

THE BIKE SHARING REPOSITIONING PROBLEM WITH FIXED ROUTES AND TIME WINDOWS

Les R. Foulds, Humberto J. Longo

Instituto de Informática – Universidade Federal de Goiás
Alameda Palmeiras, Quadra D, Campus Samambaia, 74001-970, Goiânia, GO, Brasil
{lesfoulds, longo}@inf.ufg.br

Marcus V. S. Poggi de Aragão

Departamento de Informática – Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente 225, Gávea, 22451-900, Rio de Janeiro, RJ, Brasil
{poggi}@inf.puc-rio.br

RESUMO

Este artigo aborda um problema na área de transporte urbano, o de lidar com o reposicionamento de bicicletas em sistemas públicos de compartilhamento de bicicletas. Estudamos um caso especial desta operação, onde são conhecidas a priori as rotas dos veículos e os horários disponíveis para a realização da operação, o chamado Problema de Reposicionamento de Bicicletas em um Sistema de Compartilhamento sujeito a Rotas Fixas e Janelas de Tempo (*BSRP/FRTW*). Uma formulação de programação inteira do *BSRP/FRTW* é proposta. Estratégias compatíveis com as práticas atuais para problemas semelhantes de coleta e entrega são apresentadas. Uma abordagem *branch-and-price*, com a resolução do subproblema de geração de colunas por programação dinâmica também é descrita.

PALAVRAS CHAVE. Compartilhamento de bicicletas, redistribuição, programação inteira, *branch-and-price*, geração de colunas, programação dinâmica.

PM – Programação Matemática, L&T – Logística e Transportes, OC – Otimização Combinatória.

ABSTRACT

This paper addresses a problem in the area of urban transportation, that of dealing with the repositioning of bikes in public bike sharing systems, a notoriously difficult pick-up and delivery vehicle routing problem. We study a special case of this operation, which we call the Bike Sharing Repositioning Problem with Fixed Routes and Time Windows (*BSRP/FRTW*). An integer programming formulation of the *BSRP/FRTW* is proposed. Strategies compatible with current practices for similar pickup and delivery problems are presented. A column generation approach based on a branch-and-price scheme that is solved by dynamic programming is described. Preliminary computational experience (to be reported in the future) is promising.

KEYWORDS. Bike sharing, repositioning, integer programming, column generation, *branch-and-price*, dynamic programming.

PM – Mathematical Programming, L&T – Logistics and Transport, OC – Combinatorial Optimization

1. Problem description

The shared use of public bicycles, an increasingly popular alternative to the use of private cars, can be organized in Bike Sharing Systems (BSSs). A BSS makes it possible for members of the public to use shared bikes for personal travel within an urban area. The terms *fixed bikesharing* and *free floating bikesharing* have been around since the mid-2000s and accurately describe station-based and non-station-based systems, respectively. A fixed BSS has stations at fixed points and bikes must be picked up and dropped off at those stations. A free floating BSS provides flexibility to customers about where to drop off a bike after use (Liu et al. [2018]). This paper focuses on only fixed BSSs.

A traditional fixed BSS consists of a network of stations, each with a number of bikes available to be rented by members of the public. As most people almost always follow the same routine of origin-destination travel each working day, generally, the network of stations required for a good geographic coverage is not very wide. If the bike demand pattern is known, which usually varies throughout the day, the appropriate number of bikes can be provided at the BSS stations for the effective service of the BSS users. Usually bikes are rented for short periods of time and the BSS users can choose the pickup and drop off stations, within the bike-sharing network, without interference from a system manager. In fact, bikes are often dropped off at different stations from their pickup stations, creating an imbalance between supply and demand at certain stations, a major challenge. Hence, special light vehicles must be used to pick up bikes from stations in surplus and transport them to stations in deficit in order to satisfy a particular service level.

The first public BSS, using standard bikes, was introduced in Amsterdam in 1965. Unfortunately, the system was sabotaged by theft and vandalism and was withdrawn within a week. After a number of small-scale experiments in various European cities, a major, improved second generation BSS, with specially designed bikes, was begun in Copenhagen in 1995. However, users still remained anonymous and the system was again hampered by bike theft. The issue of user anonymity was addressed via a series of third generation BSSs. The first of these, at the University Portsmouth, introduced a swipe card, which enabled better user tracking. In 2005, the first large-scale fourth generation BSS (called Velov) with 1,500 bikes was initiated in Lyon. With 15,000 users and bikes being used an average of 6.5 times each day, the system quickly commanded significant interest. However, the Velib BSS launched in Paris in 2007 turned out to be a major breakthrough in BSS design and operation. In its first year the system had 20,000 bikes, 1,500 stations, 200,000 users and over 26 million bike rentals. Velib quickly mushroomed and its huge success was a major influence in the introduction of further BSSs worldwide. China has been a major centre of BSSs since their introduction there in the early 2000s. The city of Hangzhou, for instance, initiated a BSS in 2008 and by September 2016 it operated with over 3,600 stations and more than 84,000 bicycles, which were able to serve up to 473,000 users per day. According to Ma [2017], there were more than 170 BSSs in China by the end of 2016. Modern BSSs have now become prevalent throughout the world due, in part, to improvements in information and communications technology, enabling better user identification and tracking, automatic billing and monitoring, electronic locking and releasing of bikes, cashless payment, GPS, mobile phone access, and on-board smart systems. Nowadays, there are currently over 1,500 cities worldwide that have BSSs (some with multiple systems), involving millions of bikes in total.

Alvarez-Valdes et al. [2016] define three phases in the design and operation of a traditional, fixed BSS. The authors suggest that the first phase involves identifying the number of stations and their locations, the second phase is concerned with determining the number of bikes in the system, and the purpose of the final phase is to construct a bike repositioning system for

transporting bikes from stations in excess to stations in deficit. Clearly, the phases are highly interdependent. Furthermore, user perceptions of the convenience of a particular station are usually strongly influenced by its location and the perceived chances of a bike being available when needed which, in turn, is highly correlated to the number of bikes assigned to the station. Thus, in order to identify meaningful output, it might be beneficial to run iterations of the three phases interactively. Normally, the operation of a fixed BSS includes determining the optimal routes to be travelled by the vehicles and the loading or unloading level of each vehicle at the stations based on given design objectives. These decisions are subject to various operational, bike repositioning and station constraints. This is called the Bike Sharing Repositioning Problem (*BSRP*) [Vogel e Mattfel, 2010]. This problem is more complicated than the classical pickup and delivery vehicle routing problem because the pickup and drop-off levels at each station must also be determined [Ho e Szeto, 2017]. The vehicles can be used to perform either static or dynamic repositioning operations. The static operation involves repositioning the bikes when the BSS is closed to the public and station demand is invariant, typically at night or dawn. The dynamic operation involves repositioning the bikes throughout the working day while the BSS is in operation with real-time station demand variations. This necessitates the regular updating of vehicle routes to address station demand variations as they occur during the day.

There is a significant number of reports in the research literature on BSSs including the surveys by DeMaio [2009] and Laporte et al. [2015]. Most BSS studies can be classified under five main topics: station location, station inventory, rebalancing incentives, fleet dimensioning (the number of bikes deployed) and vehicle repositioning (Laporte et al. [2015]). The present paper focuses on fleet dimensioning and vehicle repositioning. Existing BSS reports have adopted various objectives, such as minimizing total vehicle travel cost (Chemla et al. [2013]), minimizing total unmet demand (Shui e Szeto [2018]), minimizing maximum tour length (Schuijbroek et al. [2017]), minimizing the sum of travel and handling costs (Erdoğan et al. [2015]), minimizing the total relocation and user cost (Caggiani e Ottomanelli [2013]) and minimizing weighted sums of various of the above objectives [Rainer-Harbach et al., 2015]. However, the most commonly-used objective is to minimize the total absolute deviation from perfect balance in the BSS, either by directly determining the number of bikes or indirectly in the form of penalty functions (Shui e Szeto [2018]). Here we concentrate on determining the number of bikes. Çelebi et al. [2018] developed a mathematical model and a solution method for determining the number and location of the required stations of a fixed BSS by taking into account given service levels. The next section presents a similarly constrained model that aims instead to minimize the total number of bikes in the system.

Here we study a special case of the static repositioning problem, which we call the Bike Sharing Repositioning Problem with Fixed Routes and Time Windows (*BSRP/FRTW* for short.) This problem consists of determining the repositioning operation to be performed by each available vehicle that periodically travels through a fixed sequence of stations, i.e. a fixed route, picking up and dropping off bikes, as will be explained in more detail later. Two objectives to improve the performance of a BSS as a whole can be immediately proposed: (i) determine the repositioning operation that satisfies a given level of station demand and minimizes the total number of bikes in the system; or (ii) given a fixed number of bikes (and their distribution at the stations), find the repositioning operation that minimizes the number of unmet bike demands.

In general, transport operations of the kind described above involve the subproblem of finding the routes to be traveled by the vehicles. However, finding such routes is only part of the problem to be solved. In reality, a common rule of organizations that perform the pick up or delivery operations of some kind of good is that the vehicle must repeat the same sequence of visits,

that is, repeat a fixed route. Therefore, here we assume that the transport routes have already been given, leaving open the decisions about when and what actions should be performed. Following this approach, the BSRP/FRTW repositioning operation consists of determining when and how many bikes must be picked up from or delivered to the stations by vehicles that run pre-defined routes. This approach has the advantage of being quite flexible, since it enables sets of mandatory rules and constraints to be associated with the pre-defined routes, and any activity not satisfying them will have high additional costs assigned to it. Examples of such rules are (i) that a route must always start and end at the same point (the vehicle depot), (ii) its duration must be limited, and (iii) the number of vehicles/drivers available for the transport operation is constrained.

This paper addresses only the first objective previously given (to determine the static repositioning operation that satisfies all station demand and minimizes the total number of bikes in the system), since this is more tactical in nature and therefore more plausible in practice. The above discussion allows us to define this problem more precisely, as follows:

BSRP/FRTW – Bike Sharing Repositioning Problem with Fixed Routes and Time Windows:

- **Input:** A road network and its characteristics (distances, travel times, etc.), a set of stations, a set of routes, the number of available vehicles and their respective capacities, a time horizon and the expected demand for bikes at each station, within the time horizon.
- **Output:** The minimum number of bikes, their initial distribution at the stations and the corresponding repositioning operation that satisfies all station demand.

The proposed BSRP/FRTW problem is static in the sense that it is possible to start the system daily, i.e., it is always possible to make an initial distribution of bikes between stations before releasing them to the BSS users. For this to be feasible, there must be sufficient time between the closure of the rental activities in any one day and the restarting of rental activities on the following day. (Consequently, the repositioning operation is often carried out during the night or at dawn).

2. BSRP/FRTW Modeling

In order to control the availability of bikes at stations within the time horizon considered, as well as the demands at each station, the time horizon is partitioned into subintervals. Thus, from an initial bike distribution at the stations at the beginning of the first time window, the repositioning operation can be defined by changes in the loads of the vehicles associated with the repositioning operation at each subsequent time window. Since a vehicle must use a predefined route to move bikes between the stations, the pick-and-drop actions can only be performed at the time windows in which the route passes through the stations. We assume that estimates of bike demand at the stations over the time horizon are given. In fact, from a usage log of the system, available statistical techniques can be used to estimate (with acceptable reliability) the station demand for the given time windows. Thus, the problem can be formally defined as follows:

BSRP/FRTW – Bike Sharing Repositioning Problem with Fixed Routes and Time Windows:

- K is the number of stations of the BSS; T is the number of windows into which the time horizon is partitioned; d_w is the duration of each time window; $G = (N, A)$ is an oriented graph that defines the street network (stations are represented by the nodes in N and the streets that join them by the arcs in A); $c_t(a)$, $\forall a \in A$, is the time required to traverse the path a ; m_t^k is the estimate of bike delivery (pickup, if m_t^k is less than zero) at station k and time window t , for $k = 1, \dots, K$ and $t = 1, \dots, T$; $\mathcal{S}$ is a set of routes such that each route $S \in \mathcal{S}$, with p stations, is defined by $S = (s_1, s_2, \dots, s_p)$ with a particular p for each route and $s_i \in N$, $i = 1, \dots, p$; and R is the number of vehicles (all with identical capacity C).

- The objective is to determine the minimum number of bikes, their initial distribution at the stations and the corresponding repositioning operation that satisfies all the demands.

A fixed BSS requirement is that at each time window at least the number of bikes that are estimated to be rented is available at each station. This number is given by the symmetric value of n_t^k , whenever this value is negative; or zero, otherwise. This is equivalent to requiring that each station k must contain at the beginning of the time window t a minimum number of m_t^k bikes, which will be available to be rented during time window t . It is also necessary to define M_t^k as the capacity of the station k during the time window t .

The BSRP/FRTW is now formulated as an integer linear programming problem. The variables specify the number of bikes at the stations during the time horizon considered and control the transport operation of bikes by vehicles that follow the given routes. The objective function minimizes the total number of bikes in the BSS. The resulting formulation (Model IP-BSRP/FRTW), with an exponential number of variables, follows:

$$\text{Minimize } z = \sum_{k=1}^K v_0^k, \quad (1)$$

subject to

$$v_{t-1}^k - v_t^k + \sum_{j \in J} (a_{tj}^k \cdot x_j) = 0, \quad k = 1, \dots, K; t = 1, \dots, T; \quad (2)$$

$$\sum_{j \in J} (b_{tj} \cdot x_j) \leq R, \quad t = 1, \dots, T; \quad (3)$$

$$\sum_{j \in J} (\langle a_1 \rangle_{tj}^k \cdot x_j) \leq 1, \quad k = 1, \dots, K; t = 1, \dots, T; \quad (4)$$

$$\sum_{j \in J} (\langle a_2 \rangle_{tj}^k \cdot x_j) \leq 1, \quad k = 1, \dots, K; t = 1, \dots, T; \quad (5)$$

$$x_j \in \{0, 1\}, \quad \forall j \in J; \quad (6)$$

$$m_t^k \leq v_t^k \leq M_t^k \quad k = 1, \dots, K; t = 1, \dots, T. \quad (7)$$

The stock of bikes at each station k at the beginning of the time window 1 is defined by the variables v_0^k and the goal is to minimize their sum. The variables v_t^k define the number of bikes at station k at the end of the time window t , for $k = 1, \dots, K$ and $t = 1, \dots, T$. The binary variables x_j indicate whether the repositioning operation j , defined for a specific route, is used or not.

The coefficient a_{tj}^k is the variation in the bike load of the vehicle which, in carrying out the repositioning operation j , passes through station k during time window t . That is, a_{tj}^k is the number of bikes delivered (picked up) at station k by the repositioning operation j , if the value of that coefficient is negative (positive). The coefficient b_{tj} is 1, if the repositioning operation j occurs during the time window t , or 0, otherwise; $\langle a_1 \rangle_{tj}^k$ is 1, if $a_{tj}^k < 0$, or 0, otherwise; and $\langle a_2 \rangle_{tj}^k$ is 1, if $a_{tj}^k > 0$, or 0, otherwise.

The set of constraints (2) specifies the conservation of the bike flow resulting from its repositioning over the stations, the set (3) specifies the maximum number of routes that can occur simultaneously in each time window and the sets (4) and (5) restrict, respectively, at each station and time window, the pickup or delivery operations of bikes to be performed by only one vehicle at a time. The set J , in the constraints sets (2) – (6), contains the indices that specify all possible valid bike transport operations, defined in the set $\mathcal{S}$ of routes considered.

The cardinality of the set J of all possible transport operations is exponential in the number of BSS stations and in the capacity of the vehicles, which requires the use of some implicit enumeration technique for identifying the variables x_j . The integrality of these variables suggests the use of a branch-and-price scheme if the goal is to find an optimal solution to the problem. Otherwise, a solution to the linear relaxation of the model provides a lower bound on the optimal solution value and a good starting point for routines based on rounding the variable values and for search heuristics.

In the following sections, a column generation approach is described for solving this formulation. The master problem considers estimates of demands and other constraints associated with the stations. The column generation subproblem, which is presented in Section 3, provides valid repositioning operations for the routes to be used by the given set of vehicles. Section 4 presents two dynamic programming algorithms which are used to find optimal values for the coefficients a and b for a given route. These values define when, where and how many bikes must be loaded or unloaded from a vehicle, that is, they define a valid repositioning operation for a route.

3. Column generation subproblem

Consider an optimal solution to the linear relaxation of the model *IP-BSRP/FRTW*, which considers only a subset $J_0 \subseteq J$ of columns (variables x_j). The subproblem in question consists of finding a column (variable) $j \in J \setminus J_0$ with a negative reduced cost $\bar{c}_j$ or proving that such a column does not exist. A column of the *IP-BSRP/FRTW* model represents a valid transport operation and is defined by the coefficients a in the constraints (2). The coefficients b , $\langle a_1 \rangle$ and $\langle a_2 \rangle$ that appear in constraints (3)–(5) are derived directly from coefficients a . Thus, given a route $S = \{s_1, \dots, s_p\}$, each coefficient $a_{tj}^{s_k}$ of a column $j \in J$ represents a pickup or delivery action of bikes at station k and time window t to be performed by a vehicle that traverses route S . All the column coefficients a that are null are relative to the stations and time windows not covered by the route S .

The segment of a column represented by the coefficients a defines a bike repositioning operation over the stations. This operation will be valid if each action (coefficient) $a_t^{s_k}$ respects the maximum capacity of the vehicle at each time window. Thus, the constraints $-C \leq a_t^{s_k} \leq C$, for $t = 1, \dots, T$ and $k = 1, \dots, p$; and $\sum_{k=1}^K \sum_{t'=0}^t a_{t'}^{s_k} \leq C$, for $t = 1, \dots, T$; must be satisfied. Furthermore, it does not make sense to perform two actions in sequence at the same station, i.e. a particular vehicle must not pickup (or deliver) bikes from a station and then repeat one of these actions at the same station. In addition, the set of actions between any two time windows must also respect the capacity of the vehicle involved, that is, the sum of the respective coefficients must belong to the interval $[-C, \dots, C]$.

Let w_t^k , u_t , $\langle w_1 \rangle_t^k$ and $\langle w_2 \rangle_t^k$ be the dual multipliers associated with the constraints (2), (3), (4) and (5) of the model *IP-BSRP/FRTW*, respectively. Thus, the column generation subproblem must verify, for a given route S , whether or not $\bar{c}_j < 0$. This value is defined as follows:

$$\bar{c}_j = - \sum_{k=1}^K \sum_{t=1}^T [(w_t^{s_k} \cdot a_t^{s_k}) + (\langle w_1 \rangle_t^{s_k} \cdot \langle a_1 \rangle_t^{s_k}) + (\langle w_2 \rangle_t^{s_k} \cdot \langle a_2 \rangle_t^{s_k})] - \sum_{t=1}^T (u_t \cdot b_t). \quad (8)$$

If there are no restrictions associated with the routes, this subproblem should only check if there is any column (valid schedule of repositioning operations for a route) with a reduced negative cost. The opposite indicates that an optimal solution for the linear relaxation of the model *IP-BSRP/FRTW* has been found. Section 4 describes two algorithms to find the optimal values for coefficients a , and consequently b , for a particular route. Thus, if several routes are given, one of these algorithms must be executed for each of them if the objective is to find the route with the best

transport operation schedule. The running time of this procedure can be very high. However, once a column (transport operation for a route) of negative cost has been found, the column generation algorithm can then be implemented.

4. Column generation by dynamic programming

In this section we describe two algorithms based on dynamic programming for the generation of columns for the model *IP-BSRP/FRTW*. The first assumes that a route $S = (s_1, s_2, \dots, s_p)$ will be traversed by a vehicle only once a day. In addition, its start and duration are unconstrained, i.e., it can be started in any time window (one that allows all stations to be visited within the considered time horizon) and the stops have no fixed duration. The second algorithm assumes that the route S will be traversed several times a day, with the other parameters (beginning and duration of the stops) pre-fixed.

We assume that a vehicle starts and ends a route $S = (s_1, s_2, \dots, s_p)$ without any load. Let $Q = \{0, 1, \dots, C\}$ be the set of all possible quantities of bikes transported by a vehicle, and $P(s_k, t, q)$ be the cost of the dynamic programming state when the load of the vehicle is q bikes at the end of the time window t , immediately after visiting the station s_k on route S . The initial state, which can be associated with an artificial station s_0 , is $P(s_0, 0, 0)$. The dynamic programming recurrence relation for the algorithms of column generation for the model *IP-BSRP/FRTW* is:

$$P(s_k, t, q) = \min_{\substack{q' \in Q \\ t' \in \mathcal{T}(s_k, t)}} \left\{ P(s_{k-1}, t', q') + \sum_{\ell=t'+1}^t u_\ell - (w_t^{s_k} \cdot (q - q') + \langle w_1 \rangle_t^{s_k} \cdot \varphi_1 + \langle w_2 \rangle_t^{s_k} \cdot \varphi_2) \right\}, \quad (9)$$

where φ_1 is 1, if $(q - q') < 0$ or 0, otherwise; φ_2 is 1, if $(q - q') > 0$, or 0, otherwise; $\mathcal{T}(s_k, t)$ is the set of time windows from which it is possible to exit the station s_{k-1} and arrive next at station s_k in the time window t ; and w_t^k , u_t , $\langle w_1 \rangle_t^k$ and $\langle w_2 \rangle_t^k$ are the dual multipliers associated with the constraints (2), (3), (4) and (5) of the model *IP-BSRP/FRTW*, respectively.

4.1. Fixed routes

Consider the case where any particular route S is traversed only once. Let $c_t(s_{k-1}, s_k)$ be the cost (time) required to travel directly from station s_{k-1} to station s_k and d_w be the duration of a time window. The first time window (i_{s_k}) in which the station s_k can be reached from the s_{k-1} is defined as:

$$i_{s_k} = \begin{cases} 1, & \text{if } k = 1; \\ i_{s_{k-1}} + \left\lceil \frac{c_t(s_{k-1}, s_k)}{d_w} \right\rceil, & \text{if } 2 \leq k \leq p. \end{cases} \quad (10)$$

Since the time horizon has been partitioned into T intervals, the vehicle can reach the station s_k , arriving from s_{k-1} , in any time window $t = i_{s_k}, \dots, T - i_{s_p} + i_{s_k}$. The upper limit of this window allows the other stations on the route to be visited in regular time. So, if t is the time window when the vehicle arrives at station k , we define the set $\mathcal{T}(s_k, t) = \{i_{s_k}, \dots, t\}$ as the set of time windows from which it is possible to exit the station s_{k-1} and arrive at station s_k in time window t . Note that the vehicle can arrive at station s_k in a time window prior to t and remain there until the window t .

The column generation algorithm is now given in detail:

Algorithm 1 – Fixed route

1 Initialization

Initialize $P(s_k, t, q)$, for all $k, t \in q$, with a high positive value and $P(s_0, 0, 0)$ with zero. Set k equal to 1.

2 Iteration.

Using the recurrence relation (9), calculate $P(s_k, t, q)$ for all $t \in \mathcal{T}(s_k, t) = \{i_{s_k}, \dots, T - i_{s_p} + i_{s_k}\}$ and all $q \in Q$. Increase k by 1. If $k \leq p$, repeat this step.

3 Optimal solution.

The value of the optimal solution is given by: $P^* = \min_{t=i_{s_p}, \dots, T} \{P(s_p, t, 0)\}$.

The values of the coefficients b_t and $a_t^{s_k}$ (hence $\langle a_1 \rangle_t^{s_k}$ and $\langle a_2 \rangle_t^{s_k}$), associated with the optimal solution, can be found by traversing the states (or decisions) retroactively from any state with the value P^* . This state represents the end of the route, when all bikes still being transported by the vehicle must be unloaded. As defined above, the arrival at the last station p can occur in any time interval between i_{s_p} and T .

4.2. Fixed routes with repetition

Assume now that the repositioning of bikes is carried out by vehicles that travel along a set of short routes, which cover all the stations and may be repeated several times over the time horizon considered. In addition, a route is assumed to be carried out within a rigid schedule, with predefined start and arrival times at each station and duration of the stops (for pickups and deliveries or a time gap between repetitions).

Thus, a route can now be defined as:

$$\sigma = (t_1, t_c, t_d, r, S = (s_1, s_2, \dots, s_p)), \quad (11)$$

where t_1 specifies the time window at the beginning of which the route S should be started; t_c is the number of time windows that the vehicle must remain at each station (while loading or unloading); t_d is the duration of the stop (the number of time windows) that the vehicle must make between two of the r repetitions of the route; and s_1 to s_p are the p stations to be visited, with $s_1 = s_p$. The total number of required time windows is denoted by W_S .

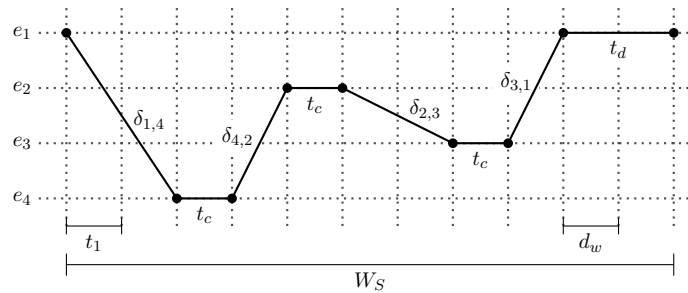


Figure 1: BSRP – route example.

Figure 1 depicts an example of a complete cycle of a route with four stations, which are labeled e_1, e_2, e_3 and e_4 . The route is $S = (s_1, s_2, s_3, s_4, s_5) = (e_1, e_4, e_2, e_3, e_1)$. The number of time windows required for the vehicle to travel from the station s_i to s_j is defined as

$\delta_{i,j} = \lceil c_t(s_i, s_j)/d_w \rceil$, where d_w is the duration of each time window. The total number of required time windows is $W_S = 11$.

Consider the first of the r repetitions of a route. The first time window (i_{s_k}) in which the station s_k can be reached from s_{k-1} is now fixed and can be defined as:

$$i_{s_k} = \begin{cases} t_1, & \text{if } k = 1; \\ t_1 + \left\lceil \frac{c_t(s_1, s_2)}{d_w} \right\rceil, & \text{if } k = 2; \\ i_{s_{k-1}} + \left\lceil \frac{c_t(s_{k-1}, s_k)}{d_w} \right\rceil + t_c, & \text{if } 3 \leq k \leq p. \end{cases} \quad (12)$$

Such a route must respect the maximum time limit available for its r repetitions. Thus, the total time windows required to complete a route S is given by $W_S = i_{s_p} + t_d - t_1$, which includes the time windows that compose the mandatory stop before another repetition of the route. This value must allow for the execution of the r repetitions of the route from a time window t_1 , that is, $t_1 + r \cdot W_S \leq T$, where T is the total number of time windows of the considered time horizon. This expression can be written in detail as follows:

$$t_1 + r \cdot \left[\sum_{k=1}^p \left\lceil \frac{c_t(s_{k-1}, s_k)}{d_w} \right\rceil + t_c \cdot (p - 2) + t_d \right] \leq T. \quad (13)$$

The cost $P(s_k, t, q)$ of a state of the recurrence relation refers to the cost of there being q bikes in the vehicle, at the end of the time window t , immediately after visiting the station s_k having arrived from station s_{k-1} . Note that now, from the second repetition of a route, the station preceding station s_1 is station s_p . This detail does not invalidate the recurrence relation, but it should be considered during the generation of the column.

In the specialization of the column generation algorithm, to consider the r repetitions of a route S , k' is the station prior to the arrival at the station k (in analogy to the time window t' and load q' precedent). Thus, when calculating the cost $P(s_k, t, q)$, the recurrence should assume the value $P(s_{k'}, t', q')$ and not $P(s_{k-1}, t', q')$. In the algorithm, detailed below, n is a counter for the number of repetitions of the route S :

Algorithm 2 – Fixed route with repetition

1 Initialization.

Initialize $P(s_k, t, q)$, for all k , t and q , with a high positive value and $P(s_0, 0, 0)$ with zero. Set k equal to 1, t equal to 0 and n equal to 0.

2 Iteration.

Set $\mathcal{T}(s_k, t) = \{t\}$ and, after, $t = n \cdot W_S + i_{s_k}$. If $n > 0$ and $k = 1$ set k' equal to p , otherwise set k' equal to $k - 1$. Using the recurrence relation (9), calculate $P(s_k, t, q)$ for all $q \in Q$. Increase k by 1. If $k \leq p$, repeat this step, otherwise increase n by 1.

3 Route repetition.

Set k equal to 1. if $n < r$, repeat the step 2.

4 Optimal solution.

The value of the optimal solution is given by: $P^* = P(s_p, t = r \cdot W_S + t_1 - t_d, 0)$.

5. Branch-and-Price

As observed in Section 2, the cardinality of the set J is exponential in the number of stations and in the capacities of the vehicles. The exact resolution of the *IP-BSRP/FRTW* model when implementing a branch-and-price algorithm requires a solution space partitioning scheme. This scheme should allow the exclusion of the fractional LP solution relative to the current node of the enumeration tree and not overcomplicate the column generation subproblem. This second requirement virtually excludes standard partitioning schemes based on setting the value of a variable to any integral value.

Regarding the model *IP-BSRP/FRTW*, suppose the pair $(V, X) = (\{v_t^k \mid k = 1, \dots, K; t = 0, \dots, T\}, \{x_j \mid j \in J_0\})$ is an optimal solution to a linear relaxation that considers only a subset $J_0 \subseteq J$ of columns. To set a variable $x_j \in X$ equal to 1 does not complicate the column generation subproblem, it merely reduces the size of the subproblem. However, setting a variable to 0 is equivalent to preventing the subproblem from generating a particular column. Thus, in certain stages of the enumeration of solutions, a large number of columns could be prevented from being provided by the subproblem. This impediment, in general, greatly complicates the resolution of the subproblem by the dynamic programming algorithms presented in Section 4.

One way of overcoming this drawback is to use a scheme based on fixing (as 0 or 1) the variables $v \in V$ and the slack variables $\langle s_1 \rangle_t^k$ and $\langle s_2 \rangle_t^k$ of the constraints (4) and (5), respectively. Doing this does not change the structure of the column generation subproblem. Setting a variable v_t^k , to any value v in the interval $[m_t^k, \dots, M_t^k]$, specifies that at the end of the time window t , the station k must contain v bikes. On the other hand, setting a slack variable $\langle s_1 \rangle_t^k$ to its maximum (or minimum) value determines that at most one vehicle can stop at station k in the time window t for delivery. In relation to pickups the reasoning is identical for the variables $\langle s_2 \rangle_t^k$.

Therefore, after identifying a slack variable $\langle s_1 \rangle_t^k$ (or $\langle s_2 \rangle_t^k$) that is fractional, the pair $(\langle s_1 \rangle_t^k, \langle s_2 \rangle_t^k)$ can be set to $(0, 1)$, $(1, 0)$ or $(1, 1)$. The configuration $(0, 0)$ does not need to be considered, as it would make the model *IP-BSRP/FRTW* infeasible. This is because the coefficients of the constraints (4) and (5) are defined, respectively, as:

$$\langle a_1 \rangle_{tj}^k = \begin{cases} 1, & \text{if } a_{tj}^k < 0; \\ 0, & \text{otherwise;} \end{cases} \quad (14)$$

and

$$\langle a_2 \rangle_{tj}^k = \begin{cases} 1, & \text{if } a_{tj}^k > 0; \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

By setting the pair $(\langle s_1 \rangle_t^k, \langle s_2 \rangle_t^k)$ as $(1, 0)$, the column generation algorithms described in Section 4 should consider the state $P(k, t, q)$ when $(q - q') \geq 1$. This is equivalent to preventing the vehicle from stopping at station k , in the time window t , to deliver any of the q' bikes that it is carrying. If the pair $(\langle s_1 \rangle_t^k, \langle s_2 \rangle_t^k)$ is set to $(0, 1)$, the state $P(k, t, q)$ is deleted when $(q - q') \leq -1$. Thus, the vehicle is prevented from stopping at station k , in the time window t , to pick up any number of bikes (the maximum is $(q' - q)$) up to its maximum load. Finally, the fixing as $(1, 1)$ prevents pickup or delivery at station k during the time window t . In this case, the practical effect is the elimination of the station k from the route.

This scheme does not guarantee the optimality of the best solution found, since even a solution in which such variables have integral values, some variable $x_j \in X$ can still be fractional. In fact, there is no guarantee that an integer solution will ever be found. So, in principle this scheme

needs to be combined with other options. However, the scheme substantially reduces the search space, which should be sufficient to ensure that practical solutions are found.

The computational effort in generating columns throughout the branch-and-price scheme can be reduced. The objective is to find useful solutions in the search nodes in which the variable $x_j \in X$ has a fractional value and the others have integer values. To achieve this, at the end of the execution of the algorithm, an integer linear programming problem can be constructed with all the columns generated during the execution of the branch-and-price routine. This alternative approach improves the chances of finding a practical solution for a particular instance.

6. Conclusion

The subject of this study is a problem in the area of urban transportation, which deals with the repositioning of bikes in public bike sharing systems, the Bike Sharing Repositioning Problem (BSRP for short). Existing work on the BSRP is extended by allowing fixed routes, time windows, multiple vehicles and route repetition. An integer programming formulation of the resulting variation of the BSRP is presented. Solution strategies based on current practice for similar pickup and delivery problems (which are often of high complexity) are discussed. More specifically, a column generation approach based on a branch-and-price scheme which is solved by dynamic programming is described.

Preliminary computational experience with the reported branch-and-price algorithm is promising. As future work to be reported later, the authors intend to validate the proposed model and methods and are currently exploring ways to reduce the computational effort in generating columns throughout the branch-and-price scheme. Particularly, the approach described at the end of Section 5 is now being tested with a set of synthetically generated instances. The results obtained so far partially confirm the expectation of finding useful solutions while reducing the necessary computational effort.

Acknowledgments

The authors are grateful for the financial support of FAPEG (Fundação de Amparo à Pesquisa do Estado de Goiás) through the call 07/2014 – Fapeg Universal.

References

- Alvarez-Valdes, R., Belenguer, J. M., Benavent, E., Bermudez, J. D., Muñoz, F., Vercher, E., e Verdejo, F. (2016). Optimizing the level of service quality of a bike-sharing system. *Omega*, 62: 163–175.
- Caggiani, L. e Ottomanelli, M. (2013). A Dynamic Simulation based Model for Optimal Fleet Repositioning in Bike-sharing Systems. *Procedia - Social and Behavioral Sciences*, 87:203–210. SIDT Scientific Seminar 2012.
- Çelebi, D., Yörüsün, A., e Işık, H. (2018). Bicycle sharing system design with capacity allocations. *Transportation Research Part B: Methodological*, 114:86–98.
- Chemla, D., Meunier, F., e Calvo, R. W. (2013). Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*, 10(2):120–146.
- DeMaio, P. (2009). Bike-sharing: History, impacts, models of provision, and future. *Journal of Public Transportation*, 12(4):41–56.
- Erdoğan, G., Battarra, M., e Calvo, R. W. (2015). An exact algorithm for the static rebalancing problem arising in bicycle sharing systems. *European Journal of Operational Research*, 245(3): 667–679.
- Ho, S. C. e Szeto, W. Y. (2017). A hybrid large neighborhood search for the static multi-vehicle bike-repositioning problem. *Transportation Research Part B: Methodological*, 95:340–363.

- Laporte, G., Meunier, F., e Wolfler Calvo, R. (2015). Shared mobility systems. *4OR*, 13(4):341–360.
- Liu, Y., Szeto, W., e Ho, S. C. (2018). A static free-floating bike repositioning problem with multiple heterogeneous vehicles, multiple depots, and multiple visits. *Transportation Research Part C: Emerging Technologies*, 92:208–242.
- Ma, L. (2017). Site Visits, Policy Learning, and the Diffusion of Policy Innovation: Evidence from Public Bicycle Programs in China. *Journal of Chinese Political Science*, 22(4):581–599.
- Rainer-Harbach, M., Papazek, P., Raidl, G. R., Hu, B., e Kloimüller, C. (2015). PILOT, GRASP, and VNS approaches for the static balancing of bicycle sharing systems. *Journal of Global Optimization*, 63(3):597–629.
- Schuijbroek, J., Hampshire, R. C., e van Hoes, W.-J. (2017). Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research*, 257(3):992–1004.
- Shui, C. e Szeto, W. (2018). Dynamic green bike repositioning problem—A hybrid rolling horizon artificial bee colony algorithm approach. *Transportation Research Part D: Transport and Environment*, 60:119–136. Special Issue on Traffic Modeling for Low-Emission Transport.
- Vogel, P. e Mattfel, D. C. (2010). Modeling of repositioning activities in bike-sharing systems. In *Proceedings of the 12th World conference on transport research (WCTR) – Jul, 7th 2010*, p. 1–13, Lisbon, Portugal.