EURO J Transp Logist
DOI 10.1007/s13676-012-0017-6

RESEARCH PAPER

Static repositioning in a bike-sharing system: models and solution approaches

Tal Raviv · Michal Tzur · Iris A. Forma

Received: 2 May 2012 / Accepted: 12 December 2012

Abstract Bike-sharing systems allow people to rent a bicycle at one of many automatic rental stations scattered around the city, use them for a short journey and return them at any station in the city. A crucial factor for the success of a bike-sharing system is its ability to meet the fluctuating demand for bicycles and for vacant lockers at each station. This is achieved by means of a *repositioning* operation, which consists of removing bicycles from some stations and transferring them to other stations, using a dedicated fleet of trucks. Operating such a fleet in a large bike-sharing system is an intricate problem consisting of decisions regarding the routes that the vehicles should follow and the number of bicycles that should be removed or placed at each station on each visit of the vehicles. In this paper, we present our modeling approach to the problem that generalizes existing routing models in the literature. This is done by introducing a unique convex objective function as well as time-related considerations. We present two mixed integer linear program formulations, discuss the assumptions associated with each, strengthen them by several valid inequalities and dominance rules, and compare their performances through an extensive numerical study. The results indicate that one of the formulations is very effective in obtaining high quality solutions to real life instances of the problem consisting of up to 104 stations and two vehicles. Finally, we draw insights on the characteristics of good solutions.

T. Raviv (✉) · M. Tzur · I. A. Forma
Industrial Engineering Department, Tel Aviv University, 69978 Tel Aviv, Israel
e-mail: talraviv@eng.tau.ac.il

M. Tzur
e-mail: tzur@eng.tau.ac.il

I. A. Forma
e-mail: irisforma@eng.tau.ac.il; Irisf@afeka.ac.il

I. A. Forma
Afeka Tel Aviv Academic College of Engineering, Bnei Efraim 218, Tel Aviv, Israel

Published online: 08 January 2013

Keywords Bike-sharing systems · Static repositioning · Pickup and delivery problem

Introduction and problem definition

Bike-sharing systems allow people to rent a bicycle at one of the many automatic rental stations scattered around the city, use them for a short journey and return them at any station in the city. Recently many cities around the world established such systems in order to encourage their citizens to use bicycles as an environmentally sustainable and socially equitable mode of transportation, and as a good complement to other modes of mass transit systems (mode-sharing).

A rental station typically includes one terminal and several bicycle stands. The terminal is a device capable of communicating with the electronic lockers, which are attached to the bicycle stands. When a user rents a bicycle, a signal is sent to the terminal that the locker has been vacated. A user can return a bicycle to a station only when there is a vacant locker. All rental and return transactions are recorded and reported in real time to a central control facility. Thus, the state of the system, in terms of the number of bicycles and number of vacant lockers available at each station, is known to the operator in real time. Moreover, operators of bike-sharing systems make this information available to the users online.

A crucial factor in the success of a bike-sharing system is its ability to meet the fluctuating demand for bicycles at each station. In addition, the system should be able to provide enough vacant lockers to allow the users to return the bicycles at their destinations. Indeed, one of the main complaints heard from users of bike-sharing systems relates to unavailability of bicycles and (even worse) unavailability of lockers at their destination, see, e.g., Shaheen and Guzman (2011) and media reports Brussel Nieuws (2010) and Tusia-Cohen (2012). Persistent unavailability of bicycles and/or lockers engenders distrust among the system's users and could eventually lead them to abandon it.

We measure user dissatisfaction with the system through the expected number of shortage events. Such an event occurs when a user who wishes to rent a bicycle arrives at an empty station or a user who wishes to return a bicycle arrives at a full station, with no vacant lockers. In order to reduce shortages, operators of bike-sharing systems are responsible to regularly remove bicycles from some stations and transfer them to other stations, using a dedicated fleet of light trucks. We refer to this activity as *repositioning* of bicycles. The goals of the operators are to minimize the number of shortages incurred in the system and the fleet's operational costs.

Consequently, repositioning of bicycles in the system involves routing decisions concerning the vehicles, starting from and returning to the depot, and inventory decisions concerning bicycles in the rental stations. The latter involves determining the number of bicycles to be removed or placed in each station on each visit of the vehicles. Ideally, the outcome of this operation would be to meet all demand for bicycles and vacant lockers, but this may not be possible due to demand uncertainty, capacity constraints of the vehicles and the stations, and the inherent imbalances in the renting and return rates at the various stations. The imbalance is sometimes

temporary, e.g., high return rate in a suburban train station during the morning and a high renting rate during the afternoon, or persistent, e.g., relatively low return rates in stations located on top of hills. Therefore, a new modeling approach is required; one that will correctly represent a practical objective of the repositioning operation, related to the users' satisfaction with the system. Toward that we define a penalty function, which represents the expected number of shortages at any inventory level at each station. The use of such an objective, combined with new model characteristics, is the essence of this paper.

The repositioning operation can be carried out in two different modes: one is during the night when the usage rate of the system is negligible; the other is during the day when the status of the system is rapidly changing. We refer to the former as the *static bicycle repositioning problem* (SBRP) and to the latter as the *dynamic bicycle repositioning problem* (DBRP). Some operators use static repositioning, some dynamic, and some use a combination of the two, Callé (2009).

In this paper, we focus on the static mode of operation which benefits from a practical advantage because it allows the repositioning fleet to travel swiftly in the city without contributing to traffic congestion and parking problems. Static repositioning is useful for arranging the inventory of bicycles in the system toward the next day. When combined with dynamic repositioning, it reduces the amount of work required in the latter mode. The static problem needs to be solved once at the beginning of every night, based on the status of the system at that time and the demand forecast for the next day. While the problem lies in the general domain of vehicle routing problems, it involves some unique characteristics that require appropriate modeling. The model we present in this paper generalizes existing routing models from the literature, as shown in the next section.

We formally define the SBRP as follows. The input of the problem is a set of stations, with one designated as the depot; initial inventory, capacity, and a convex penalty function for each station; a travel-time matrix between stations; and a set of non-identical capacitated repositioning vehicles. A solution is defined by a route for each vehicle and the quantity of bicycles to load or unload at each station along this route. The planned routes must satisfy a time constraint. The length of each route is calculated as the sum of the travel times between all pairs of consecutive stations along the route plus the time needed for loading and unloading at the stations, which depends on the quantities of bicycles handled. The goal is to minimize a weighted objective that consists of the sum of the stations' penalty costs and the total operating costs of the vehicles. This problem definition generalizes our preliminary work, see Forma et al. (2010).

The penalty function at each station may represent any objective of the operator, as long as it is convex in the ending inventory of the station. We advocate the usage of penalty functions that represent the expected number of shortages in a station during the next day given its initial inventory, its capacity, and the stochastic characteristics of the demand process. Raviv and Kolka (2012) present an efficient procedure to calculate these functions and prove their convexity. For the sake of completeness, we summarized the relevant results from their paper in Appendix A. For more details, the reader is referred to that paper.

The contribution of this paper is in presenting a new and practical approach to modeling the SBRP. It involves defining a non-traditional objective function, which

is related to the satisfaction of users in the system. It also includes other new characteristics such as loading and unloading times within a time constrained setting. Based on this approach we present two mixed integer linear program (MILP) formulations, which differ from each other in their modeling choices and underlying assumptions. On the methodological side, the above formulations are strengthened by some valid inequalities and dominance rules that are likely to be useful in other routing problems, especially those that are generalized by this work. Finally, we applied our methods on a variety of large instances based on real data and achieved small optimality gaps within a reasonable time.

The rest of this paper is organized as follows: in "Literature review", we review the literature, describe related work from several application areas and identify the gaps that should be addressed. In "Model formulation", we present our modeling approach by specifying the underlying assumptions, elaborating on the chosen objective function, and presenting our mathematical formulations. In "Algorithmic enhancements", we discuss algorithmic enhancements that are useful in solving the mathematical models effectively. In "Numerical experiments", we describe our numerical experiments, the results, and their analysis. Finally, in "Conclusions and discussion", we discuss possible extensions and directions for further research.

Literature review

In this section, we first review recent studies on bike-sharing systems and in particular on the repositioning operation in those systems. Then we discuss the relation of the repositioning problem to classical routing and inventory/routing problems from the literature.

Modern bike-sharing systems have become prevalent only in the last few years; therefore, the existing literature analyzing these systems is relatively new. There are various interesting research questions concerning the establishment, operation and analysis of bike-sharing systems. Indeed, some works study strategic problems, such as Shu et al. (2010) and Lin and Yang (2011) who address the question of bike rental stations' capacity and locations. Others present empirical analysis, e.g., DeMaio (2009) and Hampshire and Marla (2011). Fricker and Gast (2012) study the system's behavior and the effect of various load-balancing strategies on their performances. They conclude that in asymmetric systems, repositioning of bicycles by trucks is necessary even when an incentive mechanism to self-balance it is put in place. Vogel and Mattfeld (2010) present a stylized model to assess the effect of dynamic repositioning efforts on service levels. Their model is useful for strategic planning but is not detailed enough to support repositioning operations.

Several approaches to modeling and optimizing the repositioning problem have recently been developed. There are essential differences between them in the underlying assumptions concerning the perceived system's behavior and the problem's objective, as we discuss next. Benchimol et al. (2011) and Chemla et al. (2011) address the static repositioning problem, assuming that a given target inventory level exists for each station in the system so that the objective of the repositioning operation is to achieve this target at minimum travel cost. The method

by which the target levels are obtained is not specified. Since no deviation from the target level is allowed, no time constraint is imposed on the repositioning operation (otherwise, it may not be feasible). A result of these assumptions is that the problem resembles the deterministic C-delivery TSP (Chalasani and Motwani 1999) or a single vehicle, single commodity pickup and delivery problem (PDP), discussed below. Erdoğan et al. (2012) extend the above studies by allowing the final inventory at each station to be within a pre-specified interval instead of at a given target value.

The models presented in this paper generalize the above-mentioned studies since we do not specify a target inventory level or interval, but rather allow reaching any inventory level and use a convex penalty function to express its cost. Indeed, the SBRP can be cast as the problem presented in Erdoğan et al. (2012) (resp., Chemla et al. (2011)) by selecting penalty functions that assign a value of zero for each final inventory level inside the desired interval (resp., at the target level) and a very large number for each inventory level outside it (resp., different from it).

Contardo et al. (2012) present a mathematical programming formulation for the dynamic repositioning problem (DBRP) and use decomposition schemes to obtain lower bounds and feasible solutions. Although their formulation bears some similarity to our time-indexed formulation, see "Time-indexed formulation", there are two important differences: first, their setting is deterministic and does not take into consideration the stochastic nature of the demands for bicycles and for lockers while in our model this stochasticity can be expressed by the convex penalty function. Second, our model includes loading and unloading times, which are proportional to the number of bicycles loaded/unloaded, whereas Contardo et al. (2012) do not refer to these times and it is unclear whether and how they can be incorporated in their formulation or in their solution method. In practice, loading and unloading times comprise a major portion of the repositioning time; hence, their inclusion is crucial for a correct representation of the problem and for obtaining good repositioning plans.

Nair and Miller-Hooks (2011) use a stochastic programming approach to perform repositioning in shared mobility systems. Their model assumes that the cost of moving objects (bicycles or cars) between two given stations is known and represented by a fixed plus-linear function, so that no routing constraints and costs are considered. This assumption may be realistic for the one-way car-sharing systems that motivated their work, but too simplistic for the bicycle repositioning problem addressed here. Moreover, their model calculates vehicle shortages based on the *net* demand during some planning period, which does not take into consideration the *dynamics* of the demand during that period. Such calculations may be valid only when a relatively short planning period is considered.

Raviv and Kolka (2012) show how to calculate the expected number of shortages as a function of the inventory level at the beginning of the day, which can be used in our model to represent the dynamics of the demand. Their method requires as input the rates of two independent non-homogenous Poisson demand streams for users seeking to rent bicycles and users seeking to return bicycles at a single station, during some planning period, e.g., during the next day. Although their model considers only a single station system, they demonstrate through simulation that

their results are robust to the inherent interactions between stations in a large real system.

There are some similarities between bike-sharing systems and car-sharing systems. The latter may belong to the round-trip, or to the one-way type of systems, based on whether the users have to return the cars to the same parking space or can return them to any station. For the former see, e.g., Mukai and Watanabe (2005) and for the latter see Uesugi et al. (2007). However, there are major differences between bike- and car-sharing systems, since typically a car is moved from one location to another by assigning a driver to it, while bicycles are moved in batches by repositioning vehicles.

Repositioning problems bear similarities to other routing problems from the literature. The basic operation performed during repositioning is that of pickup and delivery of identical items (bicycles). Berbeglia et al. (2007) survey the literature on static PDPs and classify them according to various parameters. By this classification, the repositioning problem presented in this study is most similar to a many-to-many single commodity PDP with a non-linear objective function, for which no studies are available. Moreover, in PDPs, the quantities picked-up or delivered to a node are given, as opposed to our SBRP where they are decision variables.

The studied problem which is most similar to the SBRP is the one-commodity pickup and delivery traveling salesman problem (1-PDTSP), introduced by Hernández-Pérez and Salazar-González (2004a). The 1-PDTSP is a generalization of the well-known TSP where each customer has supply or demand of a given amount of a single product. They present an ILP model, and describe a branch-and-cut procedure for solving it. Hernández-Pérez and Salazar-González (2004b) present heuristic methods for the problem and demonstrate their applicability for instances with up to 500 nodes. Louveaux and Salazar-González (2009) consider the 1-PDTSP with stochastic demand or supply. They study the problem of finding the smallest vehicle capacity that assures feasibility, i.e., being able to satisfy all demands; for a given vehicle's capacity, they search for a tour that minimizes the objective function, which includes a penalty that is proportional to the unsatisfied demand.

Our model for the SBRP extends the 1-PDTSP in the following four ways: (a) the number of vehicles that perform the task may be greater than one; (b) the number of items picked-up or delivered to each customer is a decision variable with an associated convex term in the objective function, rather than a parameter; (c) the total travel time allotted to the task is restricted; (d) a time is associated with the loading and unloading of items (bicycles). Note that the SBRP also differs from the 1-PDTSP by the fact that the stations may be visited by the same vehicle more than once. However, in order to simplify the problem, one of our formulations excludes such solutions.

Bicycle repositioning problems also bear similarities to the Inventory Routing Problem (IRP), where one needs to determine the routes of the vehicles, as well as the quantities of items to load or unload at each visited node. However, in the IRP, as in most inventory models, demand only depletes the available inventory. Returns, which increase the available inventory at a facility, either do not occur at all, or represent a significantly lower volume in the system. For a recent survey on IRP, see Bertazzi et al. (2008).

Finally, another closely related routing problem is the Swapping Problem (SP), first introduced by Anily and Hassin (1992). The goal in the SP is to compute a shortest route such that a vehicle of a single unit capacity can rearrange objects of known types from nodes where they are initially located to nodes where they are requested. Anily and Hassin (1992) show that the problem is NP-Hard and present a 2.5 approximation algorithm. The SP is more general than our problem in that objects may belong to more than one type; however, in the SP the demand at the nodes is limited to one unit, and there is only one vehicle with a capacity limited to one. Further studies on the SP focus mainly on special cases of the problem for which a polynomial time optimization or approximation algorithms are possible, e.g., see Anily et al. (1999). A recent study by Bordenave et al. (2010) presents a constructive approach and several improvement heuristics that provide near optimal solutions (optimality gap of less than 1 % on average) of instances with up to 10,000 nodes.

To summarize, this paper closes a significant gap in the routing literature, by introducing and solving a model whose goal is to minimize a convex function over the final inventory at the nodes, rather than merely minimize linear traveling/shortage/holding costs (or any combination of these cost components). The convexity of the objective function allows dealing directly with the stochastic nature of the demand at the nodes even though the model is deterministic. While the model presented here is customized for the SBRP, we hope that it will open a new line of research on problems where routing and inventory decisions should be made simultaneously in a stochastic setting.

Model formulation

As mentioned in the "Introduction", our objective function includes a term, which represents the users' satisfaction with the system (the sum of the penalty functions of all the stations), and a term that represents the operating costs that are proportional to the total distance travelled by the repositioning vehicles. The weighted sum of the two components is the total cost of the problem. An important special case of the above bi-objective function is when the weight of the operating costs is zero, expressing a situation in which the marginal operating costs are negligible relative to the importance of the service quality provided to the users.

The SBRP is to determine the route of each vehicle and the number of bicycles to load or unload at each station visited by the vehicles, such that the total penalty and operational costs are minimized. We assume the existence of a depot as the starting and ending point of each vehicle's route. However, we note that the depot may be viewed as a regular station, except that it typically has a relatively large capacity, large initial inventory and no demand. The above decisions are subject to capacity constraints of the vehicles, the stations and the depot, as well as time constraints concerning the total traveling, loading and unloading times. The latter two components are assumed to be linear in the number of bicycles loaded/unloaded.

Additional modeling choices are associated with the permissible actions of the fleet of vehicles performing the repositioning operation, in particular, limitations on

the allowed set of routes and/or limitations on the quantity of bicycles loaded/unloaded at each station. Such limitations may result from operational considerations or from computational limitations. We describe and discuss these limitations in detail later in this section, when presenting our mathematical formulations. Table 1 at the end of this section summarizes the capabilities and assumptions of the formulations presented.

In the static version of the problem studied here, repositioning is performed while the demand for bicycles and vacant lockers is assumed to be zero. Indeed, in reality the demand during the night is negligible. A given length of time (say, 5 h, from 1 am to 6 am every night) is allotted to the repositioning operation, and its purpose is to improve the starting conditions (bicycle initial inventory levels at the stations) of the next day.

Next, we present notation that is common to all our formulations.

Notation

The static repositioning problem is described by the following sets and parameters:

N	Set of stations, indexed by $i = 1, \ldots, \|N\|$
N_0	Set of nodes, including the stations and the depot (denoted by $i = 0$), $i = 0, \ldots, \|N\|$
V	Set of vehicles, $v = 1, \ldots, \|V\|$
s_i^0	Number of bicycles at node i before the repositioning operation starts
c_i	Number of lockers installed at station $i \in N_0$, referred to as the station's capacity
k_v	Capacity (number of bicycles) of vehicle $v \in V$
$f_i(s_i)$	A convex penalty function for station $i \in N$, the function is defined over the integers $s_i = 0, \ldots, c_i$
t_{ij}	Traveling time from station i to station j
α	Weight/scaling factor (in the objective function) of the operating costs relative to the penalty costs
T	Repositioning time, i.e., time allotted to the repositioning operation
L	Time required to remove a bicycle from a station and load it onto the vehicle
U	Time required to unload a bicycle from the vehicle and hook it to a locker in a station

To further clarify the above notation recall that static repositioning occurs during the night, and it starts when the quantity of bicycles at node i is s_i^0. Due to the repositioning operation, a different inventory level, s_i, is reached at the end of it, which determines the penalty cost. The penalty functions, $f_i(s_i)$, are calculated in a pre-processing step for every possible inventory level, i.e., from zero to the maximum capacity of each station i.

We suggest penalty functions $f_i(s_i)$ that represent the expected number of shortages during the next working day. Note that while the model is a deterministic mixed integer program, the stochastic nature of the demand is already incorporated via the above definition of the $f_i(\cdot)$ functions. The shortages may be weighted

according to whether they are incurred due to a lack of bicycles or a lack of vacant lockers, since each type of shortage represents a different level of users' dissatisfaction. This is a consideration, which can be incorporated in the pre-processing step, when employing the procedure of Raviv and Kolka (2012) and is part of our model's input.

Finally, note that the parameters α, L and U may be vehicle dependent (i.e., α_v, L_v and U_v) with no change in the formulations below. Such generalization may represent a non-homogenous fleet of vehicles, where larger vehicles, e.g., incur higher operating costs and/or different loading and unloading times. For simplicity, we omit this generalized notation.

Arc-indexed formulation

Our first mathematical formulation of the problem is referred to as the *arc-indexed* (AI) formulation. Initially, the first term of the objective function is represented by a sum of (discrete) convex functions, while later these functions are linearized *in an exact manner* through a set of constraints. In this formulation, we assume that each station may be visited at most once by each vehicle, although a certain station may be visited by several vehicles. This assumption reduces the feasible solution set and may result in inferior solutions. However, it simplifies the problem significantly so that relatively large instances can be solved by a commercial solver. We discuss the implications of this assumption in the numerical study section. Let us define the following decision variables:

- x_{ijv} Binary variable which equals one if vehicle v travels directly from node i to node j, and zero otherwise
- y_{ijv} Number of bicycles carried on vehicle v when it travels directly from node i to node j. y_{ijv} is zero if the vehicle v does not travel directly from i to j
- y^L_{iv} Number of bicycles loaded onto vehicle v at node i
- y^U_{iv} Number of bicycles unloaded from vehicle v at node i
- q_{iv} Auxiliary variables used for sub-tour elimination constraints
- s_i Inventory level at node i at the end of the repositioning operation

The following parameter needs to be defined and used specifically in this formulation:

- M An upper bound on the number of arcs in a vehicle's tour whose length is at most T time units, where the vehicle visits each station at most once ($M = |N_0|$ is a trivial such upper bound; it may be strengthened when T is small, by solving a simple integer program).

We are ready to present the AI formulation,
(P1)—Arc-indexed (AI) formulation.

$$Min \sum_{i \in N} f_i(s_i) + \alpha \sum_{i \in N_0} \sum_{j \in N_0} \sum_{v \in V} t_{ij} x_{ijv} \tag{1}$$

s.t.

$$s_i = s_i^0 - \sum_{v \in V} (y_{iv}^L - y_{iv}^U) \quad \forall i \in N_0 \tag{2}$$

$$y_{iv}^L - y_{iv}^U = \sum_{j \in N_0, j \neq i} y_{ijv} - \sum_{j \in N_0, j \neq i} y_{jiv} \quad \forall i \in N_0, \forall v \in V \tag{3}$$

$$y_{ijv} \leq k_v x_{ijv} \quad \forall i, j \in N_0, i \neq j, \forall v \in V \tag{4}$$

$$\sum_{j \in N_0, j \neq i} x_{ijv} = \sum_{j \in N_0, j \neq i} x_{jiv} \quad \forall i \in N_0, \forall v \in V \tag{5}$$

$$\sum_{j \in N_0, j \neq i} x_{ijv} \leq 1 \quad \forall i \in N, \forall v \in V \tag{6}$$

$$\sum_{v \in V} y_{iv}^L \leq s_i^0 \quad \forall i \in N_0 \tag{7}$$

$$\sum_{v \in V} y_{iv}^U \leq c_i - s_i^0 \quad \forall i \in N_0 \tag{8}$$

$$\sum_{i \in N_0} (y_{iv}^L - y_{iv}^U) = 0 \quad \forall v \in V \tag{9}$$

$$\sum_{i \in N} (L y_{iv}^L + U y_{iv}^U) + \sum_{i \in N} (L y_{0iv} + U y_{i0v}) + \sum_{i,j \in N_0 : i \neq j} t_{ij} x_{ijv} \leq T \quad \forall v \in V \tag{10}$$

$$q_{jv} \geq q_{iv} + 1 - M(1 - x_{ijv}) \quad \forall i \in N_0, j \in N, i \neq j, \forall v \in V \tag{11}$$

$$x_{ijv} \in \{0, 1\} \quad \forall i, j \in N_0 : i \neq j, \forall v \in V \tag{12}$$

$$y_{iv}^L \geq 0, y_{iv}^U \geq 0, \text{ integer} \quad \forall i \in N_0, \forall v \in V \tag{13}$$

$$y_{ijv} \geq 0 \quad \forall i, j \in N_0 : i \neq j, \forall v \in V \tag{14}$$

$$s_i \geq 0 \quad \forall i \in N_0 \tag{15}$$

$$q_{iv} \geq 0 \quad \forall i \in N_0, \forall v \in V \tag{16}$$

The objective function (1) minimizes the total cost of the system, consisting of the sum of the penalties incurred at all stations and the total operating costs, appropriately weighted by a factor of α. Constraints (2) are inventory-balance constraints at the nodes (the stations and the depot). Constraints (3) represent the conservation of inventory on the vehicles, and constraints (4) limit the quantity carried on each vehicle to its capacity. These constraints also set the quantity carried on a vehicle to zero when it travels directly from i to j if the vehicle does not use that arc. Constraints (5) are vehicle flow-conservation equations. Constraints (6) insure that each station is visited at most once by each vehicle and constraints (7) (resp., (8)) limit the quantity picked-up by all vehicles from a station (resp., delivered to a station) to the quantity available there initially (resp., the residual capacity of the station), see the "Discussion" below. Note that constraints (7) also imply non-negativity of the inventory variables, while constraints (8) also insure that the inventory at each station and at the depot is bounded by its capacity; therefore, these two restrictions are not written explicitly. Constraints (9) stipulate that all the

bicycles that are loaded onto the vehicles are also unloaded. Constraints (10) limit the total loading and unloading times plus the travel times to the total time available for the repositioning operation. Constraints (11) are sub-tour elimination constraints that are similar to those of Miller et al. (1960). Finally, (12) and (13) are binary and general integrality constraints, respectively, and (14)–(16) are non-negativity constraints. Note that the integrality of y_{ijv} and s_i is implied by the integrality of y^L_{iv}, y^U_{iv} and s^0_i.

Our next step is to replace the functions $f_i(s_i)$ by a linear term and linear constraints. Then the entire objective function would be linear (since the second term, $\alpha \sum_{i\in N_0} \sum_{j\in N_0} \sum_{v\in V} t_{ij}x_{ijv}$, is also linear) and the above formulation would become a MILP. This is possible, since each $f_i(s_i)$ is defined over integer inventory values and supported by a piecewise linear and convex function.

Recall that the values of $f_i(s_i)$ are known for all i and all $s_i = 0, \ldots, c_i$ from the pre-processing step. Using these values, we define and calculate for all $i = 1, \ldots, N$ and $u = 0, \ldots, c_i - 1$ the following parameters (u is the index over which s_i is defined):

$$b_{iu} \equiv f_i(u+1) - f_i(u)$$
$$a_{iu} \equiv f_i(u) - b_{iu} \cdot u.$$

From the above calculations, we can observe that b_{iu} is the marginal penalty (which may be negative) of the $(u+1)^{th}$ bicycle at station i. Together, a_{iu} and b_{iu} represent the intercept and slope, respectively, of the linear function that supports the convex cost function $f_i(.)$ at the level of u, see Fig. 1. These linear functions are added as constraints to the MILP and insure that the correct $f_i(s_i)$ value will be incurred in the objective function, given the chosen decision variable, s_i.

Specifically, linearity of the formulation is achieved by defining the following decision variables:

g_i Penalty incurred at station i;

(g_i is equal to $f_i(s_i)$, for the value of s_i determined by the MILP solution.)

We can now replace the previous convex terms in the objective function by linear terms:

$$Min \sum_{i\in N} g_i + \alpha \sum_{i\in N_0} \sum_{j\in N_0} \sum_{v\in V} t_{ij}x_{ijv} \tag{17}$$

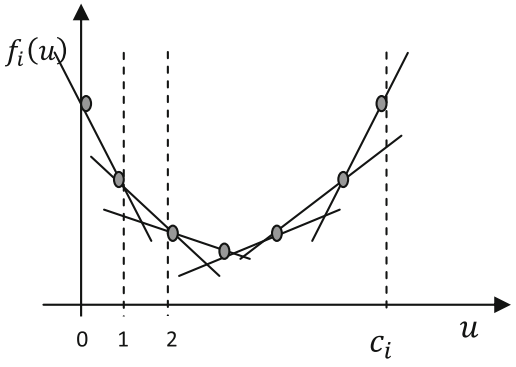

Fig. 1 Linear functions supporting $f_i(u)$

and add the following constraints to the formulation:

$$g_i \geq a_{iu} + b_{iu} s_i \quad \forall i \in N, u = 0, \ldots, c_i - 1 \tag{18}$$

The resulting MILP formulation consists of the objective function (17) subject to constraints (2)–(16) and (18). Note that since the convex term in (1) is defined over integer values only (number of bicycles), the linearization scheme, described by (17) and (18), is exact and not approximated. The following assumptions are embedded in the above formulation. First, as noted above, each station may be visited at most once by each vehicle, see constraints (6). However, a certain station may be visited by several vehicles. Thus, the total quantity picked-up from a station or delivered to it by all vehicles is limited as stated in constraints (7)–(8). Together these constraints verify that bicycles are not transshipped from a station before they are brought to it (or delivered to a station before space is available) during another visit of a different vehicle. In other words, these constraints exclude situations in which the inventory level at a station is negative or exceeds its capacity, in the interim of the repositioning period. In fact, these restrictions are more severe than required in practice, because they exclude the possibility of a vehicle picking-up bicycles that were previously brought by another vehicle, or delivering bicycles to a station that has available capacity due to another vehicle removing bicycles from it in an earlier visit. The time-indexed formulation presented in "Time-indexed formulation" relaxes this restriction, but is harder to solve.

Second, it is assumed that all bicycles brought to the depot by the vehicles are unloaded there, even if the vehicle is about to continue its trip and needs to load new bicycles in order to deliver them to other stations. This is due to the inability of the AI model to keep track of the total number of bicycles loaded and unloaded at the depot, and consequently the time that these operations consume. While this assumption is somewhat conservative, its effect on the solution is marginal.

Finally, vehicles may redistribute bicycles in the system in any way that improves the objective function. This means, e.g., that a vehicle may deliver bicycles to a station even if the station already has more than its "ideal" quantity, that is, the quantity that minimizes the first term of the cost function. Such an action may be justified if the bicycles are brought from a station where their presence is more costly. Since the objective is to minimize the overall cost in the system, this capability is desirable.

Solving the problem using the above formulation may become quite time consuming for problems of a realistic size. To reduce the running time, we propose two directions: one is adding valid inequalities that are specific to this formulation. Another is solving the problem through a two-stage heuristic, a method which is used also to reduce the running times of the other formulations, and is described in "A two-phase solution method".

The following valid inequalities may be added to the formulation:

$$\sum_{j \in N} x_{0jv} \geq 1 \quad \forall v \in V \tag{19}$$

Constraints (19) verify that each vehicle departs from the depot at least once. We observed numerically that this valid inequality proved to be particularly useful,

since without it, the number of vehicles that "leave" the depot in the LP relaxation, is fractional, according to the required capacity. Forcing a whole vehicle to leave the depot propagates by the vehicle flow-conservation Eq. (5) to all other visited nodes.

$$y^L_{iv} \le \min(s^0_i, k_v) \sum_{j \in N_0} x_{ijv} \quad \forall i \in N, \forall v \in V \tag{20}$$

$$y^U_{iv} \le min(c_i - s^0_i, k_v) \sum_{j \in N_0} x_{ijv} \quad \forall i \in N, \forall v \in V \tag{21}$$

Constraints (20)–(21) are somewhat similar to constraints (7)–(8), but refer to the loading and unloading quantity limitations of a single vehicle (rather than all vehicles). This enables tightening the right-hand side of the constraint by including the vehicle's capacity, and conditioning it only for cases in which the vehicle visits the station.

$$y^L_{iv} + y^U_{iv} \ge \sum_{j \in N_0} x_{ijv} \quad \forall i \in N, \forall v \in V \tag{22}$$

Constraints (22) eliminate some solutions where a vehicle enters a station without loading or unloading any bicycle there. It is valid when the distance matrix satisfies the triangle inequality because then it is always possible to skip a station with no loading and unloading activities.

In addition, if vehicles are identical it is possible to break symmetry by adding the following constraints:

$$\sum_{j \in N} j \cdot x_{0jv} \le \sum_{j \in N} j \cdot x_{0,j,v+1} \quad \forall v \in V \tag{23}$$

In "Numerical experiments", we demonstrate the capabilities of this formulation in solving problems of moderate size.

Time-indexed formulation

Our second formulation is based on discretizing the time available for repositioning into slots of short periods, say 5 min each. The length of each such period is denoted by τ. We define decision variables with an additional index representing the time period. We refer to this formulation as a *time-indexed* (TI) formulation. The advantage of this formulation is that the state of the system is represented by the decision variables at all points in (the discretized) time, which enables formulating complex situations properly. Indeed, the TI formulation extends the feasible region compared to the AI formulation as far as the visits and quantities loaded/unloaded at each station are concerned. Namely, it is no longer necessary to limit the number of bicycles transshipped via other stations as described in the AI formulation [see constraints (7) and (8)], and it is no longer necessary to limit each vehicle to visit each station at most once. In addition, the solution can prescribe the dwelling of the repositioning vehicle at some stations to allow possible synchronization with other vehicles. The ability to redistribute bicycles in the system in any way that reduces costs exists in the TI formulation in the same way as it does in the AI formulation, see the "Discussion" there.

In the formulation below, the discretized times are referred to as periods. We define a discretized travel-time matrix, denoted by t'_{ij}. These travel times are calculated by dividing the actual travel time by the period length τ, and rounding it up to the next integer, to insure feasibility. In addition, we set $t'_{ii} = 1$, where traveling from node i to itself represents remaining at the node for one period. Let $T' = T/\tau$ represent the number of periods in the planning horizon. T' is assumed to be integer.

The above discretization procedure is applied to the traveling times, and the smaller τ is chosen to be, the closer the resulting model is to the continuous time case. However, loading and unloading times are typically much shorter than traveling times, and discretizing them in the same manner would result in a model with too many decision variables or an unreasonable deviation from the continuous case. Therefore, later on we will show how loading/unloading times can be kept continuous and still incorporated in the discrete periodic model. To simplify the presentation, we first introduce the TI formulation assuming that loading/unloading times are zero and afterward explain how to generalize the formulation to reincorporate them. In this first, more simplistic model, it is assumed that the loading and unloading operations are carried out at the beginning of a discretized period before the vehicle leaves the node.

In this formulation, we present the linearized objective function and supporting constraints directly, both of which are identical to those of the AI formulation.

The decision variables used in the time-indexed formulation are as follows:

x_{ijtv} Binary variable that equals one if vehicle v starts to travel from node i to node j in period t, and zero otherwise
y^L_{itv} Number of bicycles loaded onto vehicle v at node i during period t
y^U_{itv} Number of bicycles unloaded from vehicle v at node i during period t
y_{ijtv} Number of bicycles carried from node i to node j by vehicle v during period t
s_{it} Inventory level at node i at the end of period t
g_i Cost incurred at station i (as in the AI formulation)

(P2)—Time-indexed (TI) formulation.

$$Min \sum_{i\in N} g_i + \alpha \sum_{i\in N_0}\sum_{j\in N_0}\sum_{t=1}^{T'}\sum_{v\in V} t_{ij}x_{ijtv} \tag{24}$$

s.t.

$$g_i \geq a_{iu} + b_{iu}s_{iT'} \quad \forall i \in N, u = 0,\ldots,c_i - 1 \tag{25}$$

$$s_{i0} = s_i^0 \quad \forall i \in N_0 \tag{26}$$

$$s_{it} = s_{i,t-1} + \sum_{v\in V}\left(y^U_{itv} - y^L_{itv}\right) \quad \forall i \in N_0, t = 1,\ldots,T' \tag{27}$$

$$s_{it} \leq c_i \quad \forall i \in N_0, t = 1,\ldots,T' \tag{28}$$

$$\sum_{j\in N_0} x_{0j0v} = 1 \quad \forall v \in V \tag{29}$$

$$\sum_{j\in N_0} x_{j,0,T'-t'_{j0},v} = 1 \quad \forall v \in V \tag{30}$$

$$\sum_{j\in N_0} x_{j,i,t-t'_{ji},v} = \sum_{k\in N_0} x_{iktv} \quad \forall i \in N_0, t = 1,\ldots,T'-1, \forall v \in V \tag{31}$$

$$\sum_{j\in N_0} y_{j,i,t-t'_{ji},v} = \sum_{k\in N_0} y_{iktv} + y^U_{itv} - y^L_{itv} \quad \forall i \in N_0, t = 1,\ldots,T'-1, \forall v \in V \tag{32}$$

$$y_{ijtv} \le k_v x_{ijtv} \quad \forall i \in N_0, t = 1,\ldots,T', \forall v \in V \tag{33}$$

$$y^L_{itv} \le min(c_i, k_v) \sum_{j\in N_0} x_{ijtv} \quad \forall \mathrm{i} \in \mathrm{N}_0, \mathrm{t} = 1,\ldots,\mathrm{T}', \forall \mathrm{v} \in \mathrm{V} \tag{34}$$

$$y^U_{itv} \le min(c_i, k_v) \sum_{j\in N_0} x_{ijtv} \quad \forall \mathrm{i} \in \mathrm{N}_0, \mathrm{t} = 1,\ldots,\mathrm{T}', \forall \mathrm{v} \in \mathrm{V} \tag{35}$$

$$y^L_{itv} \ge 0, y^U_{itv} \ge 0, \text{integers} \quad \forall i \in N_0, t = 1,\ldots,T', \forall v \in V \tag{36}$$

$$x_{ijtv} \in \{0,1\} \quad \forall i,j \in N_0, t = 1,\ldots,T', \forall v \in V \tag{37}$$

$$y_{ijtv} \ge 0 \quad \forall i,j \in N_0, t = 1,\ldots,T', \forall v \in V \tag{38}$$

$$s_{it} \ge 0 \quad \forall i \in N, t = 0,\ldots,T'. \tag{39}$$

The first term of the objective function (24) is identical to the linearized first term of the objective function of the AI formulation, while the second expresses the total traveling time of the vehicles in terms of the actual (non-discretized) times. Constraints (25) are the linear constraints that support the convex cost function, defined here with respect to the inventory at each station at the end of the last period, T'. It is equivalent to (18) in the arc-index formulation. Constraints (26) and (27) define the initial inventory and the inventory balance at the nodes while constraints (28) specify that the inventory at each node during each period be bounded by its capacity. Constraints (29) and (30) specify that each vehicle departs from the depot and returns to it at the beginning and at the end of the repositioning operation, respectively. Constraints (31) are vehicle flow-conservation equations; they stipulate that when a vehicle enters a node at some period (after traveling to it a certain number of periods according to its origin), it will leave the node at that period, possibly going to the same node itself. Thus, these constraints schedule the movement of vehicles consistently with the (discretized) distance matrix. Constraints (32) represent the conservation of inventory (bicycles) on the vehicles in each period, and constraints (33) limit the quantity carried by each vehicle in each period to the vehicle's capacity and stipulate that bicycles be carried only on the chosen arcs. Constraints (34) [resp., (35)] insure that no bicycles are loaded (resp., unloaded) at a node in each given period if the node is not visited during that period. Finally, constraints (36)–(39) are integrality and non-negativity constraints. In this formulation, any reference to an index of a vector or matrix that is out of bounds should be replaced by zero.

The above-mentioned important advantages of the TI formulation are made possible due to monitoring the inventory at each station in every period. These advantages enable additional flexibility in forming solutions, which is likely to improve the repositioning operation and consequently the optimal objective function's value. On the other hand, the additional flexibility needs to be weighed against the possible increased difficulty in solving this formulation to optimality, see "Numerical experiments".

We define $t'_{ii} = 1$ to allow remaining at the stations. Note that remaining at a station may be desirable exactly for the reasons indicated above, namely, to synchronize the visits of different vehicles at the station. As the AI formulation does not keep track of time, this capability cannot be observed and utilized by its solution.

On the other hand, the TI formulation suffers from a limitation which results from discretizing the travel times. To insure feasibility, the traveling times are rounded up, which causes unnecessary slacks in the schedule. To overcome this problem, as well as to allow loading/unloading times back into the formulation, we extend the above formulation by modifying and adding some constraints. The modifications are non-trivial, since the revised model combines a discrete and continuous representation of time. This enables us to gain the advantages of discreteness, discussed above, together with increased accuracy of the real time constraints of the system.

First, we add the following two sets of constraints:

$$\sum_{i,j\in N_0}\sum_{t\le w} t_{ij}x_{ijt-t'_{ij},v} + \sum_{i\in N_0}\sum_{t\le w}(Ly^L_{itv} + Uy^U_{itv}) \le w\cdot\tau$$
$$\forall w = 1,\ldots,T',\forall v\in V \tag{40}$$

$$\sum_{i,j\in N_0}\sum_{t\le w} t_{ij}x_{ijt-t'_{ij},v} + \sum_{i\in N_0}\sum_{t\le w}(Ly^L_{itv} + Uy^U_{itv}) \ge (w-2)\cdot\tau$$
$$\forall w = 1,\ldots,T',\forall v\in V \tag{41}$$

In constraints (40), the time restriction is enforced every period (i.e., every τ continuous time units). It allows monitoring closely the total time spent on all activities (traveling and loading/unloading). While the first term on the left-hand side of the constraint represents the total completed travel time of vehicle v on all arcs up to a certain discretized period w, the second term on the left-hand side of the constraint represents the total loading/unloading time spent by that vehicle up to the same period. Note that both terms on the left-hand side of this constraint represent continuous times, and so does the right-hand side. Since the actual travel times may be lower than the time of an integer number of periods, a slack may be created by the first term on the LHS of the constraint. This slack may be used by the second term of the LHS of the constraint, by loading/unloading a larger number of bicycles than is actually possible in a certain number of periods, see also constraints (42) and (43) below. In particular if a vehicle remains at some station i during a period, which is represented by traveling from node i to node i, the whole τ units of time can be spent loading and unloading bicycles since $t_{ii} = 0$ while $t'_{ii} = 1$.

Table 1 Comparing the assumptions and capabilities of the two formulations

Assumption	Arc-indexed	Time-indexed	Example of a scenario that is compatible with TI but not with AI
Vehicles are allowed to visit each station an arbitrary number of times	No. Each vehicle can visit each station at most once. Each station can be visited by several vehicles	Yes	A vehicle travels from the depot to station 1, unloads some bicycles, travels to station 2, loads some bicycles and then returns to station 1 to unload them
Transshipments are allowed, that is, vehicles can unload bicycles at nodes, to be loaded later on	Yes, but the maximum number of bicycles that can be unloaded (resp., loaded) cannot exceed the initial residual capacity (resp., initial inventory level) at the node	Yes, in an unlimited manner	A station is initially empty. At some point, a vehicle unloads 15 bicycles at the station and later on another vehicle loads five bicycles at this station (that initially were not there)
Vehicles can remain at stations in order to synchronize transshipments	No. Synchronization is guaranteed via restriction on loading and unloading quantities. See previous assumption	Yes	A vehicle arrives at a station, waits there for 5 min for a rendezvous with another vehicle and loads some bicycles that are unloaded from the other vehicle
Bicycles can be redistributed in the system in any way that reduces total costs, even if it is not locally optimal to do so.	Yes	Yes	
All bicycles are assumed to be unloaded at each visit of each vehicle to the depot	Yes	No	A vehicle arrives at the depot with ten bicycles on board, loads additional five bicycles and continues on its route. In the AI formulation, the ten bicycles must be unloaded first, and then the 15 bicycles are loaded. The time for the unloading and loading operations is spent
Other considerations and comments	This model delivered the best results for most of the instances in our numerical experiment, in spite of its restrictive assumptions	Accuracy is lost due to the time discretization. The model can be adapted to the dynamic problem	

In constraints (41), a lower bound is enforced on the time restriction, with the purpose of keeping the time of the planned schedule close to its execution. This is important for the accuracy of the inventory levels at the nodes, which is necessary for the synchronization among vehicles, i.e., making sure that no vehicle plans to pick-up bicycles which have not been brought there yet (by another vehicle). While lack of synchronization may still apply in the interval consisting of two periods defined between the upper and lower bounds of constraints (40) and (41), we believe that it is quite negligible.

Next, consider constraints (34) and (35) and replace them by the following constraints:

$$y^L_{itv} \leq \min(2\bar{L}, c_i, k_v) \sum_{j \in N_0} x_{ijvt} \quad \forall i \in N_0, t = 1, \ldots, T', \forall v \in V \tag{42}$$

$$y^U_{itv} \leq \min(2\bar{U}, c_i, k_v) \sum_{j \in N_0} x_{ijvt} \quad \forall i \in N_0, t = 1, \ldots, T', \forall v \in V \tag{43}$$

where the following two parameters are defined and assumed to be integers:

$\bar{L}$ Maximum number of bicycles that can be loaded during one period ($= \tau/L$)
$\bar{U}$ Maximum number of bicycles that can be unloaded during one period ($= \tau/U$)

These constraints limit the loading/unloading quantities during one period (beyond the original limitations), which are related to the time it takes to perform these operations. The limit is expressed as twice the actual number, to allow utilizing the slack that may have been created by the actual (rather than rounded up) travel times in constraints (40).

We conclude our discussion on the AI and the TI formulations with a summary of the capabilities and assumptions embedded in them, see Table 1.

Additional formulations

As the repositioning problem is relatively new and it is yet unclear which formulation approach performs best, we developed two additional formulations, which were tested numerically as well. The first is based on the idea of defining the journey of each vehicle according to the sequence of nodes that the vehicle visits. In this way, only one node index is required in the routing decision variables (the index of the node where the vehicle is located), instead of the two required in the previous two formulations (denoting the arc on which the vehicle traverses). Another index, in addition to the vehicle index, is the position (the stop number) of the node in the sequence. The sequence-indexed (SI) formulation handles time in an exact manner as in the AI formulation, but it allows several stops at a station as in the TI formulation. However, if the number of vehicles is greater than one, it constrains the number of bicycles transported to and from a certain station, as in constraints (7) and (8) of the AI formulation. Although it appears that this formulation would enjoy the benefits of both previous formulations, its numerical performance was found to be inferior to the arc-indexed formulation. Nevertheless, we found this formulation to be interesting and

potentially useful for variations of the problem discussed here, therefore it is presented in Appendix B.

Finally, we considered a formulation that is motivated by the similarity of our problem to the SP. As mentioned in the literature review, the SP is similar to ours in that objects need to be moved from nodes where they are initially available, to nodes where they are demanded. Consequently, using this formulation required us to define supply and demand nodes, and duplicate stations when (typically) a supply or a demand of more than one unit (bicycle) is required. This resulted in a problem with a lot more nodes than the number of stations, and thus was numerically inefficient. The formulation can be found in Raviv et al. (2012).

Algorithmic enhancements

Solving the formulations presented in the previous section using a commercial mixed integer solver may be impractical, even for instances of moderate size. In this section, we discuss ways to speed-up the computation times of the various formulations. They include solving the problems in two stages ("A two-phase solution method") and reducing the number of binary variables based on geographical considerations ("Arc deletion"). Some of these techniques are heuristic, while others are optimal, as explained below. In "Numerical experiments", we demonstrate that even when heuristic techniques are used, they have a marginal impact on the solution, and they typically contribute to improving the overall solution when a reasonable budget of time is allowed to solve the problem.

A two-phase solution method

The two-phase solution method is motivated by the variables representing the number of bicycles loaded or unloaded at the various nodes. While these variables are required to be non-negative integers, adding a tremendous difficulty in solving the problem, their precise values tend to have only a minor effect on the optimal routing decisions. Thus, in the two-phase solution method the problem is first solved while ignoring the integrality constraints of the loading/unloading variables, so that a solution to the routing decisions is obtained. In the second phase, the obtained routing variables are fixed to their solution from the first phase, and the rest of the problem is then solved, this time with the integrality constraints of the loading and unloading variables included. We demonstrate in more detail the motivation and implementation of this approach in the AI formulation, where the adaptation to the TI formulation is done in a similar manner.

In the two-phase solution approach, the first phase includes solving problem (P1′) which is identical to problem (P1), except that the integrality constraints in (13) are removed. It is still a mixed integer program, but one which is simpler to solve. Then, the second phase involves solving another mixed integer program, denoted by (P1″), obtained by fixing the values of the x-variables to the optimal values they obtained

in (P1′) and where the integrality of y^L_{iv} and y^U_{iv} is added back. Obtaining an optimal solution to problem (P1″) is achieved very quickly.

Specifically, consider the AI formulation before it is strengthened by the valid inequalities, given by the objective function (17) subject to constraints (2)–(16) and (18). In (P1″), the objective function (17) remains unchanged, and so do constraints (2), (3), (7)–(9) and (13)–(16), which do not include the x-variables. Then, constraints (5), (6), (11) and (12) which include x-variables only (and the related q-variables) are removed. Finally, the remaining constraints [(4) and (10)], which include both x and y variables, are modified as follows, using x^*_{ijv} to denote the solution of the x-variables obtained in the first phase:

$$y_{ijv} \le k_v x^*_{ijv} \quad \forall i, j, v \tag{44}$$

$$\sum_{i \in N_0} (L y^L_{iv} + U y^U_{iv}) \le T - \sum_{i,j \in N_0 : i \ne j} t_{ij} x^*_{ijv} \quad \forall v \in V \tag{45}$$

In these modified constraints, note that (44) in fact set to zero all y_{ijv} variables whose equivalent x^*_{ijv} variables equal zero. This is equivalent to removing all those variables, as indeed performed by the pre-solver. This modifies, indirectly, constraints (3) and (14), so that they are defined only for y_{ijv} variables whose equivalent x^*_{ijv} variables are equal to one. The modified constraints (45) are a tighter version of the time constraint (10), where the total travel time of a vehicle is subtracted from both sides of the inequality. The modifications with respect to the valid inequalities (19)–(23) are similar to those described above.

The solution to (P1″) delivers a feasible solution to (P1), with integer values of y^L_{iv} and y^U_{iv} and with an objective value that is numerically shown (in "Numerical experiments") to be very close to the lower bound obtained by solving (P1) without integrality constraints on these variables. We explain the attractiveness of this method by noting that the integrality constraint in (13) may be omitted if there are no loading and unloading times, i.e., $L = 0$ and $U = 0$. In such a case there always exists an integer optimal solution to (P1), because once the values of the x's are determined, the problem can be cast as a minimum cost flow problem with integer capacity and demand parameters. Positive values of loading or unloading times add a *knapsack* component to the problem through constraints (10) and hence further complicate it. However, we observed empirically, that even if the integrality constraints are removed, the number of non-integer values obtained in the optimal solution is small (in most cases, no more than two non-integers along the route of each vehicle).

Arc deletion

While the previous section described a way to (heuristically) overcome the integrality requirement concerning the loading and unloading variables, in this section we focus on the routing variables (the x-variables) in the TI model with the goal of significantly reducing their number. In this case, the reduction is *exact* rather than heuristic, and results from the following geographical considerations. In an

urban environment, the movement of vehicles is limited to the road network and is subject to various regulations such as "one-way streets", "no left turns", etc. Consequently, the shortest valid journey between a pair of stations is likely to pass via other stations. Thus, we can represent the stations network as a sparse graph where stations (i,j) are connected by an arc if and only if there is no other station, say k, such that $t_{ik} + t_{kj} = t_{ij}$. Otherwise, the journey from $i$to j can be represented by a path of two arcs, (i,k) and (k,j), which takes the same amount of time as the direct path from i to j. Using this observation, a significant number of variables that correspond to arcs may be reduced. This idea is implemented by defining the set $\delta_i = \{j \in N_0 : t_{ij} < t_{ik} + t_{kj} \forall k \in N_0\}$ for all $i \in N_0$. In order to accommodate the *arc deletion*, we revise the definition of the indices of the routing variables, and possibly other related variables. For example, the x and y variables in (37) and (38) are defined as follows:

$$x_{ijtv} \in \{0,1\},\ y_{ijtv} \geq 0 \quad \forall i \in N_0, j \in \delta_i, \forall v \in V, t = 1,\ldots,T', v \in V$$

In addition, reference to undefined variables should be removed by revising all constraints in which these variables appear.

Note that the above arc deletion procedure is different from the *heuristic concentration* method, see e.g., Rosing and ReVelle (1997). The latter is a known approach to reducing the number of arcs by using information from previous runs, but it cannot insure that the removed arcs are not part of the optimal solution. Our procedure, on the other hand, removes only arcs that, without a doubt, can be replaced by alternative paths with the same cost. To the best of our knowledge, this idea has not been introduced in the literature and we believe it may be beneficial for other vehicle and IRPs. By using the arc deletion procedure, we drastically reduce the size of our mixed integer program and increase the size of instances that can be solved in a reasonable amount of time. In the largest instances that we tested, about 80 % of arcs could be removed.

We remark that the above method may not be applied to the AI formulation because in this formulation each vehicle is allowed to visit each station only once. If visits are "wasted" on constructing routes to other nodes when the inventory or the residual capacity on the vehicle is not sufficient to serve the station, then this limitation becomes too restrictive.

Numerical experiments

In this section, we present results that were obtained when solving instances of practical size with the MILP formulations introduced in "Model formulation", together with the algorithmic enhancements presented in "Algorithmic enhancements". We first describe and analyze instances that are based on data of the Vélib system (in Paris). Then we apply the formulation that performed best to an entire real system consisting of 104 stations of Capital Bikeshare in Washington DC.

Based on the Paris dataset, we created a set of 48 benchmark problems using actual locations of some 60 Vélib stations located in Paris's 1st quarter. We conducted an experiment with all the combinations of the following parameter values:

- Number of stations—30 and 60 stations, where the smaller instances are subsets of the larger ones.
- Penalty function—representing the expected number of shortages. Two penalty functions were considered, based on two different demand patterns. The first is based on a fictitious (but likely a representative) demand pattern while the second is based on an estimated demand pattern from data that we collected from the Vélib website during some 10 working days. See Appendix A for details on the procedure used to calculate these functions.
- Number of repositioning vehicles—one and two vehicles.
- Length of the repositioning time—2.5 and 5 h (9,000 and 18,000 s).
- α, the weight of the operating/travel costs per second relative to an expected shortage of one unit—three values were considered: 0, 1/900 (low) and 1/300 (high). The meaning of, say, $\alpha = 1/900$, is that traveling 900 s (= 15 min) is equivalent (in cost) to an additional expected shortage. When $\alpha = 0$, it means that we are willing to travel *any* distance in order to save on the expected number of shortages. In this case, the objective function reduces to its first term only.

The initial inventory levels at the stations were selected randomly. The travel-time matrix was calculated based on the L_1 (Manhattan) metric, assuming average travel speed of 1 m/s. The loading and unloading times were set to be 1 min/bicycle. The vehicle's capacity was set to 20 bicycles, which is the capacity of the light trucks used by Vélib. The location of the depot was selected arbitrarily in the first quarter. The capacity and the initial inventory of the depot were set to be large enough so that they were not binding. The dataset for our benchmark problems is available from the authors upon request.

We implemented the AI, the AI with the two-phase procedure (AI2), and the TI formulation with the two-phase procedure (TI2) using IBM-Ilog OPL, and solved the above instances using IBM-Ilog CPLEX 12.3 on an Intel i7 2600 @ 3.4 GHz with 16 GB of RAM. In all our experiments, we used CPLEX's default settings and set the solver time limit to 2 h. Under such a time limit, the main memory was never exploited. For the two-phase procedure, the time limit was imposed only on the first phase, since the time needed to perform the second phase was negligible (less than 1 s). The optimality tolerance of the solver was left as CPLEX's default of 0.01 %.

We also experimented with the single-phase TI formulation as well as with the two-phase SI and Swapping Based formulations (see Appendix B), but found that the performance of these formulations was generally inferior to that of the formulations reported below. These results are omitted for the sake of brevity.

In Table 2, we report on the results of our experiments with the AI formulation for the 48 problem instances created as explained above. The five left columns of the table describe each instance in terms of the number of stations, the penalty function, the number of vehicles used for repositioning, the repositioning time in seconds and the α parameter. In the sixth column, we specify the ideal/initial expected number of shortages that would be obtained if all stations could reach their ideal inventory levels/left with their initial inventory level. Note that these numbers are crude lower and upper bounds on the optimal value of the objective function.

Table 2 Results of the AI model—Paris instances

Stations	Penalty function	Vehicles	Time (s)	α	Ideal/ initial	To add/remove	Best integer	Opt. gap (%)	CPU time (s)	Shortage	Added/ removed	Travel time (s)
30	1	1	9,000	Zero	107.58/ 288.03	246/64	214.24	0.01	9	**214.24**	49/43	3,078
				1/900			217.66	0.01	24	**214.24**	49/43	3,078
				1/300			224.30	0.01	11	215.92	54/34	2,514
			18,000	Zero			167.07	0.01	642	**167.07**	95/68	6,582
				1/900			174.30	0.01	912	167.30	97/77	6,302
				1/300			186.16	0.01	67	171.18	112/53	4,494
		2	9,000	Zero			170.84	7.66	–	**170.84**	103/50	5,544
				1/900			177.64	8.19	–	171.61	104/46	5,428
				1/300			188.61	7.39	–	171.63	107/52	5,094
			18,000	Zero			116.83	3.13	–	116.83	193/77	12,792
				1/900			131.11	3.41	–	**116.75**	191/74	12,926
				1/300			158.35	4.85	–	120.83	194/84	11,256
	2	1	9,000	Zero	51.42/ 186.03	301/9	139.52	0.00	4	**139.52**	53/13	2,628
				1/900			142.44	0.00	4	**139.52**	53/13	2,628
				1/300			148.10	0.01	3	140.15	55/0	2,384
			18,000	Zero			109.33	0.01	1,171	**109.33**	99/7	6,114
				1/900			115.98	0.01	366	109.64	102/9	5,702
				1/300			128.46	0.01	47	**109.93**	103/9	5,558
		2	9,000	Zero			108.90	2.80	–	**108.90**	103/8	5,580
				1/900			115.10	3.09	–	**108.90**	103/8	5,580
				1/300			127.50	3.25	–	**108.90**	103/8	5,580
			18,000	Zero			69.13	3.83	–	69.13	182/16	14,018
				1/900			84.70	4.19	–	**69.12**	182/16	14,018
				1/300			113.77	6.39	–	74.25	194/36	11,856

Table 2 continued

Stations	Penalty function	Vehicles	Time (s)	α	Ideal/ initial	To add/remove	Best integer	Opt. gap (%)	CPU time (s)	Shortage	Added/ removed	Travel time (s)
60	1	1	9,000	Zero	206.82/ 537.69	449/160	460.87	0.01	3,520	**460.87**	43/43	3,838
				1/900			465.13	0.01	2,685	**460.87**	43/43	3,838
				1/300			473.66	0.01	4,221	**460.87**	43/43	3,838
			18,000	Zero			382.60	0.01	464	**382.60**	99/95	6,088
				1/900			389.09	0.01	1,039	382.91	103/88	5,564
				1/300			401.46	0.01	675	382.91	103/88	5,564
		2	9,000	Zero			404.48	9.65	–	404.48	84/80	7,890
				1/900			405.13	8.57	–	**397.05**	89/85	7,268
				1/300			429.76	11.23	–	406.36	91/74	7,020
			18,000	Zero			310.14	12.74	–	310.14	190/137	13,084
				1/900			320.43	12.50	–	**306.56**	195/128	12,486
				1/300			357.45	15.86	–	313.21	189/145	13,272
	2	1	9,000	Zero	92.36/ 339.41	547/47	289.55	0.52	–	**289.55**	43/26	3,816
				1/900			293.81	0.78	–	289.57	43/26	3,818
				1/300			301.37	0.75	–	291.80	51/11	2,872
			18,000	Zero			242.59	0.01	248	**242.59**	98/58	6,166
				1/900			249.39	0.01	85	242.61	99/59	6,098
				1/300			262.72	0.01	181	242.96	100/60	5,928
		2	9,000	Zero			252.71	8.82	–	252.71	92/29	6,932
				1/900			259.32	8.94	–	**251.87**	93/37	6,702
				1/300			272.89	9.74	–	252.95	99/26	5,982
			18,000	Zero			195.39	11.18	–	195.39	177/65	14,722
				1/900			205.61	9.72	–	**191.25**	192/63	12,926
				1/300			234.66	11.19	–	191.28	191/68	13,014

The gap between them indicates the savings opportunity that could be gained from repositioning. In the seventh column, we present the total number of bicycles that should be added/removed to/from all stations in the system (excluding the depot) in order to bring it from its initial state to the ideal one.

The rest of the columns summarize the results for each instance. The first column reports on the value of the best integer obtained. We then present the provable relative optimality gap, calculated according to the CPLEX convention, that is,

$$Relative\ Gap(AI) = \frac{Best\ Integer\ Solution - Lower\ Bound}{Best\ Integer\ Solution}.$$

The next column presents, for the instances that could be solved within the 2-h time limit, the CPU time (in seconds) required to obtain the optimal solution (within the optimality tolerance of 0.01 %).

The column entitled "shortage" represents the expected number of shortages associated with the inventory level obtained after repositioning according to the best integer solution, i.e., the value of the first term of the objective function. For the case of zero traveling cost ($\alpha = 0$), this value is identical to the objective function, and it is an important measure even when $\alpha > 0$. Hence, the value (or values, when more than one exists) which achieves the minimum expected shortages among the three values of α that were considered for a certain instance is marked in bold.

The next column presents the number of bicycles that were actually added and removed from the various stations (excluding the depot). In the right-most column, the total traveling time of the vehicle(s) is presented, in seconds. This time is net of the time spent on loading and unloading bicycles and it is proportional to the traveling cost component of the objective function.

We obtained feasible solutions for all 48 instances with an average optimality gap of about 4 %. It appears that the optimality gap is not very sensitive to the traveling cost, and the penalty function being used. Interestingly, the average optimality gap was only slightly affected by the length of the repositioning time. This may imply that while geographical decomposition of the problem may be beneficial, a temporal decomposition is not likely to be beneficial when using this formulation. However, since the dimensions of the MILP are proportional to the number of vehicles and to the square of the number of stations, the performance is significantly affected by these parameters. For example, instances with 30 stations and a single vehicle were solved to optimality (within the 0.01 % tolerance) in 531 s on average, while none of the instances with 60 stations and two vehicles was solved to optimality within the 7,200 s time limit. The average optimality gap of these instances was 5.97 %.

It is apparent from Table 2 that when $\alpha = 1/900$, the total traveling time is very similar to the total traveling time when $\alpha = 0$, i.e., when the travel costs are ignored altogether. However, when α rises to 1/300, the model chooses to save on traveling at the expense of some additional shortages. An interesting observation is that the expected shortages obtained for instances with low travel costs ($\alpha = 1/900$) are on average 0.41 % lower than the ones obtained for the corresponding instances with zero travel costs, and they are strictly lower in half of the cases. This clearly could not be the case if these instances were solved to optimality. However, it appears that

the travel costs serve as a guide for the branch and bound search for an efficient route. Hence, it may be beneficial to introduce small travel costs to the model even if all the costs that are related to the vehicles are sunk.

An interesting observation is that in some instances, the actual number of bicycles removed from all stations is larger than the number of bicycles that need to be removed in order to reach their ideal inventory levels. This implies that bicycles were removed even from stations that ended up below their ideal inventory level. While this situation may seem counter-intuitive at first, it represents one of the attractive features of our formulations. Indeed, shortages can be saved by moving bicycles from stations that are near their ideal inventory level, to stations that are far off this level.

Typically, in the solutions for zero and low travel cost (α) instances, the vehicles spent about a third of the time in traveling, while most of the remaining time was spent in loading and unloading bicycles at stations. In the solutions of the higher travel cost instances, only about a quarter of the time was spent in traveling, and in some cases, the vehicle was even idle for some time.

Next, in Table 3 we present the results of an experiment with the same instances, using the two-phase procedure applied to the arc-indexed formulation (AI2). The information reported in this table is in the same structure as Table 2, expect that we added the column headed "Ph. 2 Gap" which reports on the relative difference between the results obtained at Phase 1 (when the loading and unloading variables are not forced to be integers) and the integer feasible results of Phase 2. That is,

$$Phase\,2\,Gap = \frac{Best\ Integer\ solution\ of\ Phase\,2 - Best\ Integer\ solution\ of\ Phase\,1}{Best\ Integer\ solution\ of\ Phase\,2}.$$

Note that the Phase 2 gap values are minor, with an average and maximum gap of 0.17 and 0.35 %, respectively, which demonstrates the effectiveness of the two-phase method. The "Opt Gap" column has a slightly different meaning. It compares the best integer (feasible) solution of Phase 2 with the lower bound of Phase 1. Note that the latter is a valid lower bound for the original AI formulation, and thus the "Opt Gap" value is a valid optimality gap for the performance of the AI2 method. In the CPU time column, we report only on the Phase 1 solution time, since the second phase was solved to optimality in a fraction of a second in all the instances.

In the best integer and optimality gap columns we set in bold faces (resp., denote with an *) the values that correspond to instances for which the two-phase procedure, AI2, produced better (resp., same) results relative to the AI formulation. It is apparent that although AI2 did not dominate AI completely, it delivered better solutions for some of the harder instances that could not be solved to optimality by AI within 2 h. It also converged to optimality faster than AI in most of the easier instances and exhibited an overall better performance. The average (resp., maximum) optimality gap obtained by AI2 was 2.27 % (resp., 7.5 %) compared to 3.97 % (resp., 15.86 %) obtained by AI.

As with the AI formulation, the performance of AI2 was adversely affected by both the number of stations and the number of vehicles, while the effect of the length of the repositioning time is minor. It is worth mentioning that although most

Table 3 Results of the AI2 formulation—Paris instances

Stations	Penalty function	Vehicles	Time (s)	α	Ideal/ initial	To add/ remove	Best integer	Opt. gap (%)	Ph. 2 gap (%)	CPU time (s)	Shortage	Added/ removed	Travel time (s)
30	1	1	9,000	Zero	107.58/ 288.03	246/64	214.52	0.35	0.35	6	214.52	48/45	3,132
				1/900			218	0.35	0.35	5	214.52	48/45	3,132
				1/300			224.54	0.26	0.25	9	215.19	51/36	2,806
			18,000	Zero			167.1	0.22	0.21	141	167.1	95/67	6,530
				1/900			174.36	0.21	0.20	198	167.1	95/67	6,530
				1/300			186.16*	0.06	0.05	59	171.18	112/53	4,494
		2	9,000	Zero			171.43	**4.74**	0.19	–	171.43	105/49	5,256
				1/900			**177.27**	**4.27**	0.18	–	171.43	105/49	5,256
				1/300			**188.58**	**2.56**	0.08	–	171.6	107/53	5,094
			18,000	Zero			**116.8**	**2.21**	0.11	–	116.8	190/72	13,096
				1/900			**130.73**	**2.17**	0.10	–	**116.47**	192/73	12,832
				1/300			**158.17**	**3.11**	0.00	–	120.65	195/84	11,256
	2	1	9,000	Zero	51.42/ 186.03	301/9	139.52*	0.02	0.02	5	139.52	53/13	2,628
				1/900			142.44*	0.02	0.02	4	139.52	53/13	2,628
				1/300			148.10*	0.04	0.04	4	140.15	55/0	2,384
			18,000	Zero			109.52	0.34	0.33	651	109.52	100/7	5,886
				1/900			116.06	0.32	0.31	124	**109.52**	100/7	5,886
				1/300			128.46*	0.11	0.10	33	109.93	103/9	5,558
		2	9,000	Zero			108.90*	**0.58**	0.13	–	108.9	103/8	5,580
				1/900			115.10*	**0.80**	0.13	–	108.9	103/8	5,580
				1/300			127.50*	**0.12**	0.11	4,981	108.9	103/8	5,580
			18,000	Zero			**68.98**	**2.88**	0.28	–	68.98	183/16	13,896
				1/900			**84.41**	**3.39**	0.23	–	**68.97**	183/17	13,896
				1/300			113.93	**4.74**	0.03	–	70.5	191/23	13,028

Table 3 continued

Stations	Penalty function	Vehicles	Time (s)	α	Ideal/ initial	To add/ remove	Best integer	Opt. gap (%)	Ph. 2 gap (%)	CPU time (s)	Shortage	Added/ removed	Travel time (s)
60	1	1	9,000	Zero	206.82/ 537.69	449/ 160	461.33	0.18	0.17	5,669	461.33	43/43	3,774
				1/900			465.52	0.18	0.17	4,557	461.33	43/43	3,774
				1/300			473.91	0.17	0.16	2,535	461.33	43/43	3,774
			18,000	Zero			382.60*	0.06	0.05	522	**382.6**	99/95	6,088
				1/900			389.09*	0.13	0.13	479	382.91	103/88	5,564
				1/300			401.46*	0.13	0.12	456	382.91	103/88	5,564
		2	9,000	Zero			**403.47**	**6.56**	0.08	–	403.47	87/71	7,438
				1/900			408.66	**5.88**	0.24	–	400.27	86/82	7,554
				1/300			**423.6**	**6.43**	0.02	–	**398.9**	88/79	7,410
			18,000	Zero			**297.18**	**5.53**	0.07	–	297.18	189/142	13,242
				1/900			**311.67**	**5.90**	0.10	–	**296.98**	189/144	13,218
				1/300			**342.2**	**7.50**	0.15	–	297.59	187/154	13,384
	2	1	9,000	Zero	92.36/ 339.41	547/47	289.76	0.74	0.15	–	289.76	43/26	3,764
				1/900			293.94	0.83	0.15	–	**289.76**	43/26	3,764
				1/300			301.37*	**0.72**	0.09	–	291.98	51/11	2,818
			18,000	Zero			242.59*	0.12	0.11	388	242.59	98/58	6,166
				1/900			249.44	0.11	0.11	484	242.59	98/58	6,166
				1/300			262.72*	0.07	0.07	497	242.96	100/60	5,928
		2	9,000	Zero			**251.87**	**5.58**	0.15	–	**251.87**	94/30	6,594
				1/900			259.54	**6.21**	0.27	–	252.14	93/38	6,664
				1/300			274.07	**6.62**	0.14	–	253.8	98/27	6,082
			18,000	Zero			**188.79**	**5.17**	0.12	–	188.79	177/67	14,678
				1/900			**203.19**	**4.64**	0.12	–	**187.87**	184/64	13,788
				1/300			**233.03**	**5.64**	0.08	–	189.58	190/81	13,034

of the larger instances could not be solved to optimality within 2 h, most of the progress toward the obtained solution was achieved by the solver during the very first minutes. Thus, the AI2 (and AI) formulations are applicable even under much tighter time constraints.

Next, we present the results of our experiment with the TI2 procedure and while applying the arc deletion enhancement described in "Algorithmic enhancements". We applied this formulation to all our 48 instances with two different time discretization levels, namely $\tau = 300$ s and $\tau = 450$ s per period, resulting in 96 runs in total. None of the problems converged to optimality within the 2-h time limit, but high quality solutions could be obtained for many instances. For each instance we set in bold faces (resp., denote with an *) the value that corresponds to the discretization level that produced a better (resp., same) result.

The results are presented in Table 4, where the columns' headings are similar to those of the previous tables and thus we omit their description. As with the AI2 formulation, the Phase 2 objective function values were very close (within 0.59 %) to those of Phase 1 and thus the latter are not reported in the table. Cells related to instances that could not be solved within the time limit were marked with dashes ("–").

It is apparent from Table 4 that the performance of the TI2 formulation, similar to the AI and AI2 formulations, is adversely affected by the number of stations and the number of vehicles. However, in contrast to AI and AI2, TI2 is also adversely affected by the repositioning time, because it directly affects the number of variables in the model. This suggests that when using the TI2 formulation, temporal decomposition of the problem may be beneficial.

We observe that the crude time discretization $(\tau = 450\ \text{s})$ typically delivers better solutions and smaller optimality gaps for the harder problem instances, while the finer time discretization $(\tau = 300\ \text{s})$, which results in a much larger mathematical model, yields better solutions only for a few single vehicle instances, mostly with 30 stations. For five (resp., eight) out of the 48 instances the solver could not even obtain an integer feasible solution within the time limit when using the crude (resp., fine) time discretization.

In Fig. 2, we compare the values of the solutions obtained for each instance using AI, AI2, and TI2 with $\tau = 300$ and $\tau = 450$ s. For each method, we counted the number of instances for which it achieved the best-known solution, i.e., had the minimal objective value among all methods. Note that in some cases, this minimal value was obtained by more than one method. In the figure, the one and two vehicle instances were distinguished by the shade used in the chart. It is apparent from the figure that while the *optimal* solution value of the time-indexed formulation may be lower, due to the additional flexibility of this formulation, the arc-indexed formulations "win" in most of the instances and in particular in all of the harder two vehicle instances, since they are easier to solve. The AI formulation works best in most of the single vehicle instances and AI2 works best in most of the, harder, two vehicles instances. The time-indexed formulations achieved the best-known solution in only a few of the single vehicle instances. The rougher time discretization procedure seems to perform somewhat better than the finer one, probably because it is easier to solve.

Table 4 Results for the time-indexed formulation with the two-phase procedure and arc deletion (TI2)—Paris instances

Instances					Fine time discretization $\tau = 300$ s			Crude time discretization $\tau = 450$ s		
Stations	Penalty function	Vehicles	Time (s)	α	Best integer	Opt. gap (%)	Shortage	Best integer	Opt. gap (%)	Shortage
30	1	1	9,000	Zero	214.50*	4.83	214.5	214.50*	2.57	214.5
				1/900	**217.98**	5.41	214.5	218.31	3.54	215.19
				1/300	224.40*	6.33	215.05	224.40*	3.37	215.05
			18,000	Zero	174.64	12.99	174.64	**170.37**	10.68	170.37
				1/900	179.54	12.83	174.1	**177.38**	11.64	171.62
				1/300	189.5	12.78	172.27	**185.89**	10.93	170.14
		2	9,000	Zero	180.18	13.41	180.18	**173.16**	9.90	173.16
				1/900	184.72	12.80	177.85	**177.96**	9.21	171.88
				1/300	200.32	14.94	182.44	**189.2**	9.40	172.12
			18,000	Zero	–	–	–	**125.54**	10.73	125.54
				1/900	186.04	33.15	174.99	**159.72**	22.21	146.64
				1/300	**198.6**	26.60	169.54	215.73	32.41	179.58
	2	1	9,000	Zero	139.52*	3.76	139.52	139.52*	2.89	139.52
				1/900	142.44*	3.96	139.52	142.44*	3.08	139.52
				1/300	148.16*	4.44	141.63	148.16*	3.28	141.63
			18,000	Zero	109.66	9.68	109.66	**108.74**	8.83	108.74
				1/900	116.25	10.60	109.8	**115.1**	9.52	108.9
				1/300	127.6	11.27	109.65	**127.5**	10.92	108.9
		2	9,000	Zero	115.44	11.83	115.44	**110.29**	7.30	110.29
				1/900	119.12	10.11	112.62	**116.49**	7.29	110.11
				1/300	130.66	10.53	112.54	**128.22**	8.80	109.62
			18,000	Zero	91.85	28.81	91.85	**74.48**	12.68	74.48
				1/900	–	–	–	**89.01**	12.09	73.69
				1/300	126.76	20.26	96.59	**123.77**	18.33	93.29

Table 4 continued

Instances					Fine time discretization $\tau = 300$ s			Crude time discretization $\tau = 450$ s		
Stations	Penalty function	Vehicles	Time (s)	α	Best integer	Opt. gap (%)	Shortage	Best integer	Opt. gap (%)	Shortage
60	1	1	9,000	Zero	462.75	5.40	462.75	**460.82**	4.45	460.82
				1/900	466.39	5.56	462.75	**465.01**	5.18	460.82
				1/300	473.66*	5.79	462.75	473.66*	5.21	462.75
			18,000	Zero	416.62	14.87	416.62	**384.53**	7.62	384.53
				1/900	452.5	20.66	447.51	**392.05**	8.31	385.81
				1/300	418.24	12.23	396.71	**416.5**	11.60	396.34
		2	9,000	Zero	443.11	17.35	443.11	**402.06**	8.62	402.06
				1/900	435.88	14.67	426.82	**419.36**	11.16	410.58
				1/300	470.18	18.83	447.56	**440.4**	13.12	413.18
			18,000	Zero	–	–	–	–	–	–
				1/900	–	–	–	–	–	–
				1/300	–	–	–	–	–	–
	2	1	9,000	Zero	289.76*	5.44	289.76	289.76*	5.09	289.76
				1/900	294.68	6.27	290.98	**293.94**	5.52	289.76
				1/300	301.86	6.97	292.47	**301.37**	6.48	291.98
			18,000	Zero	249.56	8.99	249.56	**247.11**	7.93	247.11
				1/900	269.13	13.89	260.97	**256.56**	9.54	250.22
				1/300	279.15	13.82	257.64	**275.06**	12.35	254.57
		2	9,000	Zero	264.1	12.06	264.1	**260.69**	10.59	260.69
				1/900	**264.36**	9.96	257.64	264.4	9.83	257.08
				1/300	279.21	11.64	260.36	**276.3**	10.28	255.55
			18,000	Zero	–	–	–	–	–	–
				1/900	–	–	–	**242.59**	24.43	230.47
				1/300	–	–	–	–	–	–

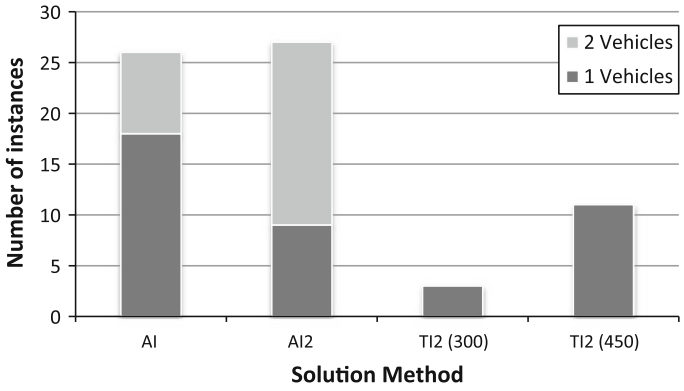

Fig. 2 Number of instances for which each solution method achieved the best-known solution

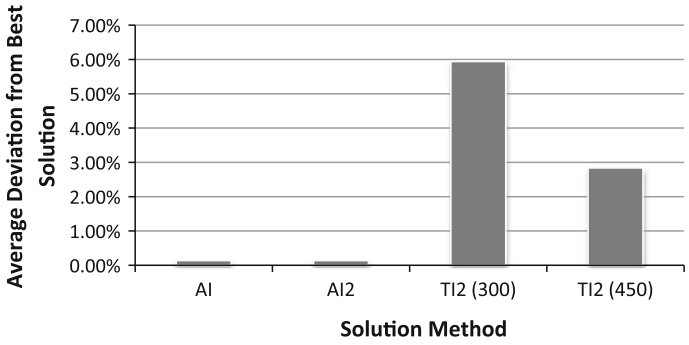

Fig. 3 Average relative deviation from best known solution obtained from each of the methods

In Fig. 3, we compare the average relative deviation of the solutions obtained from each method with the best-known solutions for these instances. These averages were calculated based on the 40 problem instances that admitted a feasible solution by all four methods and therefore it is somewhat biased in favor of the TI2 procedure. Nevertheless, the figure demonstrates the significant superiority of the solutions obtained by the AI methods (on average).

We believe that the TI2 formulation may be superior to the arc-indexed formulations in systems where the size of each station is large compared to the capacity of the repositioning vehicle. In such cases, multiple visits of vehicles to the stations are desired. The Time-Indexed formulation also has the advantage that it can be modified quite easily to take into consideration demand that is expected to occur within the repositioning time, as is the case in the dynamic repositioning problem.

In our second experiment, we applied our models and solution methods on real problem instances obtained from the operator of Capital Bikeshare in Washington DC. The distance matrix represents real driving times between each pair of stations. The arrival rates of renters and returners for each hour of the day on regular

weekdays were estimated based on a complete transaction log collected during a 3-month period. With an expectation of a demand increase at Capital Bikeshare, we created another set of instances with the same network but with demand rates multiplied by two and stations' size multiplied by 1.5. This set is referred to as the "inflated" set. A penalty function for each of the systems' 104 stations was created based on the above demand data and station capacity.

We solved the static repositioning problem of these systems using one and two vehicles with repositioning times of 9,000 and 18,000 s (2.5 and 5 h, respectively). The capacity of the vehicles in the instances below is set to 25, the same as the light trucks used by the operator of Capital Bikeshare. Since we could not obtain a firm estimate of the vehicle's variable travel costs, we solved the problem with the same three α values as before, namely: 0, 1/900 and 1/300. In fact, we believe that the zero or the low travel costs are closely representative of reality, since most of the vehicle's operational cost for a predefined shift length are sunk.

We applied the AI, AI2 and TI2 (with crude time discretization) formulations with a solution time limit of 2 h and discovered that for these instances the AI2 formulation dominated. Hence, we report in Table 5 only on the results obtained from this procedure. The structure of the table is similar to that of Table 3 and hence we omit its full description. A column headed "% Job done" was added, which reports on the relative reduction in the expected shortages, calculated as follows:

$$\%\ Job\ done = \frac{initial\ expected\ shortages\ -\ expected\ shortage\ after\ repositioning}{initial\ expected\ shortages\ -\ ideal\ expected\ shortages} \times 100.$$

The values in boldface in the "shortage" column are again those that achieved the minimum shortages among the three traveling cost values.

It is apparent from the table that high quality feasible solutions with optimality gaps of a few percent can be obtained using the AI2 procedure. There is no significant difference between the optimality gaps of the original instances and the inflated ones. This implies that the method is robust to the traffic volume of the system. As expected, the value of the objective function decreases as the number of vehicles and time allotted to the repositioning increases. Accordingly, the % of the job done also increases.

We observe that even when the objective function value of the best solution found is close to the ideal value and far apart from the initial one, only a small fraction of the bicycles that should have been ideally added and removed at the stations are actually moved. See e.g., the tenth and eleventh instances in the table. This is due to the fact that the first (and major) term of the objective function is convex and typically almost flat around its minimum. Indeed, it seems that in some stations the operator is almost indifferent between the exact optimal inventory and many other possibilities close to it. We believe that this very phenomenon justifies the use of a shortage function rather than setting target values for the inventory at each station and solving a "many to many" single commodity PDP with a standard objective function of total distance. Indeed, the operator should allocate its limited

Table 5 AI2 results for Washington DC instances

Demand	Vehicles	Time (s)	α	Ideal/initial	To add/remove	Best int.	Opt. gap (%)	Shortage	Travel time (s)	Added/removed	% Job done (%)
Original	1	9,000	Zero	189.33/284.29	150/197	227.7	2.47	227.7	3,757	43/43	59.60
			1/900			231.15	2.37	**227.15**	3,601	44/44	60.17
			1/300			239.28	2.66	227.16	3,635	44/44	60.16
		18,000	Zero			199.72	1.29	**199.72**	7,757	85/85	89.06
			1/900			208.84	1.93	200.29	7,702	85/85	88.46
			1/300			224.97	2.53	203.16	6,544	95/77	85.44
	2	9,000	Zero			209.66	6.34	209.66	9,770	67/66	78.59
			1/900			212.97	4.26	204.88	7,275	79/87	83.62
			1/300			229.92	5.14	**203.93**	7,797	83/85	84.63
		18,000	Zero			195.09	2.95	195.09	19,384	120/129	93.93
			1/900			207.98	3.56	**191.95**	14,429	116/127	97.24
			1/300			223.79	3.01	195.25	8,561	106/127	93.76
Inflated	1	9,000	Zero	413.47/592.48	243/314	510.87	0.32	**510.87**	3,212	48/48	45.61
			1/900			514.44	0.31	510.87	3,212	48/48	45.61
			1/300			522.83	0.76	512.48	3,105	49/49	44.71
		18,000	Zero			456.7	1.25	**456.7**	6,187	98/98	75.90
			1/900			464.02	1.48	456.97	6,344	97/97	75.74
			1/300			478.97	2.03	457.73	6,371	96/96	75.32
	2	9,000	Zero			470.22	4.54	470.22	7,634	86/86	68.34
			1/900			467.92	2.72	**460.39**	6,784	92/92	73.83
			1/300			490.89	4.82	471.15	5,923	91/81	67.82
		18,000	Zero			420.67	1.37	**420.67**	15,618	168/167	96.03
			1/900			436.58	2.09	424.41	10,950	157/184	93.94
			1/300			479.71	6.34	429.33	15,114	158/160	91.19

repositioning capacity to the stations that are expected to face the largest number of shortage events during the next day.

We observe that in Capital Bikeshare, even if the inventory levels could be set at their ideal value, about two-thirds of the lost sales are still expected to occur. This is because in some stations the difference between the demand for bicycles and lockers at certain hours of the day is too large to be buffered by the station inventory. For these stations, dynamic repositioning during the day is required. Alternatively, additional lockers at some stations (or even relocating the existing ones) could reduce the expected number of shortages. In that sense, our mathematical models provide a what-if analysis tool that may allow to rationally consider the potential benefit of adding lockers at the stations and/or relocating existing ones.

Overall, the AI2 formulation yielded good solutions to instances with 104 nodes and two vehicles and thus may be a good tool to provide to operators of bike-sharing systems. For larger systems, we believe that a practical approach is to decompose the entire area into several geographical areas, such that each area is served by one or two vehicles. In such a setting, we recommend using a common centrally located depot to allow transfers of bicycles between different areas. Such an approach is practical and easier to manage since it allows appointing an area manager, responsible for a small number of drivers who become familiar with their area. It is, in fact, a common practice in many distribution systems consisting of a large number of nodes. When such a decomposition approach is used, our methods are likely to solve the resulting separate sub problems to optimality or near optimality. A related question is how to decompose the system into such separate areas, but this is beyond the scope of this paper.

Conclusions and discussion

This paper defines and formulates a new rich inventory routing model that is motivated by the need to regulate the newly emerging bike-sharing systems. Proper regulation of these systems by repositioning bicycles among the stations is an important factor in their success. This paper considers many aspects of the static repositioning problem, in particular the stochastic and dynamic nature of the demand. We formulate the problem as a MILP, address various technical obstacles that arise in solving large instances, and analyze the results obtained. We offer two, essentially different formulations, based on different modeling assumptions. In both models, the major term in the objective function is the expected cost of events in which the service can not be provided, and a second component is added, which refers to the operating costs, both to be minimized. This objective function is non-linear, thus we rely on previous convexity results that enable us to linearize it in an exact manner.

The AI formulation typically yields better solutions than the TI formulation under the CPU time limitations of 2 h. However, we note that the feasible set of solutions of the AI model is smaller. The hardest constraint imposed by the AI model is that the stations are allowed to be visited only once by each vehicle. This limitation may exclude the best solutions in some cases, e.g., when the difference

between the initial inventory level and the desired one at some stations is large relative to the capacity of the vehicle. A simple modification of the AI formulation to mitigate the above shortfall may be to carry an additional index for most decision variables, which keeps track of the visit number of the vehicle to the station. This would clearly make it harder to solve the AI formulation.

Some extensions of the repositioning problem are easy to implement with our mathematical models. For example, one may argue that a fixed amount of time should be added every time the vehicle stops at a station, in addition to the linear loading and unloading times that are already included in our model. This can be handled by adding the fixed time to the travel times between stations. Note, however, that then it will not be possible to use the arc deletion method as is. Hence, this extension of our TI (and TI2) model requires introducing a new set of binary variables denoting actual stops at stations (rather than passing through them). One binary variable is required for each period and for each vehicle.

Proceeding to a higher level of the system design, an important extension of our models may be the inclusion of several depots in the system. We believe that the operator may benefit from using several stations, located where space is not costly, as secondary depots. These stations may be used to reduce the total distance travelled by the repositioning vehicles and serve as a local buffer to meet fluctuating demand during peak times. In fact, the inclusion of such an extension in the TI model only requires a change in the parameters (station capacities and penalty functions). The AI and AI2 models can accommodate multi-depots with minor changes. A similar related extension allows each vehicle to set its base node at a different location.

We demonstrate that our various MILP formulations are capable of solving problem instances of a moderate size of up to 60 stations with acceptable optimality gaps. The AI2 formulation can be used to solve realistic instances with two vehicles and 104 stations. However, larger systems such as Vélib, Bixi (in Montreal) or Bicing (in Barcelona) consist of many hundreds of stations. Our method is applicable to such systems when used in conjunction with a geographical decomposition approach. Another possible approach is to use heuristic methods such as Tabu search or genetic algorithm that exhibited good performances in other large-scale rich routing problems, see, e.g., Bräysy and Gendreau (2005).

An important factor in the successful implementation of static repositioning is the accuracy of the demand forecast. The demand faced by a bike-sharing system is highly sensitive to partially predictable effects such as weather conditions, public events in the city, etc. A reliable approximation of the penalty function is based on a good estimation of the demand pattern. In this study, we based our demand forecast on past demand data on similar days. However, more sophisticated data mining models may be devised in order to identify such "similar days" with respect to each forthcoming one.

As with any large-scale logistic plan, the static repositioning plan itself is subject to unpredictable events such as traffic slowdown, vehicle breakdown, etc. Future studies on the SBRP should devise methods to create a repositioning plan that is robust with respect to such unfortunate events. As for now, we recommend using the

models proposed in this paper by applying some safety margins, e.g., in terms of the time allotted to the repositioning task.

Finally, we focused in this paper on the static repositioning problem, where bicycles are moved during slack hours when the system is nearly inactive. As mentioned in the introduction, static repositioning helps reduce the amount of work required for dynamic repositioning. In cases where the capacity of the stations and the number of bicycles in the system are both large enough, static repositioning may replace the need for dynamic repositioning altogether, except maybe in unusual events (e.g., a concert in the park).

The dynamic version of the problem requires a different solution approach. In that case, the rentals (resp., returns) in each station decrease (resp., increases) its inventory level due to the users' activity at the same time the repositioning is carried out. Since the TI model already uses decision variables which keep track of the inventory level at each station in each time period (s_{it}), the corresponding inventory-balance constraints (constraints (27)) may be adjusted to reflect the dynamic situation. This can be achieved by subtracting from the right-hand side of (27), the forecasted net demand, and adding variables that represent unsatisfied demand for bicycles and lockers or waiting time of users at the stations. The objective function should be modified to represent the cost incurred by these "lost sales" or waiting times of users. While other solution approaches may be more appropriate for the dynamic version of the problem, the solution of a linear integer programming formulation may be useful as a benchmark.

Acknowledgments The authors wish to thank Mr. Brodie Hylton and Mr. Danny Quarrel from Alta Bicycle Share Ltd. for providing data on the Capital Bikeshare and Mr. Edison Avraham for help in processing the demand data. The research of the first two authors was partially supported by ISF grant no. 1109/11.

Appendix A—Computing the expected number of shortages

For the completeness of our paper, we summarize here some of the results of Raviv and Kolka (2012). In particular, we formally define the expected number of shortages function and show how to approximate it. We present this function in its general form, i.e., when the expected number of shortages for bicycles and for lockers may have different weights.

We consider a single bike-sharing station over a finite horizon $(0,\ T)$ with the following setting: the inventory level (number of bicycles) in the station at time 0 is given. At time $t \in (0,\ T)$, users who wish to rent (resp., return) a bicycle arrive at the station according to some non-homogenous Poisson process with rate μ_t (resp., λ_t). If the requested service can be provided right away, the bicycle is rented or returned and the inventory level is updated accordingly. If the service cannot be provided (i.e., due to an empty station for a renter or a full one for a returner) then the user abandons the station. In reality, the abandoning user may decide to seek the service in other stations of the system but this is out of the scope of this model. Thus, there are two undesirable types of events that may occur in a station. We assume that the system is penalized for each one of them.

p Penalty charged for each renter that abandons due to a shortage of bicycles
h Penalty charged for each returner that abandons due to a shortage of vacant lockers

The state of the station is a bounded birth and death process with birth rate μ_t and death rate λ_t. The process is depicted as a Markov Chain in Fig. 4.

Let $\pi_{ij}(t)$ denote the probability of the station being at state j at time t provided that its initial state at time 0 was i. It is possible to express the expected shortage function $F(x)$, where x denotes the initial inventory level (of bicycles) as follow:

$$F(x) \equiv \int_0^T (\pi_{x0}(t)p + \pi_{xC}(t)h)dt.$$

The first summation term in the integral represents the expected shortages accumulated due to abandonments of renters and the second due to abandonments of returners. For the penalty function used in the numerical section of this paper it was assumed that $p = h = 1$. Thus, the value of the expected shortage function represents the expected number of unsatisfied users during the period $(0,\ T)$.

Next, we note that while the arrival rates of renters and returners are non-homogenous over time, it is reasonable to assume that these rates vary in a finite number of steps over the planning horizon, say every 15–30 min. Now, the strategy to estimate $F(x)$ is as follows: the planning horizon is discretized into short periods, each of length δ, say $\delta = 1$ min. These periods are indexed by θ. For each period θ we evaluate the transition probability matrix from the beginning of the period to its end. We denote this transition probability by P_θ. The transition probability matrix from time 0 to time $t = \tau \cdot \delta$ is given by.

$$\pi(t) = \prod_{\theta=1}^{\tau} P_\theta.$$

P_θ can be calculated numerically, for more details see Ross (2010) Sect. 6.8. Based on the value of $\pi(t)$ we can obtain a reliable approximation of the expected shortage function using the following discretization procedure:

$$F(x) \approx \delta \sum_{\theta=0}^{\frac{T}{\delta}-1} (\pi_{x0}(t + 0.5\delta)p\mu_{\delta\theta} + \pi_{xC}(t + 0.5\delta)h\lambda_{\delta\theta}). \tag{46}$$

Raviv and Kolka (2012) also established bounds for the estimation error and showed that for a discretization level of 1 min the expected shortage function can be estimated very accurately in a fraction of a second for each of the stations in

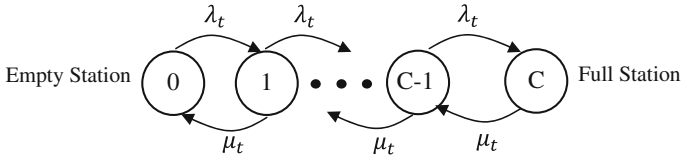

Fig. 4 Continuous time Markov chain that represents the dynamics of the bicycles inventory level

Washington's bike-sharing system. They also proved that the expected shortage function is convex.

The objective function in this paper is the sum of penalty functions, one for each station. The function for each station i is evaluated through constraints (46) for each possible inventory level $0 \leq x \leq c_i$.

For the instances that are based on real demand data from Velib and Capital Bikeshare, we estimated the demand for each half hour time slot at each station based on several working days with similar characteristics. We used the following parameters for the penalty function created for our experiment:

- $\delta = 1/60$ h. The experimentation in Raviv and Kolka (2012) shows that such a time granularity provides a very accurate approximation of the true function.
- $T = 18$ h. This represents roughly the time in which the systems in Paris and Washington are significantly active. The static repositioning is carried out during the remaining 6 h of the day, when the system is assumed idle. Empirically, we see that this is indeed the case for most of the stations in the system.
- $p = h = 1$. We assumed in our experiments that the penalties for bicycle shortage and locker shortage are identical. We are not sure that this is the case in the view of the operators. Some of them may assign a higher cost for a locker shortage since it puts the user in a very inconvenient situation. However, methodologically this should not make a difference.

Appendix B—Sequence index (SI) formulation

As motivated and described in "Additional formulations", we present here the sequence-Index (SI) formulation.

The following parameters need to be defined and used in this formulation:

A Upper bound on the number of stops on each vehicle's route. A can be safely set to be $T/\min_{i \neq j \in N_0} t_{ij}$, but can be chosen to be slightly lower with some careful consideration

$\bar{T}_j \equiv \max_{i \in N_0} t_{ij}$ The longest traveling time to node j from all nodes

The decision variables are defined as follows:

x_{iav} Binary variable which equals one if the ath stop of vehicle v is at node i, and zero otherwise

y^L_{iav} Number of bicycles loaded at node i onto vehicle v on its ath stop

y^U_{iav} Number of bicycles unloaded at node i out of vehicle v on its We first present the$a$th stop

r_{av} Time of completing loading/unloading at the ath stop of vehicle v

y_{av} Number of bicycles carried by vehicle v after loading/unloading at its a^{th} stop

s_i Inventory level at station i at the end of the repositioning operation

We first present the SI formulation when $\alpha = 0$. The case of $\alpha > 0$ requires a slight adaptation which is described just below this formulation.

(P3)—Sequence indexed (SI) formulation ($\alpha = 0$).

$$Min \sum_{i \in N} g_i \tag{47}$$

s.t.

$$g_i \geq a_{iu} + b_{iu} s_i \quad \forall i \in N, u = 0, \ldots, c_i - 1 \tag{48}$$

$$s_i = s_i^0 - \sum_{v \in V} \sum_{a} (y_{iav}^L - y_{iav}^U) \quad \forall i \in N \tag{49}$$

$$\sum_{i \in N} x_{iav} = 1 \quad \forall a = 1, \ldots, A, \forall v \in V \tag{50}$$

$$x_{01v} = 1 \quad \forall v \in V \tag{51}$$

$$x_{0Av} = 1 \quad \forall v \in V \tag{52}$$

$$r_{av} \geq r_{a-1v} - \bar{T}_j (1 - x_{jav}) + \sum_{i \in N_0} t_{ij} x_{i,a-1,v} + \sum_{i \in N_0} (L y_{iav}^L + U y_{iav}^U) \quad \forall j \in N_0, \forall a \geq 2, \forall v \in V \tag{53}$$

$$r_{1v} = L y_{01v}^L \quad \forall v \in V \tag{54}$$

$$r_{Av} \leq T \quad \forall v \in V \tag{55}$$

$$\sum_{a,v} y_{iav}^L \leq s_i^0 \quad \forall i \in N_0 \tag{56}$$

$$\sum_{a,v} y_{iav}^U \leq c_i - s_i^0 \quad \forall i \in N_0 \tag{57}$$

$$y_{iav}^L \leq \min(k_v, s_i^0) x_{iav} \quad \forall i \in N_0, \forall a = 1, \ldots, A, \forall v \in V \tag{58}$$

$$y_{iav}^U \leq min(k_v, c_i - s_i^0) x_{iav} \quad \forall i \in N_0, \forall a = 1, \ldots, A, \forall v \in V \tag{59}$$

$$y_{av} = y_{a-1v} + \sum_{i \in N_0} (y_{iav}^L - y_{iav}^U) \quad \forall a = 1, \ldots, A, \forall v \in V \tag{60}$$

$$y_{av} \leq k_v \quad \forall a = 1, \ldots, A, \forall v \in V \tag{61}$$

$$y_{0v} = 0 \quad \forall v \in V \tag{62}$$

$$y_{Av} = 0 \quad \forall v \in V \tag{63}$$

$$y_{iav}^L \geq 0, y_{iav}^U \geq 0, \text{ integers} \quad \forall i \in N, \forall a = 1, \ldots, A, \forall v \in V \tag{64}$$

$$x_{iav} \in \{0, 1\} \quad \forall i \in N, \forall a = 1, \ldots, A, \forall v \in V \tag{65}$$

$$y_{-av} \geq 0 \quad \forall a = 1, \ldots, A, \forall v \in V \tag{66}$$

$$s_i \geq 0 \quad \forall i \in N \tag{67}$$

The objective function (47) is identical to the one in the previous formulations. Constraints (48) are the linear constraints that support the convex penalty function,

defined with respect to the final inventory at each station, which is the inventory after loading/unloading during all visits. Constraints (49) define the inventory at the stations at the end of the repositioning operation. Constraints (50) make sure that each vehicle stops at one station at a time and constraints (51) [resp., (52)] verify that all vehicles depart from (resp., return to) the depot at the beginning (resp., end) of the repositioning operation. Constraints (53) define the minimal time required to reach node j which is the ath stop, when the $(a-1)$th stop is at node i. The minimal time includes, beyond the time to reach node i, the loading/unloading times at node i and the travel time between i and j. Constraints (54) define the time in which vehicle v leaves its first stop (the depot) to be the completion time of loading there (note that the vehicle cannot unload at the first stop which is the depot). Constraints (55) restrict the last visit of each vehicle to occur no later than time T. Constraints (56)–(57) limit the quantity picked-up from a node (resp., delivered to a node) in all stops of all vehicles to the quantity available there initially (resp., the remaining capacity of the station), as limited in the AI formulation. These constraints make sure that the capacity of a node is not exceeded and the inventory at the node is non-negative. Constraints (58)–(59) allow loading and unloading of bicycles at a certain node only when a vehicle stops at that node. Constraints (60) are inventory-balance constraints on each vehicle after each stop while constraints (61) are vehicle capacity constraints. Constraints (62) [resp., (63)] make sure that each vehicle is empty before it reaches the first stop (resp., after visiting the last stop, which is the depot). This stipulates that the total loading and unloading quantities are equal to each other. Finally, (64)–(65) are integrality constraints and (66)–(67) are non-negativity constraints.

We note that if there is only one vehicle, there is no need for synchronization among vehicles and hence constraints (56)–(57) can be replaced by: $0 \le s_i^0 - \sum_{a \le A'}(y_{ia1}^L - y_{ia1}^U) \le c_i \forall A' = 1, \ldots, A$ and the right-hand side of constraints (58)–(59) should be replaced by $min(k_v, c_i)x_{iav}$ in order to allow unlimited transshipment.

Several valid inequalities were developed and found to be helpful in speeding-up the running time of the above formulation, see Raviv et al. (2012). Nevertheless, as mentioned above, this formulation was found to be somewhat inferior to the two formulations mentioned in "Model formulation", so we shall omit the details of the valid inequalities.

When $\alpha > 0$, additional decision variables are needed, to keep track of the arcs traversed by the vehicles, so that the travel cost of the time spent by the vehicles is accounted for. To that end, we define:

Z_{av}= time spent by vehicle v in its ath segment.

The objective function becomes:

$$Min \sum_{i \in N} g_i + \alpha \sum_{a=1}^{A} \sum_{v \in V} Z_{av}$$

and the following set of constraints is added:

$$Z_{av} \ge t_{ij}(x_{iav} + x_{ja+1v} - 1) \forall a = 1, \ldots, A, \forall v \in V, \forall i \in N_0, \forall j \in N_0.$$

References

Anily S, Hassin R (1992) The swapping problem. Networks 22:419–433

Anily S, Gendreau M, Laporte G (1999) The swapping problem on a line. SIAM J Comput 29(1):327–335

Benchimol M, Benchimol P, Chappert B, Taille ADL, Laroche F, Meunier F, Robinet L (2011) Balancing the stations of a self service "bike hire" system. RAIRO Oper Res 45:37–61

Berbeglia G, Cordeau J-F, Gribkovskaia I, Laporte G (2007) Static pickup and delivery problems: a classification scheme and survey. TOP 15:1–31

Bertazzi L, Savelsbergh M, Speranza MG (2008) Inventory routing. In: Golden B, Raghavan S, Wasil E (eds) The vehicle routing problem: latest advances and new challenges. Springer, Berlin

Bordenave C, Gendreau M, Laporte G (2010) Heuristics for the mixed swapping problem. Comput Oper Res 38:108–114

Bräysy O, Gendreau M (2005) Vehicle routing problem with time windows. Part II: metaheuristics. Transp Sci 39(1):119–139

Brussel N (2010) Villo-bevoorrading lijdt onder files (in Dutch). Web newspaper. Publication date: 22 June 2010, Observed: 15 August 2012. http://www.brusselnieuws.be/artikel/villo-bevoorrading-lijdt-onder-files-update

Callé, E. (Director of Operation in Vélib), April 2009 (Personal communication)

Chalasani P, Motwani R (1999) Approximating capacitated routing and delivery problem. SIAM J Comput 28(6):2133–2149

Chemla D, Meunier F, Wolfler Calvo R (2011) Bike hiring system: solving the rebalancing problem in the static case. Discrete Optimization (accepted)

Contardo C, Morency C, Rousseau L-M (2012) Balancing a dynamic public bike-sharing system. https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2012-09.pdf

DeMaio P (2009) Bike-sharing: history, impacts, models of provision, and future. J Public Transp 12(4):41–56

Erdoğan G, Laporte G, Calvo RW (2012) The One-Commodity Pickup and Delivery Traveling Salesman Problem with Demand Intervals. Working paper

Forma I, Raviv T, Tzur M (2010). The static repositioning problem in a bike-sharing system. In: Proceeding of the 7th triennial symposium on transportation analysis (TRISTAN), Tromsø, Norway, pp 279–282

Fricker C, Gast N (2012). Incentives and regulations in bike-sharing systems with stations of finite capacity. http://arxiv.org/pdf/1201.1178v1.pdf

Hampshire R, Marla L (2011) An empirical analysis of bike-sharing systems. Working paper

Hernández-Pérez H, Salazar-González J–J (2004a) A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. Dis Appl Math 145:126–139

Hernández-Pérez H, Salazar-González J–J (2004b) Heuristics for the one-commodity pickup-and-delivery traveling salesman problem. Transp Sci 38:245–255

Lin J-R, Yang T-H (2011) Strategic design of public bicycle sharing systems with service level constraints. Transp Res Part E 47(2):284–294

Louveaux F, Salazar-González J–J (2009) On the one-commodity pickup-and-delivery traveling salesman problem with stochastic demands. Math Prog Ser A 119:169–194

Miller CE, Tucker AW, Zemlin RA (1960) Integer programming formulations and traveling salesman problems. J ACM 7:326–329

Mukai N, Watanabe T (2005) Dynamic location management for on-demand car sharing system, knowledge-based intelligent information and engineering systems. In: 9th international conference, KES 2005, Melbourne, Australia, pp 768–774

Nair R, Miller-Hooks E (2011) Fleet management for vehicle sharing operations. Transp Sci 45(4):524–540

Raviv T, Kolka O (2012) Optimal inventory management of a bike-sharing station. IIE Trans (To appear). http://dl.dropbox.com/u/717696/Home%20Page/Publications/Optimal%20Inventory%20Management%20of%20a%20Bike-Sharing%20Station.pdf

Raviv T, Tzur M, Forma I (2012) Static repositioning in a bike-sharing system: models and solution approaches. Unabridged version, Industrial Engineering department, Tel Aviv University, Israel. https://dl.dropbox.com/u/717696/Home%20Page/Publications/Static%20Repositioning%20-%20unabridged%20version.pdf

Rosing KE, Revelle CS (1997) Heuristic concentration: two stage solution construction. Eur J Oper Res 97(1):75–86

Ross SM (2010) Introduction to probability models. Academic Press, Amsterdam
Shaheen S, Guzman SY (2011) Worldwide Bikesharing. Access 39:22–27
Shu J, Chou M, Liu Q, Teo C-P, Wang I-L (2010) Bicycle-sharing system: deployment, utilization and the value of re-distribution. Working paper. http://bschool.nus.edu/Staff/bizteocp/BS2010.pdf
Tusia-Cohen M (2012) Invented the wheel: first year of Tel-O-Fun (in Hebrew), Web version of the daily newspaper Maariv. Publication date: 12 April 2012, Observed at 26 August 2012. http://www.nrg.co.il/online/1/ART2/357/098.html
Uesugi K, Mukai N, Watanabe T (2007) Optimization of vehicle assignment for car sharing system. In: Lecture notes in artificial intelligence, intelligent knowledge: knowledge-based intelligent information and engineering systems, pp 1105–1111
Vogel P, Mattfeld DC (2010) Modeling of repositioning activities in bike-sharing systems. In: Proceeding of the 12th world conference on transport research, 11–15 July 2010, Lisbon