European Journal of Operational Research 206 (2010) 341–349

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing and Logistics

A branch-and-price algorithm for the Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows

Gabriel Gutiérrez-Jarpa [a,b], Guy Desaulniers [c,*], Gilbert Laporte [d], Vladimir Marianov [e]

[a] Graduate Program, Pontifica Universidad Católica de Chile, Santiago, Chile
[b] Department of Industrial Engineering, Universidad Católica de la Santísima Concepción, Concepción, Chile
[c] GERAD and Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal, Montréal, Canada H3C 3A7
[d] GERAD, CIRRELT and HEC Montréal 3000, Chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7
[e] Department of Electrical Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile

A R T I C L E I N F O

Article history:
Received 11 June 2009
Accepted 27 February 2010
Available online 6 March 2010

Keywords:
Transportation
Vehicle routing
Deliveries and selective pickups
Time windows
Branch-and-price
Shortest paths with resources

A B S T R A C T

In the *Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows*, the set of customers is the union of *delivery customers* and *pickup customers*. A fleet of identical capacitated vehicles based at the depot must perform all deliveries and profitable pickups while respecting time windows. The objective is to minimize routing costs, minus the revenue associated with the pickups. Five variants of the problem are considered according to the order imposed on deliveries and pickups. An exact branch-and-price algorithm is developed for the problem. Computational results are reported for instances containing up to 100 customers.

1. Introduction

The purpose of this paper is to develop a branch-and-price algorithm for the *Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows* (VRPDSPTW) defined as follows. Let $G = (V, A)$ be a directed graph where V is the vertex set and A is the arc set. The vertex set is partitioned into $V = D \cup P \cup \{o, o'\}$, where D is the set of *delivery customers*, P is the set of *pickup customers*, and o and o' represent two copies of the *depot* called the source and the sink, respectively. Servicing a customer $i \in D \cup P$ takes s_i time units. A time window $[a_i, b_i]$ is imposed on the start of service at customer i. Every customer $i \in D$ has a delivery demand d_i and every customer $i \in P$ has a pickup demand p_i. All deliveries must be performed, whereas pickups are selective and partial pickups are not allowed. Performing a pickup at customer $i \in P$ yields a revenue u_i. For notational convenience, we define $s_o = s_{o'} = 0$, $d_i = 0$ for all $i \in V \setminus D$, $p_i = u_i = 0$ for all $i \in V \setminus P$, $[a_o, b_o] = [0, 0]$, and $[a_{o'}, b_{o'}] = [0, \bar{b}]$, where 0 and $\bar{b}$ are constants defining unrestrictive windows at vertices o and o'. At the depot, we assume that there is a fleet of identical vehicles of capacity Q sufficient to service all delivery customers. The arc set is given by $A = \{(i, j) : i, j \in V,\ i \neq j,\ a_i + s_i + t_{ij} \leqslant b_j\}$. A travel cost matrix (c_{ij}) and a travel time matrix (t_{ij}) are defined on A. Note that c_{oj} can also include real vehicle fixed costs if any, or very large artificial costs if the number of vehicles used has to be minimized. The VRPDSPTW consists of determining vehicle routes of minimum net cost (travel cost minus revenue) such that (1) all routes start at o and end at o'; (2) all deliveries are performed; (3) the vehicle load along its route never exceeds Q and (4) time windows are respected (vehicles are allowed to wait at a customer location before service starts).

The VRPDSPTW belongs to the class of *one-to-many-to-one* pickup and delivery problems, meaning that all delivery demands originate at the depot and all pickup demands are destined to the depot (Hernández-Pérez and Salazar-González, 2004; Berbeglia et al., 2007). This class of problems includes the *single demand* case, denoted P/D, where delivery and pickup customers are disjoint, and the *combined demand* case, denoted $P\&D$, where the same customer may have a pickup and a delivery. In the latter case, it is possible to create two copies of the same customer (one for the delivery and one for the pickup) and include the copies in D and P. By suitably defining the arcs incident to these copies, one can impose that every customer be visited exactly once (the *single visit* case) or one may let the delivery and pickup operations of a customer to be performed during two separate visits (the *multiple visit* case). Obviously, in the single visit case, the delivery vertex of a

* Corresponding author.
E-mail addresses: ggutierrez@ucsc.cl (G. Gutiérrez-Jarpa), guy.desaulniers@gerad.ca (G. Desaulniers), gilbert@crt.umontreal.ca (G. Laporte), marianov@ing.puc.cl (V. Marianov).

0377-2217/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2010.02.037

customer should always be visited before its pickup vertex. The single demand case as well as the combined demand case with multiple visits can be subdivided into the *mixed route* case in which pickup and delivery vertices may be visited in any order, and the *backhaul* case in which all pickup vertices of a given route must be visited after the delivery vertices of the same route. The latter problem is usually called the *Vehicle Routing Problem with Backhauls* (VRPB). Further details on this classification are provided in Berbeglia et al. (2007) and in Gribkovskaia and Laporte (2008). Our problem statement and algorithm allow the treatment of the five variants just described: *P*/*D* problems with mixed routes; *P*/*D* problems with backhauls; *P*&*D* problems with single visits; *P*&*D* problems with multiple visits allowed and mixed routes, and *P*&*D* problems with multiple visits allowed and backhauls.

One-to-many-to-one pickup and delivery problems arise naturally in reverse logistics operations in which full bottles or containers must be delivered and empty ones are collected (Dethloff, 2001; Tang and Galvão, 2002, 2006; Privé et al., 2006; Hoff et al., 2009). Min (1989) describes an application related to a public library system, whereas Wasner and Zäpfel (2004) consider the case of mail transportation.

For problems without time windows in which all pickups must be performed, an exact branch-and-cut algorithm was described by Baldacci et al. (2003) for the single vehicle case, and a branch-and-price scheme was developed by Dell'Amico et al. (2006) for the combined demand case. Toth and Vigo (1997) have developed a branch-and-bound algorithm for the VRPB, whereas Mingozzi et al. (1999) formulate the problem as an integer linear program and solve it exactly by a commercial solver after having reduced the number of variables.

Numerous heuristics have been proposed for one-to-many-to-one pickup and delivery problems. Since the focus of our paper is on the development of an exact algorithm, we refer the reader to the surveys of Toth and Vigo (2002), Berbeglia et al. (2007) and Parragh et al. (2008) for details about these heuristics.

Relatively little research has been conducted on pickup and delivery problems with selective pickups. Privé et al. (2006) have developed a heuristic for a practical problem involving the delivery of soft-drinks and the collection of empty cans and bottles to and from convenience stores in the Quebec City area. As in our problem, a revenue was associated on selective pickups, and vehicle capacity constraints prevented the collection of all pickups. Gribkovskaia et al. (2008) have applied tabu search to the single vehicle pickup and delivery problem with selective pickups arising in the routing of supply vessels through offshore installations. These installations must be supplied on a regular basis but they also generate pickup demands such as empty containers and waste. Because of the limited ship capacity, it is not always possible to perform all pickup operations and priority must be given to the most important ones (Aas et al., 2007). An exact branch-and-bound algorithm was developed by Süral and Bookbinder (2003) for the single vehicle case and a branch-and-cut algorithm was later proposed by Gutiérrez-Jarpa et al. (2009).

The contributions of this paper are to develop an exact branch-and-price algorithm for the five variants of the VRPDSPTW just introduced, and to allow for selective pickups. To our knowledge, no existing exact algorithms consider time windows or selective pickups, and all address a single variant of the VRPDSPTW. The proposed branch-and-price algorithm can be seen as an extension of one version of the algorithm introduced by Dell'Amico et al. (2006) for the combined demand problem with single visits, no time windows, and no selective pickups. Its applicability to the variants treated in this paper lies in the use of a specific network structure for each variant. This algorithm and the underlying network structures are described in Section 2, followed by computational results in Section 3 and by conclusions in Section 4.

2. Algorithm

Before describing the proposed branch-and-price algorithm, we introduce a generic integer linear program that models all cases of the VRPDSPTW mentioned above. Let R be the set of all feasible vehicle routes. Let c_r be the net cost of route r (computed as the sum of the costs c_{ij} of the arcs in r minus the sum of the revenues u_i of the pickup customers visited), and let δ_{ir} be a binary parameter equal to 1 if and only if route r visits customer i. For each route r, define a binary variable y_r equal to 1 if and only if route r is selected.

With this notation, the VRPDSPTW can be formulated as the following integer linear program:

$$\text{minimize} \quad \sum_{r\in R} c_r y_r \tag{1}$$

$$\text{subject to} \quad \sum_{r\in R} \delta_{ir} y_r = 1, \quad \forall i \in D, \tag{2}$$

$$\sum_{r\in R} \delta_{ir} y_r \leqslant 1, \quad \forall i \in P, \tag{3}$$

$$y_r \in \{0,1\}, \quad \forall r \in R. \tag{4}$$

The objective function (1) minimizes the total net cost of the selected routes. Constraints (2) ensure that each delivery customer is visited by exactly one vehicle, while constraints (3) stipulate that each pickup customer can be visited at most once.

In practice, model (1)–(4) contains a very large number of variables, namely, one for each feasible route. To overcome this difficulty, we propose solving it by means of a branch-and-price algorithm that does not require the explicit enumeration of all variables. Branch-and-price (Barnhart et al., 1998; Desaulniers et al., 1998; Desrosiers and Lübbecke, 2005) consists of a column generation algorithm embedded within a branch-and-bound scheme. Column generation is used to compute lower bounds at each node of the branch-and-bound search tree, while branch-and-bound allows the identification of an optimal integer solution.

In Sections 2.1–2.3, we provide the details of the branch-and-price algorithm used for the VRPDSPTW for the single demand and mixed route case. Section 2.4 discusses the adaptations required to handle combined demands and backhauling.

2.1. Column generation

We now describe the column generation algorithm applied at the root node linear relaxation of model (1)–(4). The adaptation of this algorithm to the other linear relaxations is straightforward. In a column generation context, the linear relaxation is called the *master problem*.

Column generation is an iterative algorithm which solves a *restricted master problem* (RMP) and a subproblem at each iteration. At iteration ℓ, the RMP is simply the master problem restricted to a subset R_ℓ of its variables. It is solved by a linear programming solver (we used the primal simplex algorithm) to provide a pair of optimal primal and dual solutions. The subproblem is an elementary shortest path problem with resource constraints (ESPPRC) whose arc costs depend on the RMP dual solution (see the details below). Its goal is to identify variables (columns) in the set $R \setminus R_\ell$ that have a negative reduced cost with respect to this dual solution. If no such variables exist, then the current RMP optimal primal solution is also optimal for the master problem (setting all y_r variables to 0 for $r \in R \setminus R_\ell$) and the algorithm stops. Otherwise, these columns are added to the current RMP before starting a new iteration. The subproblem is solved by the label-setting algorithm described in Section 2.2.