Computers & Operations Research 40 (2013) 1085–1093

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

The Pickup and Delivery Problem with Cross-Docking

Fernando Afonso Santos, Geraldo Robson Mateus, Alexandre Salles da Cunha *

Universidade Federal de Minas Gerais, Departamento de Ciência da Computação, Avenida Antônio Carlos, 6627 Belo Horizonte, Brazil

ARTICLE INFO

Available online 5 December 2012

Keywords:
Vehicle routing
Pickup and delivery
Cross-docking
Column generation
Branch-and-price

ABSTRACT

Usual models that deal with the integration of vehicle routing and cross-docking operations impose that every vehicle must stop at the dock even if the vehicle collects and delivers the same set of goods. In order to allow vehicles to avoid the stop at the dock and thus, reduce transportation costs, we introduce the Pickup and Delivery Problem with Cross-Docking (PDPCD). An Integer Programming formulation and a Branch-and-price algorithm for the problem are discussed. Our computational results indicate that optimal or near optimal solutions for PDPCD indeed allow total costs to be significantly reduced. Due to improvements in the resolution of the pricing problems, the Branch-and-price algorithm for PDPCD works better than similar algorithms for other models in the literature.

1. Introduction

Since the pioneering work of Dantzig and Ramser [1] on an optimal distribution of gasoline to gas stations by a truck fleet, the Vehicle Routing Problem (VRP) has become one of the most studied problems in Combinatorial Optimization. Over the years, a large number of studies on other Vehicle Routing Problems that include more complex operating rules and constraints was documented [2]. Among the VRP variants that received significant attention to date, we can cite the Distance-constrained VRP [3,4], the VRP with time windows [5–8], the VRP with pickup and delivery [9,10], and the VRP with Backhauls [11,12], just to name a few.

The need for better solutions for the VRP and its variants motivated, over the past decades, the development of an impressive number of algorithms, both exact[13–19] and heuristic [20–23] As the computational power available increased and the solution techniques improved, more real world applications have shown to significantly benefit from such developments. Practitioners and researchers started to integrate VRP with production planning problems, to tackle even more sophisticated logistic systems. Cross-docking is one of such production systems to which VRP has been integrated with.

Cross-docking (CD) is a recent warehousing technology aimed to reduce inventory costs in supply chain systems [24]. Goods collected by a set of inbound vehicles are delivered at the cross-docking station and after the items are consolidated and grouped at the dock, they are moved to the vehicles responsible for delivering them to their final destinations. A cross-docking station can be seen as a warehouse where a very reduced amount of goods are kept in a short term stock, since the dock does not have long term inventory holding facilities.

In order to operate in such an expedite fashion, a CD system must deal appropriately with complex issues like, for example, how truck loading and unloading operations should be scheduled at the docks [25,26] and how vehicles should be routed to collect and deliver the goods [27–30]. The way goods are collected and delivered is of crucial importance for determining the workload and the time needed to reorganize them at the docks. The more integrated the resolution of these two problems is, the more cost and time effective a cross-docking system should be. Bearing that in mind, a substantial amount of research was dedicated to propose ways to integrate the resolution of these problems. A detailed review of the papers dedicated to this matter could be found in Lee et al. [31], Wen et al. [32], Boysen and Fliedner [26], and Santos et al. [33,34].

As a result of such attempts of integration, Lee et al. [31] proposed the Vehicle Routing Problem with Cross-Docking (VRPCD). In that problem, a fleet of vehicles is in charge of collecting goods from suppliers, delivering them to their final destinations, after loading and unloading operations take place at the CD. The goods are collected and delivered considering time windows constraints. Each time a good is moved from/to a vehicle at the dock, an additional amount of time is needed to implement the operation. The goal in VRPCD is to find routes (satisfying vehicles' capacities and time windows on the nodes) such that all goods are collected and delivered to their final destinations and the total transportation cost is minimized. Later, Santos et al. [33,34] considered a slightly different VRPCD, where time windows were neglected and a cost, to be added in the objective function, is incurred whenever a good is moved from a vehicle to another at the CD.

* Corresponding author. Tel.: +55 31 34095860; fax: +55 31 34095858.
E-mail addresses: fsantos@dcc.ufmg.br (F.A. Santos), mateus@dcc.ufmg.br (G.R. Mateus), acunha@dcc.ufmg.br (A.S. da Cunha).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.11.021

To our knowledge, a common feature of all approaches that have dealt with VRPCD [31–34] is the assumption that vehicles must stop at the CD after the goods are collected from the suppliers. That applies even if the vehicle collects and delivers the same goods. Of course, allowing vehicles to avoid the stop at the CD in such cases may reduce the transportation costs while, at the same time, freeing space and resources at the station.

In this paper, we extend our previous work [33,34] and consider a VRPCD where vehicles are allowed to avoid the stop at the CD. Our model considers two types of routes: pickup and delivery routes [9,35] (when the vehicle does not stop at the CD) and routes that stop at the CD to implement load changes. We name such a problem as The Pickup and Delivery Problem with Cross-Docking (PDPCD). Therefore, the proposed PDPCD is suitable to consider all the problems between a classical Pickup and Delivery Problem [9] and a classical VRPCD [32], if all vehicles stop at the CD. We formulate PDPCD as an Integer Program and implement a Branch-and-price algorithm to solve it.

Compared to other routing models that integrate cross-docking with vehicle routing in the literature [33,34], the introduction of pickup and delivery routes allowed substantial reductions in the transportation costs. For some instances, transportation costs could be reduced by 7.1% when pickup and delivery routes were included in the model. From the computational point of view, the introduction of pickup and delivery routes as well as a better algorithm for pricing the routes that stop at the CD allowed our Branch-and-price implementation to run faster than similar algorithms in Santos et al. [33,34].

The remainder of the paper is organized as follows. In Section 2, we define PDPCD, present an Integer Programming formulation and discuss how it improves on previous models of integration between vehicle routing and cross-docking . A Branch-and-price algorithm for PDPCD is discussed in Section 3. We present our computational results in Section 4 and the paper is closed in Section 5, where we offer some conclusions.

2. Problem definition and Integer Programming formulation

Let $G=(V,A)$ be a directed graph with set of vertices $V=\{0\}\cup S\cup C$, where $S=\{1,\ldots,n\}$ and $C=\{1',\ldots,n'\}$ denote, respectively, sets of n suppliers and n customers and vertex 0 represents the CD. Consider that $P=\{(i,i',q_i):i=1,\ldots,n\}$ denotes a set of n triples, each one representing a demand (or load) $q_i>0$ to be collected from a supplier i and delivered to a customer i'. Consider as well that a homogeneous fleet of K vehicles of capacity Q is available. In the paper, we interchangeably use the terms loads, goods and demands as well as vehicles and routes.

Define costs $\{c_{ij}\geq 0:(i,j)\in A\}$ (satisfying the triangle inequalities) and $\{c_i:i=1,\ldots,n\}$ to be incurred, respectively, when the arcs of G are traversed by the vehicles and when a load q_i is moved from one vehicle to another at the CD. The cost c_i is incurred only once, when load q_i is delivered by a vehicle that does not collect it. PDPCD consists of finding K routes, one for each vehicle, in order to guarantee that each load q_i will be collected from its supplier ($i\in S$) and delivered to its customer ($i'\in C$). The load shipped in a vehicle cannot exceed the capacity Q and the goal is to minimize the transportation costs plus the sum of the costs of changing loads at the CD. Two types of routes are considered

- (Type 1) routes that start at the CD, visit a subset of suppliers, return to the CD, implement load changes at the CD, leave the CD to visit a subset of customers. After visiting the last customer, the vehicle returns empty to the CD. These routes either collect loads that they do not deliver, or deliver loads that they do not collect, or both.
- (Pickup and delivery) routes that start at the CD, visit a subset of suppliers and after the last one is visited, start delivering the collected goods to the customers, without a stop at the CD. Only after the last customer is visited, the vehicle returns empty to the CD.

To further illustrate the differences between the two types of routes, in Fig. 1(a) and (b) we depict two sets of three routes. In Fig. 1(a), only routes of type 1 are used. In Fig. 1(b), routes of both types are considered. Note that, in Fig. 1(b), the route implemented by vehicle k_1 visits customer 6′ right after collecting load q_7 at supplier 7. Therefore, vehicle k_1 delivers and collects the same set of goods and does not stop at the CD.

Depending on the geographical distribution of suppliers and customers and on how loading/unloading costs compare to arc costs, optimal solutions to PDPCD may involve both types of routes or not. If loading/unloading operations are too costly, optimal PDPCD solutions are likely to include more pickup and delivery routes. On the contrary, if load changing costs are zero, routes of type 1 may be selected more frequently.

While PDPCD has not been studied before, VRPCD has received more attention from the literature. Lee et al. [31] and Wen et al. [32] introduced Integer Programming models and Tabu search algorithms for a VRPCD variant where time windows constraints on the nodes are imposed and no costs are incurred when goods change vehicles at the CD. Recently [34,33], we suggested Integer Programming formulations and column generation algorithms for a VRPCD variation that includes loading/unloading costs at the CD but neglects time windows constraints.

In order to model PDPCD as an Integer Program (IP), assume that R denotes the set of routes of type 1 while R_d denotes the set

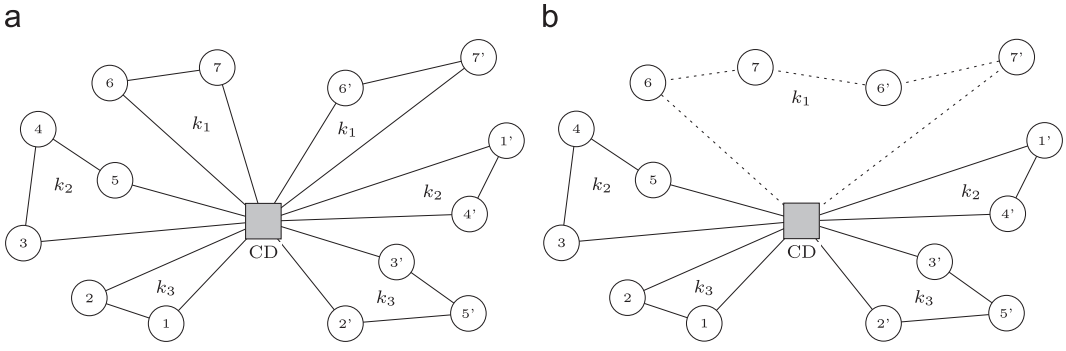

Fig. 1. Differences between possible solutions for VRPCD and PCPCD, with $K=3$, $n=7$. In the figures, k_1, k_2 and k_3 denote routes. (a) One possible solution for VRPCD. (b) One possible solution for PCPCD.