Computers & Operations Research 63 (2015) 83–101

Contents lists available at ScienceDirect

Computers & Operations Research

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

Exact algorithms for the double vehicle routing problem with multiple stacks

Manuel Iori [a], Jorge Riera-Ledesma [b,*]

[a] DISMI, University of Modena and Reggio Emilia, Via Amendola 2, Pad. Morselli, 42122 Reggio Emilia, Italy

[b] Universidad de La Laguna, Escuela Superior de Ingeniería y Tecnología, Sección de Ingeniería Informática, Departamento de Ingeniería Informática y de Sistemas, Apartado de correos 456, 38200 La Laguna, Spain

ARTICLE INFO

Available online 7 May 2015

Keywords:
Vehicle routing problem
One-to-one pickup-and-delivery
Last-in-first-out
Branch-and-cut
Branch-and-price-and-cut

ABSTRACT

The Double Traveling Salesman Problem with Multiple Stacks (DTSPMS) is a one-to-one pickup-and-delivery single-vehicle routing problem with backhaul deliveries. The vehicle carries a container divided into stacks of fixed height, each following a Last-In-First-Out policy, and the aim is to perform pickups and deliveries by minimizing the total routing cost and ensuring a feasible loading/unloading of the vehicle.

A realistic generalization of the DTSPMS arises when a single vehicle is not enough to collect all products, and therefore multiple, and possibly heterogeneous vehicles are needed to perform the transportation operations. This paper introduces and formulates this generalization, that we refer as the Double Vehicle Routing Problem with Multiple Stacks. It proposes three models, the first one based on a three-index formulation and solved by a branch-and-cut algorithm, and the other two based on two set partitioning formulations using different families of columns and solved by a branch-and-price and a branch-and-price-and-cut algorithm, respectively.

The performance of these algorithms has been studied on a wide family of benchmark test instances, observing that, although the branch-and-cut algorithm shows a better performance on instances with a small number of vehicles, the performance of the branch-and-price and the branch-and-price-and-cut algorithms improves as the number of vehicles grows. Additionally, the first set partitioning formulation yields tighter lower bounds, but the second formulation, because of its simplicity, provides better convergence properties, solving instances with up to fifty vertices to proven optimality.

1. Introduction

The *Double Traveling Salesman Problem with Multiple Stacks* (DTSPMS) is a one-to-one pickup-and-delivery single-vehicle routing problem (see, e.g., Berbeglia et al. [1] and Cordeau et al. [2]), in which a single vehicle performs all pickup operations in a first region, by loading products one at a time as they are collected, and then is transferred to a second region where it performs all the delivery operations. The problem arises in contexts where the pickup and delivery regions are widely separated, and the way the collected products are loaded into the vehicle may affect the possible delivery sequence.

Formally, the pickup and the delivery vertices, in addition to a depot in each region, are known in advance. A routing cost is associated with each pair of vertices in each region. The transportation cost between the two regions is fixed and, therefore, not considered as part of the optimization problem. Each product is associated exactly and unambiguously with a pickup vertex in the pickup region and with a delivery vertex in the delivery region. All products have identical shape and size, and the vehicle has a loading space, that is, a container, divided into stacks of fixed height. The loading operations follow a *Last-In-First-Out* (LIFO) policy on each stack. Due to the fact that each vehicle carries a single container, in the following we will use the terms vehicle and container interchangeably.

The DTSPMS requires collecting all products in the pickup region following a Hamiltonian cycle that starts at the pickup depot, and then delivering these products in the delivery region with a second Hamiltonian cycle starting at the delivery depot. The aim is to minimize the total routing cost while ensuring that there exists a feasible *loading plan* of the products into the vehicle, that is, the collected products may be loaded into the vehicle without exceeding the stacks height and may be unloaded following the LIFO policy on each stack. Fig. 1 depicts a DTSPMS solution on an instance involving 16 products (hence, containing 32 customers and 2 depots). Each product is associated with a pickup customer

* Corresponding author. Tel.: +34 922845040; fax: +34 922318170.
E-mail addresses: manuel.iori@unimore.it (M. Iori), jriera@ull.edu.es (J. Riera-Ledesma).

http://dx.doi.org/10.1016/j.cor.2015.04.016
0305-0548/

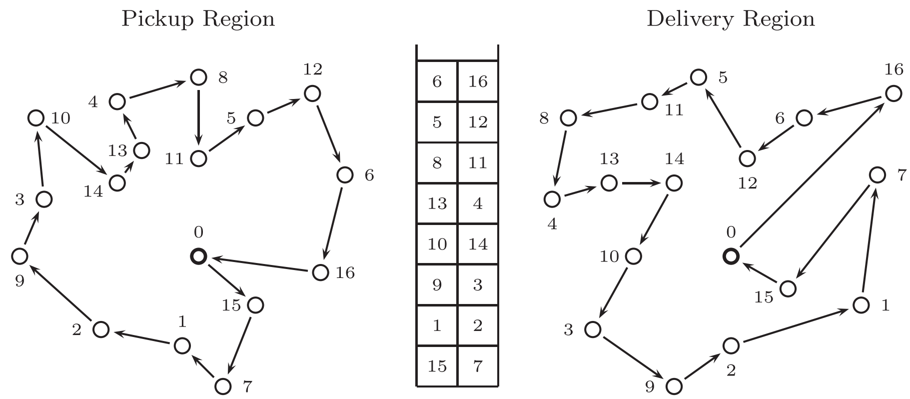

Fig. 1. Solution of a DTSPMS instance involving 16 products and a 2×8 vehicle.

and a delivery customer (product 1 is associated with pickup customer 1 and delivery customer 1, product 2 with pickup customer 2 and delivery customer 2, and so on), and the container is divided into 2 stacks of height 8.

The DTSPMS combines two instances of the well-known *Traveling Salesman Problem* (TSP), one for the pickup region and one for the delivery region, with the combinatorial problem of feasibly assigning the products to the stacks of the container. The DTSPMS was introduced by Pertersen and Madsen [3] in the context of a real-world application on multimodal transport, where a 40-foot container (configured as 3 stacks of height 11) is used to transport up to 33 pallets from a set of pickup customers to a set of delivery customers. After proposing a mathematical formulation, Pertersen and Madsen [3] presented a simulated annealing heuristic algorithm and tested it on a set of instances that have now become a widely used benchmark. Since this seminal paper there has been a growing interest in exact algorithms for the DTSPMS: Lusby et al. [4] proposed an exact solution method based on matching the *k*-best TSP tours on each of the separate pickup and delivery regions; Carrabs et al. [5] developed a branch-and-bound algorithm to solve the particular case in which the vehicle has exactly two stacks; Petersen et al. [6] studied different formulations and proposed a branch-and-cut algorithm based on an *infeasible path model*; this model was later improved by Alba Martínez et al. [7] by means of new valid inequalities and algorithmic refinements. Heuristic algorithms have been recently proposed by Felipe et al. [8–10], who implemented a variable neighborhood search, and by Côté et al. [11], who proposed a large neighborhood search.

In this paper we consider the realistic generalization of the DTSPMS where multiple containers are used, thus giving rise to multiple routes to perform the collections and the deliveries. Because of its similarity with the *Vehicle Routing Problem* (VRP), see, e.g., Toth and Vigo [12], we call this generalization the *Double Vehicle Routing Problem with Multiple Stacks* (DVRPMS).

The DVRPMS is motivated by the case where a single container is not enough to manage all products. Apart from this situation, even if one vehicle was enough to transport all products, the use of multiple containers is interesting because it increases the flexibility of the loading process and can thus lead to a possible cost reduction. This is known in applications of the multiple traveling salesman problem, such as mission planning in the context of autonomous mobile robots or crew scheduling for carrying cash among different branches of a bank (see, e.g., Bektas [13]), where the presence of side constraints may make more profitable the use of several vehicles instead of just one.

Figs. 2 and 3 show two DVRPMS solutions for the situation already depicted in Fig. 1. Fig. 2 shows a solution using four containers, each consisting of two stacks of height two. In this particular case the DVRPMS solution consists of four pairs of tours on the pickup and delivery regions, starting and finishing in their respective depots, allowing feasible vehicle loading plans, so that each tour in the pickup set has its counterpart in the delivery set. In our study we consider the general case in which the fleet is possibly composed of heterogeneous vehicles having different shapes. An example is shown in Fig. 3, where the solution consists of three pairs of tours associated with three containers of size 2×4, 1×4, and 2×2, respectively.

This paper formulates the DVRPMS by proposing first a three-index formulation, and then two set partitioning formulations obtained from a Dantzig–Wolfe decomposition of the former formulation. The two set partitioning formulations are meant to be optimized by column generation. They contain columns having different structure. In the first formulation, each column represents a route collecting some products and delivering them to their respective demanding customers. That is, each column includes a pairing between pickup and delivery customers. This approach follows previous works on one-to-one pickup-and-delivery problems (see, e.g., Ropke and Cordeau [14]). In the second formulation, instead, we independently define columns for the pickup and for the delivery regions. Consequently, we move the pairing and loading constraints from the subproblem to the master problem of the decomposition.

To solve these three formulations we have developed a *branch-and-cut* (BC) algorithm, a *branch-and-price* (BP) algorithm, and a *branch-and-price-and-cut* (BPC) algorithm. Our computational experience analyzes the behavior and performance of these algorithms on a large set of instances, and suggests the cases where it is more appropriate to use each of the three approaches.

To the best of our knowledge no previous work has approached the DVRPMS. However, our study follows a well established line of research on similar complex routing and loading problems. Cordeau et al. [15] proposed a BC algorithm for the pickup-and-delivery traveling salesman problem with LIFO loading, in which a single vehicle with a loading space organized in a single stack performs the loading/unloading operations in mixed order. Later Côté et al. [16] generalized this problem using a vehicle with multiple stacks, proposing new inequalities to manage the LIFO constraints and embedding them into a BC algorithm. Cherkesly et al. [17] have recently proposed exact approaches to the pickup-and-delivery problem with time windows and LIFO loading, developing a BC and a BPC algorithm, as well as a hybrid algorithm combining these two techniques. In the problem that they face, the pickup and the delivery operations are performed in the same region, using a homogeneous fleet of vehicles equipped with a single LIFO stack. The distance-constrained variation of this problem has been approached by Cheang et al. [18] from a heuristic point of view. For what concerns the large literature on combined routing and loading problems, we refer the reader to the recent surveys by Iori and Martello [19,20], whereas for the even larger