Computers & Operations Research 71 (2016) 149–162

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

A destroy and repair algorithm for the Bike sharing Rebalancing Problem

Mauro Dell'Amico [a], Manuel Iori [a], Stefano Novellani [a,b,*], Thomas Stützle [c]

[a] DISMI, University of Modena and Reggio Emilia, Via Amendola 2, 42122 Reggio Emilia, Italy
[b] "Guglielmo Marconi" - DEI, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy
[c] IRIDIA, Université Libre de Bruxelles (ULB), 50, Av. F. Roosevelt, CP 194/6 B-1050 Brussels, Belgium

ARTICLE INFO

Available online 22 January 2016

Keywords:
Bike sharing
Rebalancing
Destroy and repair
Speed-up techniques
Branch-and-cut

ABSTRACT

In this paper, we deal with the Bike sharing Rebalancing Problem (BRP), which is the problem of driving a fleet of capacitated vehicles to redistribute bicycles among the stations of a bike sharing system. We tackle the BRP with a destroy and repair metaheuristic algorithm, which makes use of a new effective constructive heuristic and of several local search procedures. The computational effort required for the neighborhood explorations is reduced by means of a set of techniques based on the properties of feasible BRP solutions. In addition, the algorithm is adapted to solve the one-commodity Pickup and Delivery Vehicle Routing Problem with maximum Duration (1-PDVRPD), which is the variant of the BRP in which a maximum duration is imposed on each route.

Extensive computational results on instances from the literature and on newly-collected large-size real-world instances are provided. Our destroy and repair algorithm compares very well with respect to an exact branch-and-cut algorithm and a previous metaheuristic algorithm in the literature. It improves several best-known solutions, providing high quality results on both problem variants.

1. Introduction

Bike sharing systems are public or private systems designed to increase the use of bicycles and decrease congestion, to solve the last mile problem, and to provide a mobility service to the users where other means of transportation are not available. These systems were born in the 1960s in Amsterdam (see, e.g., DeMaio [1]) and spread all over the world now, counting more than 700 operating systems and more than 200 systems planned, under construction, or about to be implemented (see e.g., DeMaio and Meddin [2]).

Bike sharing systems are composed of several bike stations located in different sites around the city, and each station is composed of a number of bike slots where bicycles can be collected or returned. In a balanced situation, each bike station must have a certain number of empty slots, to allow arrivals, and a certain number of full slots, to allow departures.

Let us define the *level of occupation* of a station as the number of bicycles present in that station, and the *balanced level of occupation* as the level of occupation that the system operator desires in a station. After a certain amount of time from the beginning of the service, the users will have moved the bicycles among the stations of the system and the level of occupation will have gone far from the balanced one. For example, the stations on the top of a hill will be typically empty and the stations at the bottom will be typically full. This situation creates inefficiency in the system such that users cannot collect or return bicycles when they need to. The system operators want the stations to be brought back to their balanced levels of occupation to avoid the inefficiency created by full and/or empty stations, so they perform a redistribution of bicycles that is called *rebalancing*. The rebalancing is performed by capacitated vehicles and it is normally required at the end of the day, when the system is closed or when the use of the system can be considered negligible. In this case, the rebalancing is called static. Some bike sharing operators may require that the rebalancing is performed when the system is open, incurring in a situation that is called dynamic rebalancing. In this paper we study the static case.

The *Bike sharing Rebalancing Problem* (BRP) is a problem related to the static rebalancing of a bike sharing system, that requires to drive a fleet of homogeneous and capacitated vehicles to rebalance the stations of a bike sharing system at minimum cost. In this paper, we present a new constructive heuristic and a set of efficient local searches that are included into a destroy and repair

* Corresponding author at: DISMI, University of Modena and Reggio Emilia, Via Amendola 2, 42122 Reggio Emilia, Italy.
E-mail addresses: mauro.dellamico@unimore.it (M. Dell'Amico), manuel.iori@unimore.it (M. Iori), stefano.novellani@unimore.it (S. Novellani), stuetzle@ulb.ac.be (T. Stützle).

http://dx.doi.org/10.1016/j.cor.2016.01.011
0305-0548/© 2016 Elsevier Ltd. All rights reserved.

metaheuristic algorithm for the BRP. This new algorithm provides shorter solving times for the easiest instances and better upper bounds for the hardest instances when compared to the branch-and-cut algorithm developed by Dell'Amico et al. [3] and good quality solutions for newly collected, large real-world instances.

Moreover, we adapted the metaheuristic algorithm to the *one-commodity Pickup and Delivery Vehicle Routing Problem with maximum Duration* (1-PDVRPD), a variation of the BRP where a maximum duration constraint is imposed to each route. Instances of the 1-PDVRPD from the literature have been solved and the best known solutions have been strongly improved. To solve the 1-PDVRPD we have also adapted the branch-and-cut algorithm of Dell'Amico et al. [3], by efficiently handling the maximum duration limit as a set of constraints of exponential size. This method applies to two-index formulations for general vehicle routing problems, so we believe it is a further interesting contribution in this field of research.

The paper is structured as follows. In Section 2, we formally define the BRP and the 1-PDVRPD. Mathematical models are proposed for both problems and a brief literature review is reported. In Section 3, we describe some properties of the BRP that allow us to speed up local searches. In Section 4, we present the framework of the proposed destroy and repair algorithm for the BRP and we describe its components. In Section 5, we describe the adaptation of the destroy and repair and of the branch-and-cut algorithms to the 1-PDVRPD. In Section 6, extensive computational results on newly collected real instances and on literature instances are presented. Section 7 concludes the work.

2. Problem description

The *Bike sharing Rebalancing Problem* (BRP) is modeled on a complete digraph $G=(V,A)$, where $V=\{0,1,\dots,n\}$ is the set of vertices including the depot (vertex 0) and the n stations (vertices $1,\dots,n$), and A is the set of arcs between each pair of vertices. Let c_{ij} be the non-negative cost associated with the arc $(i,j)\in A$. We assume that the triangular inequality valid for the cost matrix. For each vertex $i\in V$ a request q_i is given, with $q_0=0$. Requests can be either positive or negative. If a station has a positive request, we call it a pickup station; if, instead, it has a negative request, we call it a delivery station. The quantities picked up at pickup stations can be used to respond to requests of delivery stations or can be returned directly to the depot. Vehicles can leave the depot not necessarily empty, if needed. The objective is to drive a fleet of m identical vehicles of capacity Q available at the depot to respond to requests and to minimize the total cost of the traversed arcs. As commonly assumed in the related literature (see, e.g., Dell'Amico et al. [3]), we impose that a station with request $q_i=0$ must be visited, even if this implies that no bike has to be dropped off or picked up there. This is imposed to allow the routine inspection of bikes and stations. The case in which stations with null request have to be skipped can be simply obtained by removing them in a preprocessing phase.

2.1. Formulation for the BRP

In this section, we present a *mixed integer linear programming* (MILP) formulation for the BRP. The starting point is *Formulation F3* by Dell'Amico et al. [3] built on the *multiple Traveling Salesman Problem* (m-TSP), in which at most m uncapacitated vehicles based at a central depot have to visit a set of vertices, with the constraint that each vertex is visited exactly once. By defining a binary variable x_{ij}, taking value 1 if arc (i,j) is traveled by a vehicle, and 0 otherwise, the BRP can be modeled as (1)–(6).

$$\min z=\sum_{i\in V}\sum_{j\in V}c_{ij}x_{ij} \tag{1}$$

$$\sum_{i\in V}x_{ij}=1 \quad j\in V\setminus\{0\} \tag{2}$$

$$\sum_{i\in V}x_{ji}=1 \quad j\in V\setminus\{0\} \tag{3}$$

$$\sum_{j\in V}x_{0j}\le m \tag{4}$$

$$\sum_{i\in S}\sum_{j\in S}x_{ij}\le|S|-\max\left\{1,\left\lceil\frac{\left|\sum_{i\in S}q_i\right|}{Q}\right\rceil\right\} \quad S\subseteq V\setminus\{0\},S\ne\varnothing \tag{5}$$

$$x_{ij}\in\{0,1\} \quad i,j\in V. \tag{6}$$

Objective function (1) minimizes the traveling costs. Constraints (2) and (3) impose that every vertex but the depot is visited once. Constraints (4) ensure that at most m vehicles leave the depot. To guarantee the feasibility of a solution with respect to the BRP, we need to substitute the typical subtour elimination constraints by the family of constraints (5) in the m-TSP formulation, so as to ensure that requests are satisfied and vehicles capacities are not exceeded. Constraints (5) are similar to the *generalized subtour elimination constraints* for the CVRP. They state that, for each subset S of vertices, the number of arcs with both tail and head in S should not exceed the cardinality of S minus the minimum number of vehicles required to serve S. Following Hernández-Pérez and Salazar-González [4], an estimation of the minimum number of vehicles is simply obtained by computing the absolute value of the sum of the requests, dividing it by the vehicle capacity and then rounding up the result. This value can be zero when the sum of the q_i is null; in such a case, the value one is used instead, because at least one vehicle is needed (notice that S does not contain the depot) to impose the connectivity of the solution. Constraints (5) are exponentially many, so the above formulation can be solved in *branch-and-cut* (B&C) fashion, by invoking a separation procedure to divide the violated ones from the non-violated ones, and then adding the violated cuts to the model in an iterative way. For details we refer to Dell'Amico et al. [3] and to Section 5.2 below.

2.2. A generalization of the BRP: the 1-PDVRPD

The *one-commodity Pickup and Delivery Vehicle Routing Problem with maximum Duration* (1-PDVRPD) is a generalization of the BRP in which a maximum duration constraint is imposed for the routes. The problem was solved by Shi et al. [5], in which is imposed a maximum distance D to each route, where the distance is computed as the sum of the traveling costs. In this paper, we solve a more general version of the 1-PDVRPD where the maximum distance D is considered as a maximum duration in time, T, where a time t_{ij} of traveling the arc $(i,j)\in A$ is introduced and it is proportional to the cost between i and j, i.e., $c_{ij}=\tau t_{ij}$, with τ a non-negative parameter. Moreover, we consider a service time s_i in each station that depends on the quantity to be picked up or dropped off, imposing that $s_i=\sigma|q_i|$, with σ a non-negative parameter. We solve the version of the 1-PDVRPD that takes into account the traveling time t_{ij}, the service time s_i, and the maximum duration of a route T because it fits better with the BRP. The 1-PDVRPD proposed by Shi et. al [5] is a particular case of our version, where $\tau=1,\sigma=0$, and $T=D$.

For modeling the 1-PDVRPD as a MILP we need to express the maximum duration constraint. To this aim, we define a path P as an ordered sequence of vertices $P(0),P(1),\dots,P(|P|)$. We use $\mathcal{R}$ to identify the set of routes, i.e., the set of paths P starting and ending at the depot,