Technical Report - An exact approach for the static rebalancing problem that minimizes the routing costs and unmet demand under various geographic considerations

Abstract

Bike sharing systems have the potential to significantly alleviate traffic congestion, reduce emissions, and lower the need for parking space in city centers. One critical factor influencing the success of a bike sharing system is the effectiveness of rebalancing operations. These operations involve repositioning of the bikes at available stations through pickup and delivery activities performed by trucks, so that the anticipated user demand is satisfied. The Static Docked Bike Rebalancing Problem (SBRP) focuses on determining a cost-effective sequence of stations to be visited by trucks, along with the corresponding quantity of bikes to be picked up or delivered at each station.

In this work we study the static Bike Rebalancing Problem (BRP) in a docked network, where the redistribution of the bikes takes place via a fleet of capacitated vehicles. Moreover, the impact of the geographic distribution of the stations is also considered. The objective is to minimize the total routing costs and unsatisfied user demand. The BRP belongs to the class of Pickup and Delivery Vehicle Routing Problems (PDVRPs). In a PDVRP a fleet of vehicles is utilized in order to transport a set of requests from a single depot to multiple pickup and delivery customers of a distribution network. Specifically, the pickup customers require a single vehicle to collect a certain amount of freight, while delivery customers require a single vehicle to deliver a specific amount of freight to them. The BRP is a strongly NP-hard problem as it expands the well-known Capacitated Vehicle Routing Problem (CVRP), where all nodes in the network are either all pickup nodes or delivery nodes.

1. Mathematical Formulation

1.1. Initial Formulation

The Static Bike sharing Rebalancing Problem (SBRP) is formulated in this section. The goal is to determine (a) the route of each vehicle performing bike repositioning operations, and (b) the number of bikes to load or unload at each station visited by one vehicle. These are determined by minimizing the total routing costs and the total penalty of unsatisfied demand.

Let $G = (V, A)$ be a complete graph, where the set of vertices $V = \{0, 1, ..., n\}$ consists of the depot $V = 0$ and the bike stations $V = \{1, 2, ..., n\}$, and A is the set of *arcs*. Each station i has an actual demand, which is calculated by the subtraction of the forecast and the status of a station. The forecasted demand is evaluated based on the type of the station (close to public transport, residential area, area of interest for commuters) and the status of a station refers to the number of available bikes at a station before redistribution. A negative actual demand indicates excess of bikes, marking a station as a pickup station where one or more bikes can be picked up. A positive actual demand indicates a shortage of bikes, and thus the station is marked as a delivery station where bikes need to be dropped-off. A fleet of k identical vehicles of capacity Q is available at the depot to perform repositioning operations. Each arc $(i, j) \in A$ is associated with a travel distance denoted as c_{ij}. All vehicles start the rebalancing operations empty from a single depot which is also the ending point of each vehicle route. The above decisions are subject to the vehicle capacity Q and the bike station capacity P_i constraints. The repositioning operations are considered to take place every night in the SBRP described below, since we associate the user demand with public transport network operations.

Table 1 presents the notation, parameters and decision variables used to define the mathematical model.

Table 1: Notation, parameters and decision variables used in the mathematical model.

Sets:	
N	set of bike stations, including the depot
S	set of bike stations
K	set of vehicles
Parameters:	
Q	vehicle capacity
P_i	parking capacity in station $i \in N$
η_i^0	initial status in bike station $i \in N$
l_{0k}	initial load of vehicle $k \in K$
c_{ij}	routing cost between i,j for $i, j \in N$
p	penalty for unmet demand
f_i	forecasted demand in node i
Variables:	
x_{ij}^k	binary variable indicating whether a vehicle $k \in K$ traverses arc i,j $\in N$, with $x_{ij}^k = 1$ if the arc is used and 0 otherwise
y_{ik}^U	number of bikes loaded in vehicle $k \in K$ at station $i \in N$
y_{ik}^L	number of bikes unloaded in vehicle $k \in K$ at station $i \in N$
l_{ik}	load of vehicle $k \in K$ after serving station $i \in N$
η_i	status of station $i \in N$ after being served
g_i^k	binary variable, where $g_i^k = 1$ if station $i \in N$ is a load station and 0 otherwise
d_i^k	binary variable, where $d_i^k = 1$ if station $i \in N$ is an unload station and 0 otherwise

The initial mathematical formulation of the Bike Sharing Rebalancing Problem is presented and explained in the following passage.

$$\min \quad \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} x_{ij}^k c_{ij} + \sum_{i \in N} p \cdot \max\{f_i - \eta_i, 0\} \tag{1}$$

$$\text{s.t.} \quad \sum_{j \in S} x_{0j}^k = 1 \qquad \forall k \in K \tag{2}$$

$$\sum_{j \in S} x_{j0}^k = 1 \qquad \forall k \in K \tag{3}$$

$$\sum_{j \in N} \sum_{k \in K} x_{ij}^k \leq 1 \qquad \forall i \in N \backslash \{0\} \tag{4}$$

$$\sum_{i \in N} x_{ij}^k = \sum_{i \in N} x_{ji}^k \qquad \forall j \in N \backslash \{0\}, \quad k \in K \tag{5}$$

$$\sum_{i \in S} \sum_{j \in S} \sum_{k \in K} x_{ij}^k \leq |S| - 1, \qquad S \subset N, 2 \leq |S| \leq n-1 \tag{6}$$

$$l_{jk} x_{ij}^k = (l_{ik} + y_{jk}^L - y_{jk}^U) x_{ij}^k \qquad \forall i, j \in N, \quad k \in K \tag{7}$$

$$0 \leq l_{ik} \leq Q \qquad \forall i \in N, \quad k \in K \tag{8}$$

$$0 \leq y_{jk}^L \leq \sum_{i \in N} x_{ij}^k (Q - l_{ik}) \qquad \forall j \in N, \quad k \in K \tag{9}$$

$$0 \leq y_{jk}^U \leq \sum_{i \in N} x_{ij}^k l_{ik} \qquad \forall j \in N, \quad k \in K \tag{10}$$

$$y_{ik}^L \leq Q g_i^k \qquad \forall i \in N, \quad k \in K \tag{11}$$

$$y_{ik}^U \leq Q d_i^k \qquad \forall i \in N, \quad k \in K \tag{12}$$

$$y_{ik}^L \geq g_i^k \qquad \forall i \in N, \quad k \in K \tag{13}$$

$$y_{ik}^U \geq d_i^k \qquad \forall i \in N, \quad k \in K \tag{14}$$

$$g_i^k + d_i^k \leq 1 \qquad \forall i \in N, \quad k \in K \tag{15}$$

$$\eta_i^0 - \sum_{k \in K} y_{ik}^L + \sum_{k \in K} y_{ik}^U \leq P_i \qquad \forall i \in N \tag{16}$$

$$\eta_i = \eta_i^0 - \sum_{k \in K} y_{ik}^L + \sum_{k \in K} y_{ik}^U \qquad \forall i \in N \tag{17}$$

$$y_{ik}^L \geq 0, y_{ik}^U \geq 0 \qquad \forall i \in N, \quad k \in K \tag{18}$$

$$l_{ik} \geq 0 \qquad \forall i \in N, \quad k \in K \tag{19}$$

$$\eta_i \geq 0 \qquad \forall i \in N \tag{20}$$

The objective function (1) minimizes the routing cost of the bike redistribution in the network, as well as the penalty of unmet demand in a station. The term $\max\{f_i - \eta_i, 0\}$ in the objective function ensures that the penalty of unmet demand is only incurred in case we have a deficit of bikes in a station (that is, $f_i - \eta_i > 0$). Constraints (2),(3) ensure that all vehicles leave from and return at the depot. The following constraint ensures that each bike station is served at most once and by one vehicle (4). Constraint (5) ensures flow conservation in every bike station. Constraint (6) eliminates all solutions containing subtours. That is, for every subset of nodes (S: 2$\leq |S| \leq |N|-1$) it forbids the number of selected arcs within S to be equal to or larger than the number of nodes in S. Following, the number of bikes loaded on the vehicle after serving station j is calculated (7). The number of bikes loaded on the vehicle should not exceed vehicle capacity (8). Constraints (9) and (10) calculate the number of available bikes to load/unload in every station i. Constrains (11)-(14) determine whether a station is a load ($g_i^k = 1$) or unload ($d_i^k = 1$) station and connects this amount with the total available bikes. Next, constraint (15) ensures that a station can be either a load or an unload station. The limitations of parking station capacity are taken into account using constraint (16), where η_i^0 is the initial load (status) of a station i. The status of a station after being served by vehicle k is denoted by η_i in constraint (17). Finally, (18)-(20) are non-negativity constraints.

1.2. Linearization of the mathematical model

In this section the required changes to linearize the initial mathematical formulation are performed. In particular, the objective function, the constraint regarding the total load of the vehicle (7), and the number of available bikes to load or unload in each station (9)-(10), are not linear and require modifications if one wishes to solve the problem to global optimality. In order to linearize these functions, auxiliary variables are introduced, as well as a very large positive number M (big-M). These are summarized in the Table 2.

Table 2: Variables introduced for linearization.

r_i	integer auxiliary variable to penalize shortage
θ_i	binary auxiliary variable, where $\theta_i = 1$ if there is a shortage of bikes in station $i \in N$

Proposition 1. *The objective function (1) can be linearized by introducing an auxiliary variable r_i that substitutes the $\max\{f_i - \eta_i, 0\}$ part, a binary auxiliary variable θ_i, and the constraints (21)-(24):*

$$r_i \geq f_i - \eta_i \qquad \forall i \in N \tag{21}$$

$$r_i \geq 0 \qquad \forall i \in N \tag{22}$$

$$r_i \leq M\theta_i \qquad \forall i \in N \tag{23}$$

$$r_i \leq f_i - \eta_i + M(1 - \theta_i) \qquad \forall i \in N \tag{24}$$

Proof. Let the new objective function be:

$$\min \quad \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} x_{ij}^k c_{ij} + \sum_{i \in N} r_i p \tag{25}$$

where r_i is an auxiliary variable. Notice that $r_i = f_i - \eta_i$ for any $i \in N$, because constraints (21)-(24) force r_i to be equal to $f_i - \eta_i$ if $f_i - \eta_i \geq 0$ and $r_i = 0$ if $(f_i - \eta_i) \leq 0$. Thus, the linear objective function (25) together with the linear constraints (21)-(24) have the same effect as the nonlinear objective function (1). □

Proposition 2. *The nonlinear constraint (7) and the following linear constraints are equisatisfiable.*

$$l_{jk} \leq l_{ik} + y^L_{jk} - y^U_{jk} + M(1 - x^k_{ij}) \qquad \forall i, j \in N, \quad k \in K \tag{26}$$

$$l_{jk} \geq l_{ik} + y^L_{jk} - y^U_{jk} - M(1 - x^k_{ij}) \qquad \forall i, j \in N, \quad k \in K \tag{27}$$

Proof. Constraint (7) forces l_{jk} to be equal to $l_{ik} + y^L_{jk} - y^U_{jk}$ for $x^k_{ij} = 1$ and allows l_{jk} to take any value for $x^k_{ij} = 0$. Similarly, if $x^k_{ij} = 1$ constraints (26)-(27) force l_{jk} to be equal to $l_{ik} + y^L_{jk} - y^U_{jk}$ and if $x^k_{ij} = 0$ constraints (26)-(27) allow l_{jk} to take any value. □

Ultimately, nonlinear constraints (9)-(10) are substituted by (28)-(31) using again big-M formulations to avoid the multiplication of the decision variables.

$$0 \leq y^L_{jk} \leq Q - l_{ik} + M(1 - x^k_{ij}) \qquad \forall i, j \in N, \quad k \in K \tag{28}$$

$$y^L_{jk} \leq M \sum_{i \in N} x^k_{ij} \qquad \forall j \in N, \quad k \in K \tag{29}$$

$$0 \leq y^U_{jk} \leq l_{ik} + M(1 - x^k_{ij}) \qquad \forall i, j \in N, \quad k \in K \tag{30}$$

$$y^U_{jk} \leq M \sum_{i \in N} x^k_{ij} \qquad \forall j \in N, \quad k \in K \tag{31}$$

The final mathematical model formulation of the linearized model that is defined as the “Base Model" consists of the linearized objective function (25), constraints (2)-(6), (8), (11)-(24), (26)-(31). The SBRP generalizes the well-known Capacitated Vehicle Routing Problem (CVRP), where customers are either all pickup vertices or all delivery vertices and thus, it is an NP-hard combinatorial optimization problem.

1.3. Solution space reduction steps

As a step to reduce the solution space and make the model more compact, the dimension of the vehicle is removed from the binary decision variables g^k_i and d^k_i of the initial formulation. The reduced variables are rewritten as g_i and d_i and the solution space is reduced from 2^{nm} to 2^n. The equations (11)-(15) as described in the initial formulation are therefore adjusted to the following (32)-(36):

$$y_{ik}^L \leq Q g_i \qquad \forall i \in N, \quad k \in K \tag{32}$$
$$y_{ik}^U \leq Q d_i \qquad \forall i \in N, \quad k \in K \tag{33}$$
$$y_{ik}^L \geq g_i \qquad \forall i \in N, \quad k \in K \tag{34}$$
$$y_{ik}^U \geq d_i \qquad \forall i \in N, \quad k \in K \tag{35}$$
$$g_i + d_i \leq 1 \qquad \forall i \in N \tag{36}$$

1.4. Lazy constraints

In the mathematical formulation discussed in the previous section, the Explicit Dantzig-Fulkerson-Johnson formulation (**?**) is considered, as shown in equation (6). This formulation uses subsets to eliminate subtours. The idea behind the formulation is that, for every subset that could form a subtour there are at least two arcs that should connect the nodes from the subsets to nodes outside of the subset. The number of constraints, however, increases exponentially with the number of nodes making it impossible to generate all constraints for large problem instances. For this purpose, we use constraints (6) as lazy constraints. Unlike normal constraints, lazy constraints are not generated in advance but are only generated when needed instead. More specifically, constraints (6) are added as lazy constraints inside a callback which is called whenever a new candidate incumbent solution is found. The callback procedure checks whether the candidate solution forms subtours. If not, a feasible solution for the rebalancing problem is found and the solution is accepted; otherwise constraints (6) are added for each subtour.

1.5. Valid inequalities

Preliminary test results indicated, that due to the complexity (NP-hardness) of the problem, for problem instances more than 10 station locations, we cannot obtain an optimal value or a tighter lower bound (refer to Table **??**). To ensure that the optimal value or a tighter lower bound can be derived within the time limit and without out-of-memory errors, valid inequalities are proposed. Valid inequalities try to tighten the base model and improve the convergence to the globally optimal solution.

The following set of inequalities is proposed and apply to the case where the demand of each station can be fully satisfied. Let q_i represent the demand of bikes required by each bike station, i.e $q_i = f_i - \eta_i^0$. Then the following inequalities are valid:

$$x_{ij}^k + x_{jh}^k \leq 1, \quad \forall i, j, h \in N : |q_i + q_j + q_h| > Q \tag{37}$$

We note that in valid inequalities (37), only one of the x^k_{ij}, x^k_{jh} variables may take the value of 1 because of the capacity constraints. In addition, one may use several classic valid inequalities for the Vehicle Routing Problem (see **?** for a comprehensive review).

We also introduce the following valid inequalities related to the load of the vehicle. The following constraints refer to the cases where there is either a fully loaded or an entirely empty truck. In the first case, of the fully loaded truck, it is evident that the next station to be served cannot be a pickup node, this is ensured by introducing constraint (38). Similarly, in the case of an empty truck, the next station cannot be a delivery node (39).

$$q - l_{ik} + \theta_j \geq x^k_{ij} \qquad \forall i, j \in N,\ \forall k \in K \tag{38}$$

$$l_{ik} + (1 - \theta_j)M \geq x^k_{ij} \qquad \forall i, j \in N,\ \forall k \in K \tag{39}$$

2. Greedy Randomized Savings Heuristic

In an effort to improve the computational times required to obtain an optimal solution, a heuristic algorithm has been developed. The developed algorithm is capable of deriving high-quality solutions in short computational times for medium and large-scale problem instances. The output of the heuristic was then used to generate a warm-starting technique of providing an initial feasible solution to the solver before starting the optimization process.

Starting from an empty reference set of solutions P, a number of high quality solutions are generated. A greedy randomized savings heuristic algorithm (Lines 6-12) is employed for this purpose by performing minimal cost insertions of pickup (or delivery) service locations. A pseudocode of the proposed solution framework is presented next (Algorithm 1). In the beginning, empty routes are initiated each one containing just two occurrences of the depot node, corresponding to the start and end of the route (Line 4). The number of routes is equal to the number of vehicles available to perform the rebalancing operations. Set U is initialised as a copy of the set of service locations S, representing the set of yet unvisited nodes. Next, the algorithm selects at random a single pickup node to be inserted in each empty route (Line 5).

At each iteration the algorithm identifies the minimal cost insertion position for a single pickup or delivery station. The total set of routes is constructed by iteratively inserting service locations, until all stations are served. In particular, the insertion should lead to feasible vehicle routes w.r.t. the vehicle capacity constraints (Line 8) and the station capacity constraints (Line 9). The best insertion identified in

Algorithm 1 Pseudocode of the employed heuristic algorithm that generates a set P of high-quality solutions and identifies the best one Z^{best}

Set P $\leftarrow \emptyset$, **Solution** $Z^{best} \leftarrow \emptyset$, **Int** $maxIters$
while $counter < maxIters$ **do**
    Solution Z $\leftarrow \emptyset$, **Set** U $\leftarrow S$
    Initialise empty routes
    Stochastically identify a single pickup service location to be added in each empty route in solution Z
    while $U \neq \emptyset$ **do**
        Identify the min cost insertion *ins* to add to the semi-complete solution Z such that:
        (a) vehicle capacity constraints are not violated
        (b) station capacity constraints are not violated
        Apply insertion *ins*
        Reevaluate route load after insertion point
        Remove service location from U
    end while
    $P \leftarrow P \cup \{Z\}$
    if $cost(Z) < cost(Z^{best})$ **then**
        $Z^{best} \leftarrow Z$
    end if
end while
return Z^{best}

each iteration is characterised by i) the selected service location, ii) the selected and iii) the point within the selected route where the selected service location will be inserted. The selected insertion is applied and the vehicle load after the insertion point is reevaluated (Line 11). The generated solution Z is used to update the reference set P (Line 14). The algorithm is terminated after a certain number of iterations ($maxIters$) is reached, by returning the lowest objective solution Z^{best} encountered. Note that the algorithm is capable of exploring solutions where the demand of delivery service locations is not fully satisfied. Thus, the algorithm is still able to provide solutions for realistic scenarios where the stations' cannot be fully satisfied due to unavailability of bikes.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation program SUM under grant agreement No. 101103646.