ at the end of period $t \in T$ to achieve the fill levels $y_{t+1,s}$ (or $y_{1,s}$ in case of $t = \tau$).

The accumulated demand $D^{\text{acc}}_{t,v}$ can be calculated by utilizing the solution of the previous model from Section 3.3, c.f. inequalities (14) and (15). The following LP is solved for each station $s \in S' \mid x_s \neq 0$ independently. For station cells $s \in S \setminus S'$, i.e., where no station is actually built in solution x, $Q_x(s) = 0$.

$$Q_x(s) = \min \quad \sum_{t \in T} r^+_{t,s} + r^-_{t,s} \tag{29}$$

$$\text{s.t.} \quad y_{t,s} + r^+_{t,s} \geq D^{\text{acc}}_{t,s} \qquad t \in T \tag{30}$$

$$x_s - y_{t,s} + r^-_{t,s} \geq -D^{\text{acc}}_{t,s} \qquad t \in T \tag{31}$$

$$y_{t+1,s} = y_{t,s} - D^{\text{acc}}_{t,s} + r^+_{t,s} - r^-_{t,s} \qquad t \in T \setminus \{\tau\} \tag{32}$$

$$y_{1,s} = y_{\tau,s} - D^{\text{acc}}_{\tau,s} + r^+_{\tau,s} - r^-_{\tau,s} \tag{33}$$

$$0 \leq y_{t,s} \leq x_s \qquad t \in T \tag{34}$$

$$0 \leq r^+_{t,s} \leq D^{\text{acc}}_{t,s} \qquad t \in T \tag{35}$$

$$0 \leq r^-_{t,s} \leq -D^{\text{acc}}_{t,s} \qquad t \in T \tag{36}$$

Objective function (29) minimizes the number of rebalanced bikes, i.e., number of bikes that have to be delivered $r^+_{t,s}$ and number of bikes that have to be picked up $r^-_{t,s}$. Inequalities (30) compute the number of bikes that have to be delivered to the corresponding station in order to meet the given demand. Inequalities (31) compute the number of bikes that have to be picked up from the corresponding station in order to meet the given demand. Inequalities (32) state a recursion in order to compute the fill level for the next time period. Inequalities (33) state that for each station the fill level for the next day has to be again the initial fill level of the first period. Inequalities (34)–(36) are the domain definitions for the number of bikes to be moved and the fill level for each time period.

4 Multilevel Refinement Approach

Clearly, practical instances of the problem are far too large to be approached by a direct exact MIP approach. However, also basic constructive techniques or metaheuristics with simple, classical neighborhoods are unlikely to yield reasonable results when making decisions on a low level without considering crucial relationships on higher abstraction levels, i.e., a more global view. Classical local search techniques on the natural variable domains concerning decisions for individual stations may only fine-tune a solution but are hardly able to overcome bad solutions in which larger regions need to be either supplied with new stations or where many stations need to be removed. We therefore have the strong need of some technique that exploits also a higher-level view, deciding for larger areas about the supply of stations in principle. Multilevel refinement strategies can provide this point-of-view.

In multilevel refinement strategies [11] the whole problem is iteratively coarsened (aggregated) until a certain problem size is reached that can be reasonably handled by some exact or heuristic optimization technique. After obtaining a solution at this highest abstraction level, the solution is iteratively extended to the previous lower level problem instance and possibly refined by some local search, until a solution to the original problem at the lowest level, i.e., the original problem instance, is obtained. For a general discussion and the generic framework we refer to the work of Walshaw [10].

To apply multilevel refinement to BSSPP we essentially have to decide how to realize the procedures for coarsening an instance for the next higher level, solving a reasonably small instance, and extending a solution to a solution at the next lower level. In the following, we denote all problem instance data at level l by an additional superscript l. By P_l we generally refer to the problem at level l of the MLR algorithm described here.

4.1 Coarsening

We have to derive the more abstract problem instance P_{l+1} from a given instance P_l. Naturally, we can exploit the already existing customer cell cluster hierarchy for the coarsening. Remember that all customer cells appear in the cluster hierarchy always at the same level. We coarsen the problem by considering the customer cells and the station cells separately.

Coarsening of customer cells. The main strategy for coarsening the customer cells is to merge cells having the same parent cluster together with their parent. This means $V^{l+1} = C^l_{h^l-1}$ or simply $V^{l+1} = C_{h-l-1}$, i.e., each cluster node at depth $h - l - 1$ corresponds to a customer cell at level $l + 1$ representing the merged set of customer nodes contained in C_{h-l-1}. The hierarchical clustering of P_l becomes $C^{l+1} = C_0 \cup \ldots \cup C_{h-l}$. Remember that we already defined the function super(p) to return the parent cluster of some node p, and therefore super$(p^l) : C^l \to C^{l+1}$ also returns the cluster from C^{l+1} in which cluster $p^l \in C^l$ is merged into. The new demand graph $G^{t,l+1} = (C^{t,l+1}, A^{t,l+1})$ consists of the arc set $A^{t,l+1} = \bigcup_{(p^l,q^l)\in A^{t,l}} (\mathrm{super}(p^l), \mathrm{super}(q^l))$. This demand graph may again contain self-loops, but it is still simple, i.e., multiple arcs from $A^{t,l}$ may map to the same single arc in $A^{t,l+1}$ and the respective demand values are merged. Considering an arc $(p^{l+1}, q^{l+1}) \in A^{t,l+1}$, its associated demand is thus

$$d^{t,l+1}_{p^{l+1},q^{l+1}} = \sum_{(p^l,q^l)\in A^{t,l} \mid p^{l+1}=\mathrm{super}(p^l),\, q^{l+1}=\mathrm{super}(q^l)} d^{t,l}_{p^l,q^l}. \quad (37)$$

Note that the conditions for a valid demand graph and valid demand values stated in inequalities (1) and (2) will still hold when aggregating in this way, since the total ingoing and outgoing demand at each cluster $p \in C^{l+1}$ (including the demands from and to all existing subnodes) stays the same.

Coarsening of station cells. To coarsen the station cells we need to define a hierarchical clustering for them as well. For simplicity we assume from now on that $S = V$ holds, i.e., there is a one-to-one correspondence of considered station cells and customer cells. This also appears reasonable in a practical setting. We can then apply the hierarchical clustering defined for the customer cells also to the station cells. Maximum station capacities for aggregated stations $s^{l+1} \in S^{l+1}$ are naturally calculated by the sum of the respective maximum capacities of the underlying station cells, i.e., $z^{l+1}_{s^{l+1}} = \sum_{s^l \in \mathrm{sub}(s^{l+1})} z^l_{s^l}$.

Coarsening of neighborhoods. A coarsened neighborhood mapping $S^{l+1}(v^{l+1})$ for each customer cell $v^{l+1} \in V^{l+1}$ and respective attractiveness values $a_{v^{l+1},s^{l+1}}$ for station cells $s^{l+1} \in S^{l+1}(v^{l+1})$ are determined as follows. The neighborhood mapping is retained as long as the attractiveness value in the coarsened problem instance does not fall below a certain threshold $\lambda \in (0, 1)$:

$$S^{l+1}(v^{l+1}) = \left\{ s^{l+1} \in \bigcup_{v^l \in \mathrm{sub}(v^{l+1})} \mathrm{super}(S^l(v^l)) \mid a_{v^{l+1},s^{l+1}} \geq \lambda \right\} \tag{38}$$

with the aggregated attractiveness values being

$$a_{v^{l+1},s^{l+1}} = \begin{cases} 1 & \text{if } v^{l+1} = s^{l+1} \\ \frac{\sum_{v^l \in \mathrm{sub}(v^{l+1})} \sum_{s^l \in \mathrm{sub}(s^{l+1}) \cap S^l(v^l)} \left(a_{v^l,s^l}\right)}{|\mathrm{sub}(v^{l+1})| \cdot |\mathrm{sub}(s^{l+1})|} & \text{if } v^{l+1} \neq s^{l+1}. \end{cases} \tag{39}$$

4.2 Initialization

The initial problem becomes coarsened until we reach some level l where it can be reasonably solved as it is then small enough. In our experiments with binary clustering trees here we are stopping the coarsening when the clustering tree has no more than $2^5 = 32$ leaf nodes, or in other words, at a height of five. For initializing the solution at the coarsest level we utilize a MIP model. In this model, the objective stated in Section 3.2, the demand calculation for every time period stated in Section 3.3, and the rebalancing LP model stated in Section 3.4 are put together. By solving this model we obtain an optimal solution for the coarsest level, which forms the basis for proceeding with the next step of the algorithm, the *extension* to derive step-by-step a more detailed solutions.

4.3 Extension

In the extension step we derive from a solution x^{l+1} at level $l + 1$ a solution x^l at level l, i.e., we have to decide for each aggregated station $s^{l+1} \in S^{l+1}$ with $x^{l+1}_{s^{l+1}} > 0$ slots how they should be realized by the respective underlying station cells $\mathrm{sub}(s^{l+1})$ at level l. We do this in a way so that the globally fulfilled demand is again maximized by solving the following MIP.

$$\max \quad \sum_{t \in T} D(x^l, t) \tag{40}$$

$$\text{s.t.} \quad \sum_{s^l \in S^l} \left(b^{\mathrm{fix}} \cdot x^l_{s^l} + b^{\mathrm{var}} \cdot x^l_{s^l} + b^{\mathrm{reb}} \cdot Q_{x^l}(s^l) \right) \leq B^{\mathrm{tot}}_{\max} \tag{41}$$

$$\sum_{s^l \in S^l} b^{\mathrm{fix}} \cdot x_{s^l} \leq B^{\mathrm{fix}}_{\max} \tag{42}$$

$$\sum_{s^l \in \mathrm{sub}(s^{l+1})} x^l_{s^l} \leq x^{l+1}_{s^{l+1}} \qquad s^{l+1} \in S^{l+1} \tag{43}$$

$$x^l_{s^l} \in \{0, \dots, z^l_s\} \qquad s^l \in S^l \tag{44}$$

The objective (40) maximizes the total satisfiable demand. Inequalities (41) restrict the maximum total budget whereas inequalities (42) restrict the maximum fixed budget. Inequalities (43) are the bounds on the total number of slots for the station nodes $s^l \in \mathrm{sub}(s^{l+1})$. The number of parking slots in each cell $x^l_{s^l}$ is restricted by the maximum number of parking slots allowed in this cell (44).

5 Computational Results

For our experiments we created seven different benchmark sets[1], each one containing 20 different, random instances. We consider instances with 200, 300, 500, 800, 1000, 1500, and 2000 customer cells, where each customer cell is also a possible location for a station to be built. Customer cells are aligned on a grid in the plane and euclidean distances have been calculated based on which a hierarchical clustering with the complete-linkage method was computed. Demands among the leaf nodes were chosen randomly, considering the pairwise distance between customer cells, and demands below a certain threshold have been aggregated upwards in the clustering tree such that the demand graphs get sparser. Only cells within 200 meters walking distance are considered to be in the vicinity of a customer cell and respective attractiveness values are chosen randomly but in correlation with the distances. We set the maximum station size to $z_s = 40$ for all cells in all test cases. For slot costs we set $b^{\mathrm{fix}} = 1750$ €, and $b^{\mathrm{var}} = 1000$ €, which are reasonable estimates in the Vienna area gathered from real BSSs. The costs for rebalancing a single bike for one day have been estimated with 3 € per bike and per day. When projecting this cost to the optimization horizon, e.g., 1 year, we get $b^{\mathrm{reb}} = 365 \cdot 3 = 1095$ €. For coarsening of attractiveness values, we set the corresponding parameter $\lambda = 0$ and for adjusting the number of trips which can be performed in a particular time period $t \in T$ by using only a single bike we set $\omega = 1.2$. Each instance contains four time periods which we selected as follows: 4:30am to 8:00am, 8:00am to 12:00 Noon, 12:00 Noon to 6:15pm, and 6:15pm to 4:30am. The duration for each time period $t \in T$ has been set accordingly and the average trip duration has been set to $t^{\mathrm{rent}} = 10$ minutes.

[1] `https://www.ac.tuwien.ac.at/files/resources/instances/bsspp/lion17.bz2`

Table 1. Results for the multilevel refinement heuristic (MLR).

Instance				MLR				
name	#runs	$B_{\max}^{\mathrm{tot}}$ [€]	$B_{\max}^{\mathrm{fix}}$ [€]	$\overline{\mathrm{obj}}$	$\overline{\#\mathrm{coarsen}}$	$\widetilde{\mathrm{time}}$ [s]	$\overline{\mathrm{totcost}}$ [€]	$\overline{\mathrm{fixcost}}$ [€]
BSSPP_200	20	200,000.00	130,000.00	9,651.98	3	46.2	198,000.00	126,000.00
BSSPP_300	20	350,000.00	250,000.00	10,951.79	5	60.8	349,250.00	222,250.00
BSSPP_500	20	500,000.00	350,000.00	16,057.78	6	121.6	497,750.00	316,750.00
BSSPP_800	20	850,000.00	550,000.00	28,862.21	6	263.9	849,750.00	540,750.00
BSSPP_1000	20	1,000,000.00	700,000.00	28,967.58	8	346.7	998,250.00	635,250.00
BSSPP_1500	20	1,500,000.00	1,000,000.00	41,208.19	8	574.5	1,498,475.00	953,575.00
BSSPP_2000	20	2,000,000.00	1,300,000.00	55,892.06	8	803.4	1,999,250.00	1,272,250.00
			Average	27,370.22	6.3		912,960.71	580,975.00

All algorithms are implemented in C++ and have been compiled with gcc 4.8. For solving the LPs and MIPs we used Gurobi 7.0. All experiments were executed as single threads on an Intel Xeon E5540 2.53GHz Quad Core processor.

Table 1 summarizes obtained results. For every instance set we state the name containing the number of nodes, the number of different instances we have tested on (#runs), the maximum total budget ($B_{\max}^{\mathrm{tot}}$), and the maximum fixed budget ($B_{\max}^{\mathrm{fix}}$). For the proposed MLR, we list the average objective value ($\overline{\mathrm{obj}}$), i.e., the expected fulfilled demand in terms of the number of journeys, the average number of coarsening levels ($\overline{\#\mathrm{coarsen}}$), the median time ($\widetilde{\mathrm{time}}$), and the average total costs ($\overline{\mathrm{totcost}}$) as well as the average fixed costs ($\overline{\mathrm{fixcost}}$) for building the number of slots in the solution. Most importantly, it can be seen that the proposed MLR scales very well to large instances up to 2000 customer cells.

6 Conclusion and Future Work

We presented an innovative approach to the BSSPP. Previous work only considers very small instances and case studies to small parts of a city whereas we aim at solving more realistic large-scale scenarios arising in large cities. As we have to cope with thousands of customer cells and potential station cells it is most fundamental to model the potential demands efficiently. To this end, we proposed to use a hierarchical clustering and defining the demand graph on it. This approach can drastically reduce the data in comparison to a complete demand matrix with only a very reasonable information loss. Moreover, we provided MIP formulations to compute the satisfiable demand by given configurations and to compute the prospective rebalancing costs. Putting them together under the objective of maximizing the expected satisfied total demand and adding further constraints for complying with given monetary budget constraints, we obtained a MIP model that solves our definition of the BSSPP exactly. Because this MIP model can in practice still only be solved for rather small instances, we further suggested a multilevel refinement heuristic utilizing the same hierarchical clustering we are given as input. Using this approach we have shown to be able to solve instances with up to 2000 nodes in reasonable computation times.

In future work it is important to make the cost model more realistic and to test on more realistic benchmark instances. In particular, we aim at considering also fixed costs for building a station which are independent of the number of slots. Furthermore, in practice also only a small, restricted set of different station configurations is possible per station cell. These extensions introduce interesting research questions especially in relation to the multilevel refinement procedure.

Acknowledgements

We thank the LOGISTIKUM Steyr, the Austrian Institute of Technology, and Rosinak & Partner for the collaboration on this topic. This work is supported by the Austrian Research Promotion Agency (FFG) under contract 849028.

References

1. Chen, J., Chen, X., Jiang, H., Zhu, S., Li, X., Li, Z.: Determining the optimal layout design for public bicycle system within the attractive scope of a metro station. Math. Probl. Eng. 2015, 8 pages (2015), Article ID 456013
2. Chen, Q., Sun, T.: A model for the layout of bike stations in public bike-sharing systems. J. Adv. Transport. 49(8), 884–900 (2015)
3. Frade, I., Ribeiro, A.: Bike-sharing stations: A maximal covering location approach. Transport. Res. A-Pol. 82, 216–227 (2015)
4. Gavalas, D., Konstantopoulos, C., Pantziou, G.: Design & management of vehicle sharing systems: A survey of algorithmic approaches. In: Obaidat, M.S., Nicopolitidis, P. (eds.) Smart Cities and Homes: Key Enabling Technologies, chap. 13, pp. 261–289. Elsevier Science (2016)
5. Hu, S.R., Liu, C.T.: An optimal location model for a bicycle sharing program with truck dispatching consideration. In: IEEE 17th International Conference on Intelligent Transportation Systems (ITSC). pp. 1775–1780. IEEE (2014)
6. Lin, J.R., Yang, T.H.: Strategic design of public bicycle sharing systems with service level constraints. Transport. Res. E-Log. 47(2), 284–294 (2011)
7. Lin, J.R., Yang, T.H., Chang, Y.C.: A hub location inventory model for bicycle sharing system design: Formulation and solution. Comput. Ind. Eng. 65(1), 77–86 (2013)
8. Martinez, L.M., Caetano, L., Eiró, T., Cruz, F.: An optimisation algorithm to establish the location of stations of a mixed fleet biking system: An application to the city of Lisbon. Procedia Soc. Behav. Sci. 54, 513–524 (2012)
9. Saharidis, G., Fragkogios, A., Zygouri, E.: A multi-periodic optimization modeling approach for the establishment of a bike sharing network: A case study of the city of Athens. In: Proceedings of the International MultiConference of Engineers and Computer Scientists 2014 Vol II. pp. 1226–1231. No. 2210 in LNECS, Newswood Limited (2014)
10. Walshaw, C.: A multilevel approach to the travelling salesman problem. Oper. Res. 50(5), 862–877 (2002)
11. Walshaw, C.: Multilevel refinement for combinatorial optimisation problems. Ann. Oper. Res. 131(1), 325–372 (2004)
12. Yang, T.H., Lin, J.R., Chang, Y.C.: Strategic design of public bicycle sharing systems incorporating with bicycle stocks considerations. In: 40th International Conference on Computers and Industrial Engineering (CIE). pp. 1–6. IEEE (2010)