The Edge Set Cost of the Vehicle Routing Problem with Time Windows

Reinhardt, Line Blander; Jepsen, Mads Kehlet; Pisinger, David

Published in:
Transportation Science

Link to article, DOI:
10.1287/trsc.2015.0620

Publication date:
2016

Document Version
Peer reviewed version

The edge set cost vehicle routing problem with time windows

Line Blander Reinhardt, Mads Kehlet Jepsen, David Pisinger
Department of Management Engineering, Technical University of Denmark, DK-2800 Kgs. Lyngby

March 31, 2014

Abstract

We consider an important generalization of the vehicle routing problem with time windows in which a fixed cost must be paid for accessing a set of edges. This fixed cost could reflect payment for toll roads, investment in new facilities, the need for certifications and other costly investments. The certifications and investments impose a cost for the company while they also give unlimited usage of a set of roads to all vehicles belonging to the company.

This violates the traditional assumption that the path between two destinations is well-defined and independent of other choices. Different versions for defining the edge sets are discussed and formulated. Both the multigraph case and the direct path case are described and MIP-formulations of the problem is presented for both cases. A solution method based on branch-and-price-and-cut is applied to the direct path case. The computational results show that instances with up to 40 customers can be solved in reasonable time, and that the branch-cut-and-price algorithm generally outperforms CPLEX.

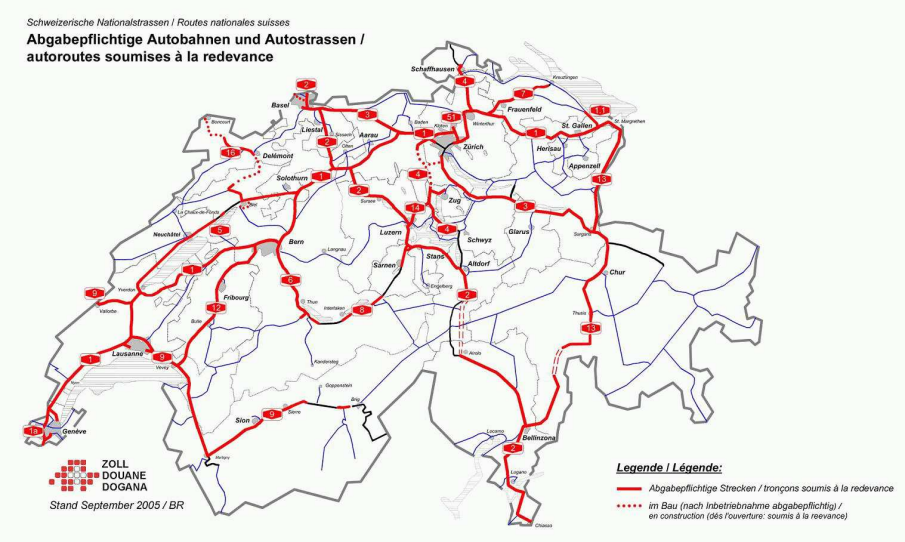


Figure 1: Main road net in Switzerland. To access all the red edges, a vignette needs to be paid. A transportation company may choose to avoid the toll roads and only use the ordinary highways

1 Introduction

In certain real-life situations the cost of a connection does not entirely depend on the cost of the individual links (edges). Frequently, in real life, a fee must be paid by the company for allowing its vehicle to access roads, areas, bridges or other. Such a fee may in some cases only be required to be paid once by

the company and is in such cases independent of the number of vehicles accessing any edge in the set. Companies routing in an area with many ferry connections may pay to access a set of ferries owned by a company at a monthly rate or at a reduced price. Here, it is important to determine which ferry companies it is most profitable to use. The same applies to toll roads and bridges, where some countries charge a company based tax for accessing all freeways in the country (see Figure 1). In war zones or areas of unrest a company may need to get a certification allowing its vehicles to travel on certain protected roads or to enter certain protected zones. Even though in some cases the access is to be paid only for the vehicle accessing the edge set the company will often wish to sign up all its vehicles for robustness and easy administration purposes. Yet another situation where a set of edges can be accessed at a fixed cost is in cases where there is an option of investing in a facility. In such problems, referred to as location-routing problems, there is often a fixed charge connected to a facility and location. Nagy and Salhi [20] give an extensive survey of location-routing problems covering many different routing problems combined with facility location problems. Belenguer et al. [2] recently presented a branch-and-cut method for the location routing problem which apart from considering multiple depots, can be seen as a special case of the model presented in this paper. In all of the mentioned cases there is a fixed charge for accessing a set of edges. For the case shown in Figure 1 there may be both an edge belonging to an edge set and an edge with no additional cost connecting the same pair of nodes. A graph where multiple edges exist between two nodes is called a multi-graph.

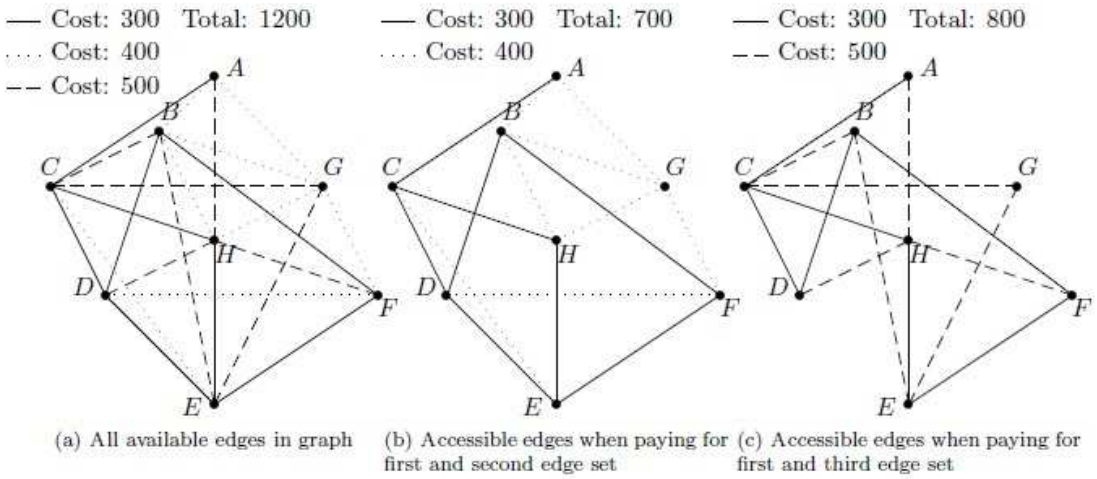
The problem of minimizing the overall cost when planning routes which have a cost associated with sets of edges is in this paper investigated as a generalization of the well known problem of routing vehicles with capacity and service time window restrictions (VRPTW).

In traditional VRPTW problems, it is assumed that the cost of an edge is well-defined and independent of other decisions. This assumption seldom holds in practice, but the literature on VRPTW somehow has overseen this fact.

In the version of the VRPTW considered here the edges of the graph belong to different edge sets. Once the cost of accessing the edge set is paid all vehicles can access the edges in the edge set. However, there might still be a price associated with each of the edges used. Note that the price for accessing the edge set is paid at most once. This cost has an influence on all the routes since once the access price is paid the edges can be accessed by another vehicle without paying the access price again. This makes the cost of the different vehicle routes interdependent. We will denote the considered problem an *edge set vehicle routing problem with time windows* (ESVRPTW). In Figure 2 an example of a network with the edges partitioned into different edge sets is shown. Figure 2 a) shows the entire set of edges, and b) and c) show accessible edges when paying for different combinations of two edge sets. Clearly the ESVRPTW is NP-hard as it is a generalization of the VRPTW. Even though edges refers to bidirectional links and directional arcs generally are used for the VRPTW this does not change the problem as the arcs representing the two directions of the edge are both assigned to the same edge set.

We will in this paper present a general model for the ESVRPTW and a Danzig-Wolfe decomposition. We then consider a simplified version of the problem, where only direct edges between two nodes may be used. This problem occurs when planning transportation of dangerous goods. To avoid unnecessary hazard, the time on the roads should be kept as short as possible, and, in particular, the vehicles are not allowed to enter a city (node) except if they have an actual delivery. The problem also occurs when transporting goods between different countries. In this case the cost of custom clearance depends on the start and origin nodes, and cannot be circumvented by passing through other countries (nodes). In a future paper we will consider the more general version, where multiple edges exist between each pair of nodes (representing alternative paths in the original road graph).

The paper is organized as follows. In the following section we give a rough overview of literature for the vehicle routing problem with time windows and describe relevant results that can be used for solving the ESVRPTW. Section 3 presents a general MIP model for the ESVRPTW and the decomposition of the problem into a master and subproblem is described. In Section 4 we consider a simplified variant of the ESVRPTW where only direct edges between two nodes may be used and a MIP model for this case is presented. In Section 5 the solution method, decomposition, and valid inequalities are described. In Section 6 various extensions of the ESVRPTW model are discussed. In Section 7 the test instances are described. Section 8 reports computational results, and finally the paper is concluded in Section 9 with a



(a) All available edges in graph (b) Accessible edges when paying for first and second edge set (c) Accessible edges when paying for first and third edge set

Figure 2: Accessible edges in a graph; (a) paying for all three edge sets at total cost 1200, (b) paying for first and second edge set at total cost 700, (c) paying for first and third edge sets at total cost 800

discussion of general multi-edge VRP problems.

2 Literature review

To the best of our knowledge, the problem of routing vehicles with an edge set cost has not yet been investigated in the published literature. However, the underlying problem, the vehicle routing problem with time windows (VRPTW), has been extensively studied. The vehicle routing problem was introduced in 1959 by Dantzig and Ramser in [6] as the truck dispatching problem. Many different exact and heuristic methods have been applied to the problem. The basis of the research in this paper is in the exact methods. In 1981 Christofides et al. [4] presented a decomposition generating q-routes for the capacitated VRP. One of the first exact methods for the VRPTW was by Kolen et al. [16] using the ideas presented in [4] and applying them to the VRPTW. This was later included in a branch-and-price method by Desrochers et al [9].

In 1987 a benchmark suite was presented for the VRPTW [22] making it easy to compare solution methods and the research society has been enticed by the problem of solving these tests. Recently there has been a strong development in solution times and problem sizes solved to optimality. In 1999 Kohl et al. [15] and Cook and Rich [5] both applied branch-cut-and-price to the VRPTW.

Some of the most recent developments in solving the VRPTW are described in [1], [8],[12], and [14]. Both Jepsen et al. [12] and Baldacci et al. [1] use the valid cuts summarized by Lysgaard et al. [19] to separate candidate edge sets for branching. Even though the cuts in [19] are implemented for the capacitated vehicle routing problem (CVRP) they may be used for the VRPTW, as the solutions to the VRPTW are a subset of the solution to the corresponding CVRP. Jepsen et al. [12] implemented a branch cut and price algorithm with a label-setting bi-directional algorithm for elementary shortest paths developed by Righini and Salani [21]. Jepsen et al. added the subset-row (SR) inequalities on the master problem variables and modified the subproblem to include the reduced cost from these inequalities. These SR inequalities are included by both Desaulniers et al. [8] and Baldacci et al. [1] in their algorithms.

Desaulniers et al. in 2008 [8] further improved the results by using Tabu search for finding improving routes in the subproblem and generalized the k-path inequalities originally formulated by Kohl et al. [15].

Baldacci et al. [1] introduce an enumeration framework. The master problem is solved using a subgradient optimization algorithm and enumeration is done by solving an ESPPRC where standard dominance is limited. To improve the dominance a lower bound for the completion of each label is found using the so-called *ng-routes*.

In this paper we formulate the ESVRPTW and solve it using the solution method used by Jepsen et al.

[12] on the VRPTW as this method can be easily adapted to solve the ESVRPTW.

3 Multigraph Model

Our mathematical model is based on the 3-index model for the VRPTW presented in Jepsen et al. [12]. In the VRPTW we may pre-calculate the cheapest cost c_{ij} of traveling from node i to node j, and ignore the underlying road network hereafter. This is, as mentioned in section 1, not the case in the general ESVRPTW problem, since the cheapest cost c_{ij}^r depends on the set of edges available, and hence there are multiple paths between two nodes.

In the multigraph version of the ESVRPTW there can be several edges between two nodes in the graph. These edges must belong to different edge sets as two edges between the same two nodes in the same edge set would dominate each other so that one can be removed in preprocessing. Let c_{ij}^r be the edge cost of using the edge (i, j) belonging to a set r. Notice that the edge cost may vary depending on which set r the edge belongs to.

Given the following sets:

R	The set of all edge sets
C	The set of customers, where C_r denotes the customers covered by the edges in set $r \in R$
V	The set of nodes representing the customers in C and the depot defined as 0
$\mathbf{A}$	The set of arcs (i, j) in V, where $\mathbf{A}_r$ is the set of arcs (i, j) belonging to the set $r \in R$
K	The set of vehicles, where $\|K\| \leq \|C\|$ as usual for VRPTW problems

The variables are defined as:

x_{ij}^{rv}	Binary variable indicating if the arc $(i, j) \in \mathbf{A}_r$ is used by vehicle $v \in K$
y_r	Binary variable which is one if an edge from edge set $r \in R$ is used and zero otherwise
t_i^v	The time vehicle v visits $i \in V$.

The parameters are defined as:

D	The capacity of the vehicles
d_i	The demand which must be delivered to node $i \in V$. The demand at the depot is zero
a_i	The availability time for customer $i \in C$. Note that $a_i \geq 0$.
b_i	The required completion time for customer $i \in C$ with $b_i \geq a_i$
θ_{ij}^r	The time it takes to travel from $i \in C$ to $j \in C$ on arc $(i, j) \in \mathbf{A}_r$.
c_{ij}^r	The cost of using an arc $(i, j) \in \mathbf{A}_r$
c_r	The cost of accessing the arcs in edge set $r \in R$

A model (**MGM**) for the multigraph case of the ESVRPTW can now be formulated. Since the problem is a generalization of the VRPTW the model presented here for the ESVRPTW is the standard VRPTW model presented in the survey by Kallehauge [13], with an additional set of constraints used to formulate the edge set costs. The cost of the edge sets are inserted into the objective. In the presented model the assumption is that each edge belongs to exactly one edge set, however, alternatives to this assumption are discussed in Section 6.

$$\textbf{MGM}: \qquad \textbf{min} \sum_{r \in R} \sum_{v \in K} \sum_{(i,j) \in \mathbf{A}_r} c_{ij}^r x_{ij}^{rv} + \sum_{r \in R} c_r y_r \tag{1}$$

$$
\begin{aligned}
s.t. \quad & y_r - \sum_{v \in K} x_{ij}^{rv} \geq 0 && \forall r \in R,\ (i,j) \in \mathbf{A}_r && (2)\\
& \sum_{r \in R} \sum_{v \in K} \sum_{j:(i,j) \in \mathbf{A}_r} x_{ij}^{rv} = 1 && \forall i \in C && (3)\\
& \sum_{r \in R} \sum_{i \in C} x_{i0}^{rv} = \sum_{r \in R} \sum_{i \in C} x_{0i}^{rv} && \forall v \in K && (4)\\
& \sum_{r \in R} \sum_{j:(j,i) \in \mathbf{A}_r} x_{ji}^{rv} - \sum_{r \in R} \sum_{j:(i,j) \in \mathbf{A}_r} x_{ij}^{rv} = 0 && \forall i \in C, \forall v \in K && (5)\\
& \sum_{r \in R} \sum_{(i,j) \in \mathbf{A}_r} d_i x_{ij}^{rv} \leq D && \forall v \in K && (6)\\
& a_i \leq t_i^v \leq b_i && \forall\, i \in V,\, v \in K && (7)\\
& (t_i^v + \theta_{ij}^r) x_{ij}^{rv} - t_j^v \leq 0 && \forall\ v \in K,\, r \in R,\, (i,j) \in \mathbf{A}_r && (8)\\
& x_{ij}^{rv} \in \{0,1\} && \forall\, r \in R,\, (i,j) \in \mathbf{A}_r,\, v \in K && (9)\\
& y_r \in \{0,1\} && \forall\ r \in R && (10)\\
& t_i^v \in \mathbb{R}_0^+ && \forall\, i \in V,\, v \in K && (11)
\end{aligned}
$$

The objective (1) is the sum of the cost on the edges used and the sum of the cost of accessing the sets of the edges used by the vehicles. The constraints (2) ensure that if an edge, in a edge set, is used then the cost of accessing the edge set is paid. Note that due to the minimization objective and the constraints (2) the integrality of the x_{ij}^{rv} variables implies integral y_r variables. Constraints (3) ensure that every customer is visited. Constraints (4) ensure that all vehicles start and end their journey at the depot. Constraints (5) ensure that vehicles arriving at a customer also leaves the same customer. Constraints (6) ensure that the capacity of a vehicle is not exceeded. Constraints (7) ensure that customers are visited in their respective time window. Finally, constraints (8) find the time of vehicle v at customer i. If vehicle v does not visit customer i any time can be chosen. Constraints (8) also ensures that the route is simple. The variables x_{ij}^v and y_r are in (9) and (10) defined to be binary and the variable t_i^v is in (11) defined to be a positive real number. Note that constraints (8) can be replaced by the linear constraints:

$$
t_i^v + \theta_{ij}^r - t_j^v \leq M(1 - x_{ij}^{rv}) \qquad \forall\ v \in K,\, r \in R,\, (i,j) \in \mathbf{A} \qquad (12)
$$

Where M must be selected so that it is greater than the duration of any route.

3.1 Tightening the edge set constraints

In the ESVRPTW each customer must be visited exactly once. This requirement is ensured by constraints (3) and can be used to tighten the constraints in (2). Since each customer is visited once, we know that if several edges belonging to the same edge set leave the same customer then at most one of them can be used, and if one of them is used then the cost of the edge set must be accounted for. See Figure 3 for an illustration.

From this observation we can construct the constraints:

$$
\sum_{v \in K} \sum_{j \in C:(i,j) \in \mathbf{A}_r} x_{ij}^{rv} \leq y_r \qquad \forall r \in R, i \in C \qquad (13)
$$

In this case the integrality of the x variables again imposes the integrality of the y variables and the number of constraints in (13) is $|C||R|$. Note that constraints (13) do not apply to the depot as more than one edge belonging to an edge set may leave the depot. Therefore the constraints of type (13) cannot entirely replace the constraints (2). However, by formulating a new edge set of constraints similar to the constraints (13) for edges entering every customer:

$$
\sum_{v \in K} \sum_{j \in C:(j,i) \in \mathbf{A}_r} x_{ji}^{rv} \leq y_r \qquad \forall r \in R, i \in C_r, \qquad (14)
$$

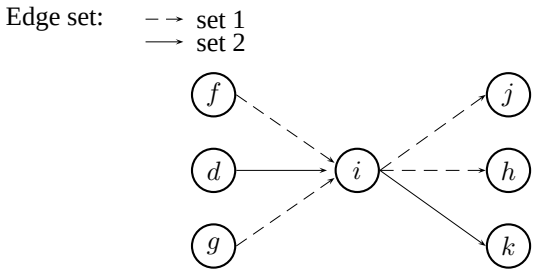


Figure 3: Illustration of the bound on outgoing edges. At most one of the arcs (i, j) and (i, h) can be used, therefore y_1 must always be greater than the sum of the outgoing arcs of i in edge set 1.

then all edges will be covered by the two constraints (13) and (14) and they can therefore replace the constraints (2). Thereby the constraints (2) can be replaced by $2|C||R|$ constraints or less. These tighter constraints will in Section 5 and onwards replace constraints (2) in the model.

4 Direct path model

As mentioned in Section 1 an important variant of the ESVRPTW occurs when transporting dangerous goods or when transporting goods between countries. In this case only the direct connection between nodes i and j may be used. Letting c_{ij} denote the cost of using arc $(i, j) \in \mathbf{A}$ we get the model (**DPM**):

$$\textbf{DPM}: \qquad \textbf{min} \sum_{v \in K} \sum_{(i,j) \in \mathbf{A}} c_{ij} x_{ij}^v + \sum_{r \in R} c_r y_r \tag{15}$$

$$s.t. \quad \sum_{v \in K} \sum_{j:(j,i) \in \mathbf{A_r}} x_{ji}^v \leq y_r \qquad \forall r \in R, i \in C_r \tag{16}$$

$$\sum_{v \in K} \sum_{j:(i,j) \in \mathbf{A_r}} x_{ij}^v \leq y_r \qquad \forall r \in R, i \in C_r \tag{17}$$

$$\sum_{v \in K} \sum_{j:(i,j) \in \mathbf{A}} x_{ij}^v = 1 \qquad \forall i \in C \tag{18}$$

$$\sum_{i \in C} x_{i0}^v = \sum_{i \in C} x_{0i}^v \qquad \forall v \in K \tag{19}$$

$$\sum_{j:(j,i) \in \mathbf{A}} x_{ji}^v - \sum_{j:(i,j) \in \mathbf{A}} x_{ij}^v = 0 \qquad \forall i \in C, \forall v \in K \tag{20}$$

$$\sum_{(i,j) \in \mathbf{A}} d_i x_{ij}^v \leq D \qquad \forall v \in K \tag{21}$$

$$a_i \leq t_i^v \leq b_i \qquad \forall\, i \in V, v \in K \tag{22}$$

$$(t_i^v + \theta_{ij}) x_{ij}^v - t_j^v \leq 0 \qquad \forall\ v \in K,\ (i,j) \in \mathbf{A} \tag{23}$$

$$x_{ij}^v \in \{0, 1\} \qquad \forall\, (i,j) \in \mathbf{A},\ v \in K \tag{24}$$

$$y_r \in \{0, 1\} \qquad \forall\ r \in R \tag{25}$$

$$t_i^v \in \mathbb{R}_0^+ \qquad \forall\, i \in V,\ v \in K \tag{26}$$

Note, that the main difference between the model **MGM** and the model **DPM** is that the r index on the variable x_{ij}^{rv}, cost parameter c_{ij}^r and time parameter θ_{ij}^r is removed in the latter model. Here the tighter constraints (13) and (14) are replacing the constraints (2) in the above model.

As before the constraints (23) can be replaced by linear constraints:

$$t_i^v + \theta_{ij} - t_j^v \leq M(1 - x_{ij}^v) \qquad \forall\ v \in K,\ (i,j) \in \mathbf{A} \tag{27}$$

Where M is a large constant greater than the duration of any route.

5 Solution method

The branch-cut-and-price method has with success been applied to the VRPTW and the best results for finding exact solutions to the problem have been produced using this method (see e.g. Jepsen et al. [12], Desaulniers et al. [8] and Baldacci et al. [1]). Since the ESVRPTW is a generalization of the VRPTW the solution methods for the VRPTW may successfully be applied to the direct path case of the ESVRPTW (model DPM). Therefore we have applied the BCP algorithm to the **DPM** using cuts for the original formulation of the VRPTW presented by Fukasawa et al. in [11] and by Lysgaard et al. [19] for the CVRP. This corresponds to the algorithm developed by Jepsen et al. [12] for the VRPTW. Jepsen et al. also introduced the Subset Row valid inequalities into the master problem formulation. We will later argue that these cuts can with the same benefits be applied to the **DPM**.

Both models for the ESVRPTW can be decomposed into a master and pricing problem similar to the Dantzig-Wolfe decomposition of the standard model for the VRPTW, where the pricing problem is to find an elementary shortest path problem with some resource constraints. In Sections 5.1 to 5.5 the solution method for the **DPM** will be described. In Section 5.6 it is discussed how this decomposition and solution method can be applied to the **MGM**.

5.1 Master Problem

The master problem (**DPMM**) for ESVRPTW is similar to the standard VRPTW master problem using the decomposition presented by Desrochers et al. [9]. However, the cost of the edge sets are kept in the master problem and these costs will be reflected in the dual variables from the solution of the linearly relaxed master problem.

$$\textbf{DPMM}: \qquad \textbf{min:} \sum_{p \in P} \sum_{(i,j) \in \mathbf{A}} c_{ij}\alpha_{ijp}\lambda_p + \sum_{r \in R} c_r y_r \tag{28}$$

$$s.t. \quad \sum_{p \in P} \sum_{j:(j,i) \in \mathbf{A}_r} \alpha_{jip}\lambda_p \leq y_r \qquad \forall i \in C, \forall r \in R \tag{29}$$

$$\sum_{p \in P} \sum_{j:(i,j) \in \mathbf{A_r}} \alpha_{ijp}\lambda_p \leq y_r \qquad \forall i \in C, \forall r \in R \tag{30}$$

$$\sum_{p \in P} \sum_{i:(i,j) \in \mathbf{A}} \alpha_{ijp}\lambda_p = 1 \qquad \forall j \in C \tag{31}$$

$$\lambda_p \in \{0,1\} \qquad \forall\, p \in P \tag{32}$$

$$y_r \in \{0,1\} \qquad \forall\ r \in R \tag{33}$$

The set P contains routes satisfying the time window constraints and the capacity constraints. When λ_p is one then route $p \in P$ is used and λ_p is zero otherwise. The constant α_{ijp} is one if the edge $(i,j) \in \mathbf{A}$ is used by the route p and zero otherwise. Constraints (29) and (30) corresponds to the constraints (13) and (14) which ensure that access to the edge set is paid if an edge from the edge set is used in one of the selected routes. Constraints (31) ensure that every customer is visited exactly once by the set of routes selected. The master problem can be recognized as a set partitioning problem with side constraints. It is important to note that the constraints (29) and (30) do not change the domain of valid solutions but only affect the value of the solutions.

5.2 Pricing problem

The linear relaxation of the master problem can be solved through delayed column generation. The pricing problem then becomes an elementary shortest path problem with resource constraints. Let $\phi'_{ir} \in \mathbb{R}_0^-$ be the dual variables of constraints (29) and let $\phi_{ir} \in \mathbb{R}_0^-$ be the dual variables of constraints (30). Let $\pi_j \in \mathbb{R}$ for $j \in C$ be the dual variables of constraint (31) and let $\pi_0 = \phi'_{0r} = \phi_{0r} = 0$. Then, the reduced cost for a route in the pricing problem becomes:

$$\bar{c}_p = \sum_{(i,j)\in\mathbf{A}} c_{ij}\alpha_{ijp} - \sum_{(i,j)\in\mathbf{A}} \pi_j\alpha_{ijp} - \sum_{r\in R}\sum_{(j,i)\in\mathbf{A}_r} \phi'_{ir}\alpha_{jip} - \sum_{r\in R}\sum_{(i,j)\in\mathbf{A}_r} \phi_{ir}\alpha_{ijp} \tag{34}$$

$$= \sum_{(i,j)\in\mathbf{A}} (c_{ij} - \pi_j)\alpha_{ijp} - \sum_{r\in R}\sum_{(i,j)\in\mathbf{A}_r} (\phi_{ir} + \phi'_{jr})\alpha_{ijp} \tag{35}$$

This can be transformed to a elementary shortest path problem with resource constraints (ESPPRC) where each edge (i,j) has the cost $\bar{c}_{ij} = c_{ij} - \pi_j - \sum_{\{r|(i,j)\in\mathbf{A}_r\}}(\phi_{ir} + \phi'_{jr})$. The resource constraints in our problem are the capacity constraint and the time window constraints. Per definition of the elementary shortest path problem, each customer is visited at most once. A label setting algorithm can be used to solve the problem to optimality.

The label setting algorithm will at each node have a set of labels representing a path from the depot to the node. A label is in our implementation represented by a triple containing: a pointer to the label of the previously visited node, a number that indicates the time consumption, and a number indicating the capacity consuption.

To reduce the number a dominance test is used to remove labels not leading to the optimal solution. Let $\bar{c}(L)$ be the cost of the path represented by label L and let $\bar{d}(L)$ and $\bar{t}(L)$ be respectively the capacity and time consumption. Let v be the node and let $\eta(L)$ be the set of feasible partial paths from v to the depot. A label L' is dominated by another label L'' if:

$$\bar{c}(L'') \leq \bar{c}(L') \tag{36}$$

$$\bar{d}(L'') \leq \bar{d}(L') \tag{37}$$

$$\bar{t}(L'') \leq \bar{t}(L') \tag{38}$$

$$\eta(L') \subseteq \bar{c}(L'') \tag{39}$$

We use the state of art methods for solving the ESPPRC based on the bidirectional shortest path algorithm presented by Righini and Salani [21].

The dominance criteria developed and presented by Desaulniers et al. [7] for removing all labels which are not efficient Pareto optimal, given that the resources are additive or the function on them is monotone, can be used here. However, when introducing the SR cuts developed by Jepsen et al. [12] the objective is no longer additive and the function used is not monotone. In [3] Blander Reinhardt and Pisinger cover several different ESPPRCs with objectives containing functions which are not monotone.

Before running the label setting algorithm, a time window reduction, as described by Desrochers et al. [9], is performed to speed up the label setting algorithm. The ESPPRC is NP-hard in the strong sense (Dror [10]), therefore it is desirable to try to find improving paths without solving the ESPPRC to optimality. For this purpose a simple heuristic has been implemented. Only if the heuristic does not find any improving paths, the ESPPRC is solved to optimality using a label setting algorithm. The heuristic is based on a greedy approach, always extending the label with the lowest cost.

5.3 Branch-cut-and-price

Branch-cut-and-price algorithms have with success been applied to several transportation problems, hence we solve the **DPMM** using this approach. The branch-cut-and-price algorithm runs through the following steps, starting with the root node as the only unprocessed node:

Step 1: Choose an unprocessed node. If several unprocessed nodes exist the node with the lowest lower bound is selected. If the lower bound of a node is above the upper bound then the node will be removed from the unprocessed node list.

Step 2: Solve the LP relaxed master problem. Update the lower bound.

Step 3: Search for routes with negative reduced cost using heuristic methods. If any route is found with negative reduced cost then add columns to the master problem and go to step 2. The heuristic used is a simple greedy approach.

Step 4: Solve the pricing problem, formulated as an elementary shortest path problem with resource constraints, to optimality. If routes with negative reduced cost are found then add them to the master problem and go to step 2. If no routes with negative reduced cost are found then the lower bound of the node is updated. If the updated lower bound is above the global upper bound then the node is deleted. Go to step 1.

Step 5: If any violated cuts are found then add them to the master problem and go to step 2.

Step 6: Mark the node as processed. If the solution to the LP relaxed master problem is integral then update the upper bound. If the solution is fractional then branch and add the children to the list of unprocessed nodes. Go to step 1.

Note that time window reduction, as described by Desrochers et al. [9], is applied in the pricing problem. This will also eliminate infeasible arcs. The applied branching rules are described in more details in the next subsection.

5.4 Cuts

As can be seen from the algorithm in Section 5.3, cuts are added to the master problem in Step 5 of the algorithm. If the added cuts are valid inequalities derived from the original formulation, the **DPM**, then the dual can be transferred directly to the cost of the arcs. Such cuts could be capacity inequalities, strengthened capacity inequalities, framed capacity inequalities, strengthened comb inequalities, multi star inequalities and generalized large multi star inequalities. However, if the cuts added are expressed in the path variables the dual cost can be more complicating to transfer as the dual of the constraints may be activated not only by a single edge but a combination of edges. However, the subset row cuts have with success been introduced into the master problem variables by Jepsen et al. [12]. The authors also developed a method of handling the reduced cost of a route for the ESPPRC where the objective function is not strictly in- or de- creasing as a result of the reduced cost achieved from the Subset Row cuts.

5.4.1 Valid Inequalities in the original form

Many valid inequalities have been developed for the VRPTW. These valid inequalities will also be applicable for ESVRPTW as the ESVRPTW does not change the set of feasible solutions but only changes the objective function. Valid inequalities for the VRPTW are described in e.g. [11], [17] and [19]. The valid inequalities in the original form applied are, as mentioned previously, the capacity inequality, the strengthened capacity inequality, the framed capacity inequality, the strengthened comb inequality, the multi star inequality and the generalized large multi star inequality. These inequalities have all been developed for the capacitated vehicle routing problem CVRP but also apply to the VRPTW. However, since they are developed for the CVRP they do not include the time window restrictions to possibly tighten inequalities. The separation algorithm used in our implementation is the one described by Lysgaard et al. [19] and accessible in the framework developed by Lysgaard [18].

5.4.2 Valid Inequalities expressed using the master problem variables

Jepsen et al. [12] introduced the Subset-Row (**SR**) inequalities to generate cuts in the set partitioning formulation of the master problem with the path variables λ. The SR inequalities are inspired by the clique and odd hole inequalities for the set-packing problem. The undirected conflict graph $G'(P, E)$ is defined as follows: Each column is a node in G' and an edge in the graph G' represents a conflict between the two nodes (columns) it connects. A conflict appears, in the case of VRPTW and also ESVRPTW, if two columns p and q both have an edge leaving the same customer. This means that the nodes representing p

and q in the conflict graph G' has an edge connecting them if there is a customer which is visited by both the path in p and the path in q.

Clearly for a fully connected subgraph $\hat{P} \in G'$ (also called a clique) it is true that

$$\sum_{p \in \hat{P}} \lambda_p \leq 1$$

Because otherwise two conflicting nodes (columns) would be selected.

For a cycle of nodes $\bar{P}$ in the conflict graph G' it must hold that

$$\sum_{p \in \bar{P}} \lambda_p \leq \left\lfloor \frac{|\bar{P}|}{2} \right\rfloor$$

as two neighboring nodes cannot be visited without introducing a conflict in the original problem.

Since restricting a node visit corresponds to a row (constraint) in the master problem then a conflict in the conflict graph corresponds to one or more conflicting rows. This lead to the name *subset row inequalities*.

The inequalities are not based on the edges directly but on the route variables and formulated as follows:

$$\textbf{SR}: \quad \sum_{p \in P} \left\lfloor \frac{1}{k} \sum_{i \in S} \alpha_{ip} \right\rfloor \lambda_p \leq \left\lfloor \frac{|S|}{k} \right\rfloor \tag{40}$$

Where S is a subset of the constraints (31), k is a positive integer less than or equal to $|S|$ and

$$\alpha_{ip} = \sum_{(i,j) \in \delta^+(i)} \alpha_{ijp}$$

The SR inequalities are cuts on the λ_i variables which are used to cut away some of the region which is feasible in the linear relaxation of the master problem but not feasible to the integer master problem.

The inequalities (40) are clearly not affected by the constraints (29) and (30) and therefore they can also be introduced into the master problem for the ESVRPTW presented in Section 5.1. Moreover the effect of introducing the SR cuts into the ESVRPTW should be the same as in the VRPTW as the set of feasible solutions does not differ between the ESVRPTW and the VRPTW.

The problem with these inequalities is that the dual of each inequality can not be mapped directly to the cost of the individual edges. Using the notation from Jepsen et al. [12] we let the dual variable of an SR inequality be σ. We can then formulate the dual cost of a route p as

$$\hat{c}_p = \bar{c}_p - \sigma \left\lfloor \frac{\left(\sum_{i \in S} \sum_{(i,j) \in \delta^+(i)} \alpha_{ijp} \right)}{k} \right\rfloor$$

Note that the dual variable σ is not activated before at least k nodes in the set S have been visited.

Therefore to introduce the reduced cost of these constraints into the pricing problem the ESPPRC needs to be modified. As mentioned in Section 5.2, the ESPPRC is solved with a label-setting algorithm using dominance for the time, load and cost criteria. Let L be a label at a node v so that each label L at v represents a path from the depot to v. The usual dominance criteria which holds for additive costs is that a label is dominated if there exists another label at the same node where all criteria are less than or equal to the dominated labels criteria values. However, this does not hold for the reduced cost introduced by the SR inequalities. One label may be better than the other even if the labels have the same or worse cost. In our implementation we use $k = 2$ and $|S| = 3$, meaning that we include cuts which do not allow two or more routes to visit two nodes from a set of three customers.

To solve the ESPPRC with the dual cost from the SR inequalities, Jepsen et al. [12] modified the dominance rule for the cost criteria in the label setting algorithm which is used when solving the ESPPRC.

In the case where the cost of cut q has been accounted for in a label L_i and not accounted for in a dominated label L_j, the dominance rule for the cost of two labels at the same node becomes:

$$\hat{c}(L_i) - \sum_{q \in Q} \sigma_q \leq \hat{c}(L_j) \tag{41}$$

where Q represents a subset of the SR cuts for which the route represented by L_i has visited two nodes in the SR cut, and the route represented by L_j has not visited two nodes in the SR cut, and $\sigma_q < 0$. The dominance rules for the cost, capacity and time constraints are the standard dominance rules given by Desaulniers et al. [7] as the functions are nondecreasing. For further details see Jepsen et al. [12].

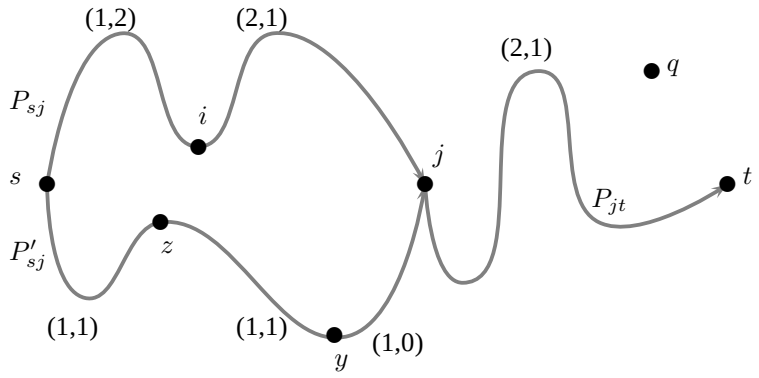


Figure 4: Assume that an edge have two associated values (cost, time). Let there be a valid SR cut where S contains 3 elements and $k = 2$. The set S contains nodes z, t, q. Let us assume that the dual cost $\sigma = -1$. At node j the label for the path sij is (3,3) and for the path szj the label is (3,2), when not considering the SR dominace rule (41). Clearly when not considering the SR dominance the path szj would dominate the path sij. When using the SR dominace rule (41) the labels at j becomes (3,3) for path sij and (4,2) for path szj. On the path $szjt$ two nodes of the set S are visited. Therefore at node t the actual cost and time of the two paths is (5,4) for $sijt$ and (6,3) for path $szjt$. Clearly these two paths cannot dominate each other.

5.5 Branching

In most algorithms for the VRPTW branching is performed on the arc variables remaining after preprocessing. We have chosen to do branching on the edge set variables as well. Since the number of edge set variables is comparably small, branching on the edge set variables can significantly reduce the number of free arc variables at a node and thereby reduce the number of nodes in the search tree. In addition, the branching of Fukasawa et al. [11] is implemented which branches on the number of vehicles servicing a set of nodes/customers. This branching can be formulated by letting $S \subset C$ be a subset of the set of arcs remaining after preprocessing and one branch is $\sum_{v \in K} \sum_{(i,j) \in \delta^+(S)} (x^v_{ij} + x^v_{ji}) = 2$ where $\delta^+(S)$ is the edges leaving the set S and the other branch where at least two vehicles services the set S is represented by constraints $\sum_{v \in K} \sum_{(i,j) \in \delta^+(S)} (x^v_{ij} + x^v_{ji}) \geq 4$. To separate candidate sets the Lysgaard cut library [18] is used. From preliminary tests it was clear that branching on the edge set variables tended to improve the solution time for the problem significantly.

5.6 Multigraph Version

In the multigraph version of the ESVRPTW there can be several edges between two nodes in the graph. These edges must belong to different edge sets as two edges between the same two nodes in the same edge set would dominate each other so that one can be removed in preprocessing. In this section the

decomposition of the path based model for the multigraph case of the ESVRPTW is presented (see model **MGM**).

Using a similar branch-cut-and-price method as presented for the direct path ESVRPTW, the path based master problem (**MGMM**) can be formulated as:

$$\textbf{MGMM}: \qquad \textbf{min:} \sum_{r\in R}\sum_{p\in P}\sum_{(i,j)\in \mathbf{A}} c^r_{ij}\alpha^r_{ijp}\lambda_p + \sum_{r\in R} c_r y_r \tag{42}$$

$$s.t. \quad \sum_{p\in P}\sum_{j:(j,i)\in \mathbf{A}_r} \alpha^r_{jip}\lambda_p \le y_r \qquad \forall i \in C, \forall r \in R \tag{43}$$

$$\sum_{p\in P}\sum_{j:(i,j)\in \mathbf{A_r}} \alpha^r_{ijp}\lambda_p \le y_r \qquad \forall i \in C, \forall r \in R \tag{44}$$

$$\sum_{r\in R}\sum_{p\in P}\sum_{j:(j,i)\in \mathbf{A}} \alpha^r_{jip}\lambda_p = 1 \qquad \forall i \in C \tag{45}$$

$$\lambda_p \in \{0,1\} \qquad \forall\, p \in P \tag{46}$$

$$y_r \in \{0,1\} \qquad \forall\ r \in R \tag{47}$$

Note that the only difference between the above model and **DPMM** is the r index of the α^r_{jip} parameter and the c^r_{ji} cost parameter.

As for the direct path ESVRPTW the linear relaxation of the multigraph ESVRPTW master problem **MGMM** can be solved through delayed column generation with ESPPRC as the pricing problem.

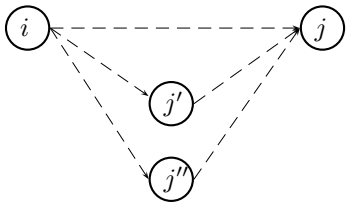


Figure 5: Transformation of a multi-graph to an ordinary graph in the ESPPRC subproblem. Nodes i and j have three intermediate edges in the multi-graph so we introduce j', j'' as auxiliary nodes. The cost of the edges $c_{ij}, c_{ij'}$ and $c_{ij''}$ is the original cost of the first to the third edge between i and j in the multigraph. The same applies to the travel time, where $\theta_{ij}, \theta_{ij'}$ and $\theta_{ij''}$ is the original travel time on the three edges. Note that $c_{j'j}$, $c_{j''j}$, $\theta_{j'j}$ and $\theta_{j''j}$ are all zero.

When solving the ESPPRC we transform the multigraph to an ordinary graph by using an auxiliary node for each extra connection between two nodes. A graphical example of this is shown in Figure 5. The edge (i, j) is the first edge in the multigraph and the paths (i, j', j) and (i, j'', j) corresponds to the second and third edge between the nodes i and j.

In a future paper we will extend the presented cuts to the multigraph version of the problem and implement them in a branch-cut-and-price framework.

6 Closely related formulations and problems

Several variants of the ESVRPTW exist. So far it has been assumed that edges only belong to one edge set, however situations occur where an edge may belong to more than one edge set. If an edge may belong to more than one edge set, the charge for using it can follow various rules, as discussed in the sequel.

In the simplest variant, the cost of using an edge must be paid for all the edge sets the edge belongs to. This case occurs if several pieces of equipment are needed to use the edge, or if several permissions are

needed to cross the edge. This problem can be formulated using constraints (13) and (14) with the only difference that an edge may be included in more than one constraint for each node.

6.1 Edges belonging to multiple edge sets

Assume that for an edge belonging to several edge sets, the access cost only needs to be paid for one of the edge sets containing the edge. This occurs when various bundled discount deals give access to the same connection. In such a case it is important to determine which deal to pay for so that the least money is spent overall. This variant of the problem can be modeled by replacing constraints (13) and (14) with the following constraints:

$$\sum_{v \in K} x_{ij}^v - \sum_{\{r | r \in R \wedge (i,j) \in \mathbf{A}_r\}} y_r \leq 0, \quad \forall (i,j) \in \mathbf{A}, \tag{48}$$

The constraint is very similar to constraints (2); however, in this case the integrality of the x_{ij}^v variables does not necessarily imply integrality of the y_r variables. Note that when replacing constraints (29) and (30) with:

$$\sum_{p \in P} \alpha_{ijp} \lambda_p \leq \sum_{\{r | r \in R \wedge (i,j) \in \mathbf{A}_r\}} y_r, \quad \forall (i,j) \in \mathbf{A}, \tag{49}$$

representing the (48) constraints in the master problem form then each edge (i, j) in the ESPPRC subproblem will have cost $c_{ij} - \pi_j - \rho_{ij}$ where ρ is the dual variable for the constraints (48). This will not add any complications to the ESPPRC algorithm as the cost of a path remains additive and the non additive cost introduced by the SR cuts are handled as in the VRPTW.

7 Test data

Following the tradition of the VRP, the test data are based on the Solomon instances [22] making it possible to relate our results to the existing literature. We have generated test instances based on the RC201 to RC204 instances by assigning subsets of edges to disjoint edge sets, and associating a fixed cost with each edge set. For the RC201 to RC204 Solomon instances different categories of test instances have been constructed. The instances can be grouped into two categories:

1. **random edge sets**: In these instances, the edges in each edge set are randomly selected. These instances should reflect a toll on accessing bridges, tunnels or ferries. These facilities are randomly scattered in the plane, but frequently a set of facilities is run by the same operator.

2. **spanning tree edge sets**: In these instances the selected edges form cheap spanning graphs of a randomly selected subset of nodes. Each subset of nodes consists of half of the total number of nodes. This case should reflect payment of toll on motorways. Motorways usually form a spanning network covering the main cities in a country.

In all test cases, each edge is assigned to at most one edge set. Moreover exactly one edge connects each pair of nodes. Thus the test cases all represent the direct path case of the ESVRPTW.

For each Solomon instance, test instances containing 3, 5 and 8 edge sets were generated, each having an associated cost for accessing the edge set.

For the **random edge sets** instances, 50% of the edges are assigned to edge sets with an additional cost. For each combination of Solomon instance and number of edge sets, test instances were generated with the costs of an edge set calculated as $\beta = 5\%$ of the average cost of the edges in the edge set multiplied by the number of nodes cover by edges in the edge set.

In the case of **spanning tree edge sets**, the cost of a given edge set is chosen as the most expensive edge in the graph minus the average value of the edges in the given edge set. This implies that edge sets containing cheaper spanning trees (i.e. fast transportation times) are more costly than the edge sets containing more expensive spanning trees.

The construction of the edge sets and costs is in both cases generated so that the edge sets are likely to be attractive. For the spanning tree edge sets the cost function is selected differently than from the random

instance		opt solution		solution times				
test	$\|R\|$	objective	R^*	CPLEX	bcp C+SR	Bp	bcp SR	bcp C
rc201	3	4239	3	*0.06	1.22 (3)	1.16(1)	1.22(3)	1.20(2)
rc201	5	4539	4	*0.06	1.09(1)	1.23(4)	1.18(3)	1.16(2)
rc201	8	4878	5	*0.09	1.62(2)	1.52(1)	1.66(3)	1.71(4)
rc202	3	3723	2	473.82	*13.55(1)	14.86(2)	16.85(3)	21.76(4)
rc202	5	4120	3	200.19	10.26(2)	*9.40(1)	11.71(4)	11.51(3)
rc202	8	4166	3	25.54	*9.96(1)	10.13(2)	17.06(4)	11.75(3)
rc203	3	3371	1	-	*29.07(1)	29.19(2)	30.30(3)	54.01(4)
rc203	5	3635	2	-	*44.34(1)	44.66(2)	47.71(3)	58.24(4)
rc203	8	3545	2	-	*75.54(1)	81.25(2)	88.53(3)	100.63(4)
rc204	3	3148	1	-	*130.83(1)	147.04(3)	183.59(4)	136.88(2)
rc204	5	3414	2	-	*123.00(1)	205.80(3)	153.72(2)	308.24(4)
rc204	8	3215	1	-	100.40(3)	*43.06(1)	119.16(4)	54.43(2)
Average Rank					1.50	2.0	3.25	3.17

Table 1: RC201-RC204 instances with 20 customers, **random edge sets**. $|R|$ is the number of edge sets, while R^* is the number of selected edge sets in an optimal solution. If the algorithm has not terminated within 7500 seconds it is indicated with "-". The best running time for each instance is marked with a "*". C indicates that the cuts implemented in [18] are used, SR indicates that SR-cuts are used, while C+SR indicates that both families of cuts are used.

edge sets as more expensive spanning trees would otherwise have an more expensive edge set cost making them very unattractive compared to the less expensive ones. Such a situation would be unlikely in a free market.

For the Solomon instances RC201 to RC204, test cases were generated with 15, 20, 30 and 40 customers.

8 Results

The program has been implemented in C++ using the COIN bcp library and CLP as the linear programming solver. The tests have been run on a Linux machine with a 64 bit Intel Xeon 2.67 GHz CPU. The CPLEX used for the CPLEX test runs is version 12.1 with all default cuts turned on. The edge set constraints have been implemented in the framework for vehicle routing problems with time windows by Jepsen et al. [12] provided to us by the authors. On the RC201 - RC204 tests with 20 customers we have tested the effect of running branch-cut-and-price with the cuts implemented in [18] only, the SR cuts [12] only, both the cuts from [18] and SR cuts, and without any cuts. In Table 1 the solution times for the four algorithms and for CPLEX are shown. In about half of the instances, CPLEX is not able to solve the routing problem within the time limit of 7500 seconds. For all four branch-and-price and branch-cut-and-price algorithms the solution was found within 500 seconds. We have ranked the results of the four branch-cut-and-price algorithms by solution times. For the objective value of the optimal solution is given in the column "objective" and the number of edge sets payed access to in the optimal solution is given in the column "sets".

In Table 1 the rank of a solution is stated in parenthesis after the solution time. The average of the ranks is calculated for each solution algorithm and shown in the last line of Table 1.

It is seen that CPLEX is the fastest for the three RC201 instances, while the branch-cut-and-price algorithm using both VRPTW cuts implemented in [18] and SR cuts is the fastest for seven instances. The plain branch-and-price algorithm is fastest for two instances while the remaining branch-cut-and-price algorithms using only one of the two described cut families are fastest on none of the instances. The ranking average clearly shows that the branch-cut-and-price algorithm using both the cuts from [18] and SR cuts has the best average rank. Therefore the branch-cut-and-price algorithm used for the tests in Tables 2, 3 and 4 includes the cuts implemented in [18] and SR cuts.

In Tables 2, 3 and 4 the optimal solution of the ESVRPTW problem is presented together with the number of edge sets payed for access to and the amount payed for edge set access in the optimal solution. Moreover a the cost having to pay if the problem was solved not considering the edge set and solved as a VRPTW. Each instance is solved using CPLEX and branch-cut-and-price using the cuts from[18] and the SR cuts and in Tables 2, 3 and 4 the lower bound at the root node and the number of branching nodes processed are reported along with the solution time.

Tables 2 and 3 consider test instances with **random edge sets**. In Table 2 and Table 3 it is seen that the branch-cut-and-price with both the cuts from [18] and SR cuts often has a significantly reduced running

instance		opt sol with set cost				opt sol without set cost				CPLEX			Bcp SR+L		
test	$\|R\|$	total cost	R^*	route cost	set cost	total cost	R^*	route cost	set cost	root LB	nodes	time (s)	root LB	nodes	time (s)
rc201	3	7100	3	4904	2196	7100	3	4904	2196	6381.68	1874	*1.08	6483.00	7	9.30
rc201	5	7173	2	5669	1504	8552	5	4904	3648	7027.00	1163	*0.14	6744.00	11	13.75
rc201	8	7685	3	5727	1958	9908	7	4904	5004	7549.67	102	*0.41	7254.00	19	28.31
rc202	3	5660	1	4891	769	6384	3	4188	2196	3804.68	-	-	5396.00	7	*52.90
rc202	5	5886	2	4382	1504	7836	5	4188	3648	4061.77	-	-	5599.00	11	*206.10
rc202	8	5915	2	4434	1481	8471	6	4188	4283	4561.27	10003	2897.36	5728.00	9	*128.22
rc203	3	5144	1	4375	769	6270	3	4074	2196	2695.67	-	-	5066.00	5	*104.29
rc203	5	5429	1	4643	786	7722	5	4074	3648	3024.38	-	-	5133.00	9	*509.70
rc203	8	5821	2	4340	1481	9874	8	4074	5800	3591.21	-	-	5276.00	25	*870.11
rc204	3	4847	1	4108	739	6017	3	3821	2196	2604.24	-	-	4650.00	5	*262.21
rc204	5	-	-	-	-	-	-	-	-	-	-	-	-	-	-
rc204	8	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 2: RC201-RC204 instances with 30 customers, **random edge sets**. $|R|$ is the number of edge sets, while R^* is the number of selected edge sets in an optimal solution. If the algorithm has not terminated within 7500 seconds it is indicated with "-". The best running time is marked with a "*".

instance		opt sol with set cost				opt sol without set cost				CPLEX			Bcp SR+L		
test	$\|R\|$	total cost	R^*	route cost	set cost	total cost	R^*	route cost	set cost	root LB	nodes	time (s)	root LB	nodes	time (s)
rc201	3	8660	2	6590	2070	9256	3	6116	3140	8591.99	58	*0.47	8283.00	3	7.73
rc201	5	9431	3	6321	3110	11340	5	6114	5224	9284.68	466	*1.14	8930.00	11	62.16
rc201	8	10064	3	7196	2868	13308	7	6116	7192	9543.44	1504	*4.85	9454.00	23	152.61
rc202	3	7805	1	6769	1036	8917	3	5777	3140	5184.79	-	-	7504	9	*226.13
rc202	5	8518	2	6440	2078	11001	5	5777	5224	5562.14	-	-	7973.00	13	*737.95
rc202	8	8755	2	6771	1984	14187	8	5777	8410	5793.10	-	-	8257.00	17	*4316.18
rc203	3	7098	1	6062	1036	8375	3	5235	3140	4057.88	-	-	6709.00	5	*430.43
rc203	5	7120	1	6162	958	10459	5	5235	5224	4259.85	-	-	6949.00	5	*1647.74
rc203	8	-	-	-	-	-	-	-	-	-	-	-	-	-	-
rc204	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
rc204	5	-	-	-	-	-	-	-	-	-	-	-	-	-	-
rc204	8	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 3: RC201-RC204 instances with 40 customers, **random edge sets**. $|R|$ is the number of edge sets, while R^* is the number of selected edge sets in an optimal solution. If the algorithm has not terminated within 7500 seconds it is indicated with "-". The best running time is marked with a "*".

time. However, for all of the instances based on RC201 CPLEX finds the solutions within seconds and always much faster than the branch-cut-and-price algorithm. Moreover note that the lower bound at the root node is often better using CPLEX on the RC201 instances. This may be due to the default CPLEX cuts such as cover, clique, Gomory fractional, multi commodity flow cuts which can not be used when using branch-cut-and-price.

In Table 2 there are two instances with 30 customers which cannot be solved within 7500 seconds using the branch-cut-and-price algorithm. However, more than half of the instances cannot be solved by CPLEX within the time limit of 7500 seconds. For the RC201 to RC204 instances with 40 customers shown in Table 3 only the RC201 instances were solved by CPLEX within the time limit, and 9 instances were not solved by the branch-cut-and-price algorithm within the time limit. Notice, that most of the instances not solved by CPLEX within the time limit were solved by branch-cut-and-price. More than half of the instances were solved by branch-cut-and-price in less than 300 seconds (5 minutes).

Table 2 and Table 3 show the running time for instances generated from RC201 to RC204 with respectively 30 customers and 40 customers. The running times in Table 2 show that the branch-cut-and-price for most of the instances runs much faster than CPLEX. The same is true for the running times in Table 3 when only considering instances where at least one of the algorithms terminated within the time limit.

Table 4 contains the test results for instances RC201 to RC204 with **spanning tree edge sets**. For the RC201 instances, CPLEX solves the instances within a second and considerably faster than the branch-cut-and-price algorithm. For the RC201 to RC204 instances the branch-cut-and-price algorithm solves the problem faster than CPLEX. The last instance has not been solved within the time limit by any of the algorithms. In general the branch-cut-and-price algorithm solves considerably more problems to optimality than CPLEX within the given time limit. This result is in accordance with observations for the VRPTW.

One could also have chosen to solve the problem by brute-force enumerating all subset of the edge sets R,

instance			opt sol with set cost				opt sol without set cost				CPLEX			Bcp SR+L		
test	$\|C\|$	$\|R\|$	total cost	R^*	route cost	set cost	total cost	R^*	route cost	set cost	root LB	nodes	time (s)	root LB	nodes	time (s)
rc201	15	3	2618	1	2386	232	2865	3	2213	652	2504.00	42	*0.05	2601.00	7	0.75
rc201	15	5	3153	3	2386	767	3522	5	2213	1309	3039.00	54	*0.04	3074.00	13	1.98
rc201	15	8	3474	4	2386	1088	4056	7	2213	1843	3474.00	0	*0.06	3406.00	13	1.92
rc202	15	3	2478	2	2061	417	2647	3	1995	652	1629.34	148540	176.17	2375.00	7	*5.03
rc202	15	5	2773	2	2480	293	3112	4	1995	1117	1980.84	37984	33.74	2675.00	9	*9.13
rc202	15	8	3154	4	2294	860	3312	4	1995	1317	2199.40	35999	93.06	2907.00	25	*40.43
rc203	15	3	2400	0	2400	0	2647	3	1995	652	1258.05	944763	556.14	2324.00	9	*5.37
rc203	15	5	2665	1	2369	296	3112	4	1995	1117	1474.43	131586	268.44	2519.00	17	*16.88
rc203	15	8	2982	2	2386	606	3312	4	1995	1317	1742.39	59345	4342.91	2769.00	27	*151.65
rc204	15	3	2240	0	2240	0	2583	3	1931	652	1076.64	-	-	2232.00	5	*6.21
rc204	15	5	2526	1	2230	296	3048	4	1931	1117	1411.57	749482	6698.49	2444.00	5	*36.80
rc204	15	8	2859	2	2253	606	3508	5	1931	1577	1694.75	374593	2438.21	2628.00	27	*402.83
rc201	20	3	3733	2	3446	284	3889	2	3291	598	3698.87	11	*0.02	3733.00	0	0.26
rc201	20	5	4302	3	3449	853	4470	4	3291	1179	4267.16	2	*0.03	4233.00	7	1.34
rc201	20	8	4931	3	3952	979	5094	7	3291	1803	4865.74	9	*0.05	4858.00	5	2.42
rc202	20	3	3464	1	3180	284	3970	3	3069	901	2582.38	38716	107.58	3439.00	5	*3.88
rc202	20	5	3862	2	3255	607	4563	5	3069	1494	3069.67	22019	56.45	3862.00	0	*1.25
rc202	20	8	4635	4	3361	1274	5402	7	3069	2333	3575.66	79652	1276.84	4402.00	13	*163.60
rc203	20	3	3042	0	3042	0	3614	3	2713	901	1429.29	-	-	3015.00	3	*7.09
rc203	20	5	3366	1	3055	311	4207	5	2713	1494	1873.09	-	-	3237.00	15	*136.39
rc203	20	8	4120	3	3122	998	5341	8	2713	2528	2562.29	-	-	3749.00	43	*2986.15
rc204	20	3	2845	0	2845	0	3475	3	2574	901	1292.22	-	-	2845.00	0	*6.82
rc204	20	5	3301	1	2990	311	4068	5	2574	1494	1778.30	-	-	3137.00	11	*1160.77
rc204	20	8	3790	2	3149	641	5102	8	2528	2574	2378.05	-	-	3594.00	29	*6469.83
rc201	30	3	5599	1	5183	416	6098	3	4904	1194	5553.11	22	*0.11	5455.00	5	4.44
rc201	30	5	6204	2	5332	872	6955	5	4904	2051	5995.16	893	*0.34	5850.00	13	12.57
rc201	30	8	6998	4	5332	1666	7809	7	4904	2905	6832.00	813	*0.24	6492.00	19	37.11
rc202	30	3	4832	1	4440	392	5382	3	4188	1194	2812.14	-	-	4715.00	7	*21.95
rc202	30	5	5478	2	4624	858	6239	5	4188	2051	3412.50	-	-	5113.00	19	*99.57
rc202	30	8	6431	5	4779	1652	7386	8	4188	3198	4212.98	-	-	5756.00	71	*1482.03
rc203	30	3	4418	0	4418	0	5268	3	4074	1194	1997.00	-	-	4418.00	0	*14.01
rc203	30	5	5191	2	4418	773	6125	5	4074	2051	2571.25	-	-	4935.00	17	*374.77
rc203	30	8	5877	3	4574	1303	7308	8	4074	3234	3228.88	-	-	5383.00	39	*4458.23
rc204	30	3	4292	0	4292	0	5015	3	3821	1194	1923.95	-	-	4292.00	0	*24.82
rc204	30	5	4953	2	4095	858	5872	5	3821	2051	2461.95	-	-	4649.00	25	*5606.51
rc204	30	8	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 4: RC201-RC204 instances, **spanning edge tree edge sets**. $|R|$ is the number of edge sets, while R^* is the number of selected edge sets in an optimal solution. If the algorithm has not terminated within 7500 seconds it is indicated with "-". The best running time is marked with a "*".

and for each subset to solve the corresponding VRPTW with the branch-cut-and-price algorithm. There are $2^{|R|}$ combinations to be considered, and each time the VRPTW algorithm must be run. For $|R| = 3$ the VRPTW algorithm must be run 8 times, while for $|R| = 10$ it must be run 1024 times. To get an estimate of the running times, using this approach, we have used the VRPTW algorithm to solve one of the $2^{|R|}$ combinations and then estimated how long time it would take to run it $2^{|R|}$ times. Since some combinations of groups may lead to an infeasible problem we consider the combination where all sets are selected.

For the instance RC201 with 30 customers it took 0.5 seconds to solve the problem for one combination of the edge sets. A rough estimate indicates that it will take around 4 seconds for $|R| = 3$, and around 128 seconds for $|R| = 8$. This indicates that for a small number of subsets it is faster to solve the a VRPTW for each combination of edge sets, while for a large number of subsets it is better to solve the ESVRPTW.

We also tested the instance RC203 with 30 customers. In this case it took 3.6 seconds to solve one combination of edge sets. A rough estimate indicates that it would take around 29 seconds to solve the problem for $|R| = 3$, and around 922 seconds to solve the problem for $|R| = 8$.

We expect that the more groups are present in the problem the more advantageous it is to solve the ESVRPTW. The implemented branch-cut-and-price branches on the group variables first and thereby can eliminate combinations of groups using upper bounds provided by other combinations. This synergy is not achieved when running the combinations individually.

9 Conclusion

The vehicle routing problem with time windows and fixed costs for accessing an edge set (ESVRPTW) has been presented in this paper. To the best of our knowledge, it is the first time this type of problem has been

investigated. A mathematical model has been presented for the ESVRPTW. We have applied the branch-cut-and-price method to the problem and shown that including the SR cuts and the cuts implemented in Lysgaard [18] for the VRPTW and CVRP improves the solution times for this problem. Many related routing problems may with advantage be implemented this way using the extensive research available for the CVRP and the VRPTW. These problems are often solved using heuristic methods. Although the heuristic solution methods are very useful and relevant in real life applications it is also important for the evaluation of the heuristics to have access to some optimal solutions.

On a more general level the paper has opened the door to considering VRPTW problems completely differently than in previous literature. Previous papers on VRPTW assume that the shortest path between two destinations is well-defined and unique, making it possible to abstract from the underlying graph. However, this assumption seldom holds in practice, and the here presented model is only one example of such models. One could also imagine problems where multiple edges (roads) are present between each pair of nodes, having different cost and time. For instance, driving along a freeway imposes a higher cost but shorter travel time in comparison to using a highway. In our future work, we will look deeper into these variants of VRPTW.

Acknowledgments

The authors wish to thank Richard Martin Lusby for valuable comments.

References

[1] R. Baldacci, A. Mingozzi, and R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59:1269–1283, 2011.

[2] J.-M. Belenguer, E. Benavent, C. Prins, C. Prdhon, and R. W. Calvo. A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Reserach*, 38:931–941, 2011.

[3] L. Blander Reinhardt and D. Pisinger. Multi-objective and multi-constraint non-additive shortest path problems. *Computers and Operations Research*, 38:605–616, 2011.

[4] N. Christofides, A. Mingozzi, and P. Toth. State-space relaxation procedures for the computation of bounds to routing problems. *Networks*, 11:145–164, 1981.

[5] W. Cook and J. L. Rich. A parallel cutting-plane algorithm for the vehicle routing problem with time windows. Technical report, Rice University, 1999.

[6] G. B. Dantzig and J. R. Ramser. The truck dispatching problem. *Management Science*, 6:80–91, 1959.

[7] G. Desaulniers, J. Desrosriers, J. Ioachim, I. M. Solomon, F. Soumis, and D. Villeneuve. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In T. G. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 57–93. Kulwer, Nowell, MA, 1998.

[8] G. Desaulniers, F. Lessard, and A. Hadjar. Tabu search, partial elementarity, and generalized *k*-path inequalities for the vehicle reouting problem with time windows. *Transportation Science*, 42:387–404, 2008.

[9] M. Desrochers, J. Desrosiers, and M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40:342–354, 1992.

[10] M. Dror. Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research*, 42:977–978, 1994.

[11] R. Fukasawa, H. Longo, J. Lysgaard, M. Poggi de Aragao, M. Reis, E. Uchoa, and R. F. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Math. Programming*, 106:491–511, 2006.

[12] M. Jepsen, B. Petersen, S. Spoorendonk, and D. Pisinger. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56:497–511, 2008.

[13] B. Kallehauge. Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers & Operations Research*, 35:2307–2330, 2008.

[14] B. Kallehauge, J. Larsen, and O. B. G. Madsen. Lagrangian duality applied to the vehicle routing problem with time windows. *Computers & Operations Research*, 33:1464–1487, 2006.

[15] N. Kohl, J. Desrosiers, O. B. G. Madsen, M. M. Solomon, and F. Soumis. 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, 33:101–116, 1999.

[16] A. W. J. Kolen, A. H. G. Rinnooy Kan, and H. W. J. M. Trienekens. Vehicle routing with time windows. *Operations Research*, 35:266–273, 1987.

[17] A. N. Letchford, R. W. Eglese, and J. Lysgaard. Multistarts, partial multistars and the capacitated vehicle routing problem. *Math. Programming*, 94:21–40, 2002.

[18] J. Lysgaard. CVRPSEP: a package of separation routines for the capacitated vehicle routing problem, 2003.

[19] J. Lysgaard, A. N. Letchford, and R. W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming Serie A*, 100:423–445, 2004.

[20] G. Nagy and S. Salhi. Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177:649–672, 2007.

[21] G. Righini and M. Salani. Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, 3:255–273, 2006.

[22] M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35:254–265, 1987.