Technical University of Denmark

The vehicle routing problem with edge set costs

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

Publication date:
2011

Document Version
Publisher's PDF, also known as Version of record

The vehicle routing problem with edge set costs

Report 8.2011

DTU Management Engineering

Line Blander Reinhardt
Mads Kehlet Jepsen,
David Pisinger
April 2011

DTU Management Engineering
Department of Management Engineering

The vehicle routing problem with edge set costs

Line Blander Reinhardt, Mads Kehlet Jepsen, David Pisinger
Department of Management Engineering, Technical University of Denmark

April 26, 2011

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 contributions impose a cost for the company while they also give unlimited usage of a set of roads to all vehicles belonging to the company. Different versions for defining the edge sets are discussed and formulated. A MIP-formulation of the problem is presented, and a solution method based on branch-and-price-and-cut is applied to the problem. The computational results show that instances with up to 50 customers can be solved in reasonable time, and that the branch-cut-and-price algorithm generally outperforms CPLEX. It also seems that instances get more difficult when the penalized edge sets form a spanning tree, compared to when they are randomly scattered.

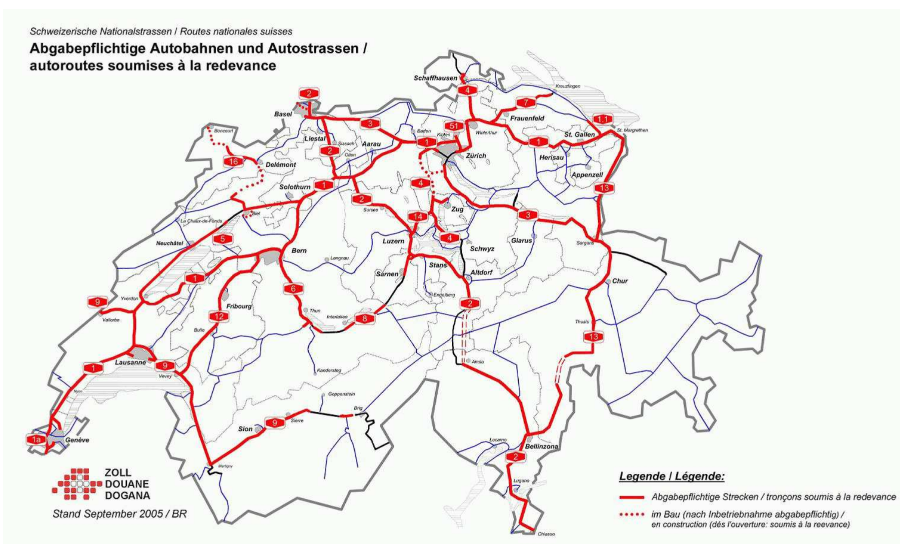
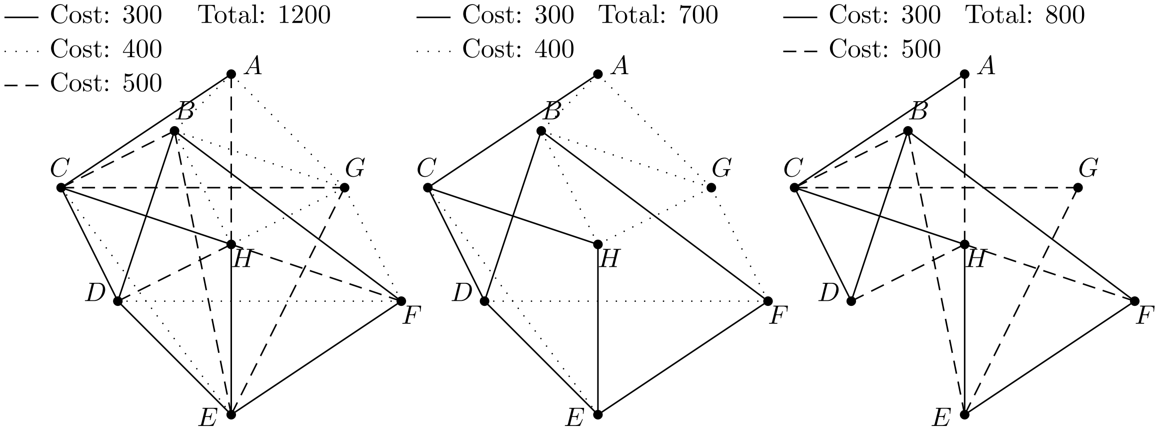


Figure 1: Main road net in Switzerland. To access all the red edges, a vignette needs to be paid. A transportation company may choose to avoid the toll roads and only use the ordinary highways

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



(a) All 3 edge sets of the graph. (b) Accessible edges when the two edge sets shown are paid for. (c) Accessible edges when the two edge sets shown are paid for.

Figure 2: The graph of a) all three edge sets b) two edge sets and c) another two edge sets

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 [19] 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. In all of the mentioned cases there is a fixed charge for accessing a set of edges. Apart from considering multiple depots, the problem in [2] can be seen as a special case of the model presented in this paper.

The problem of minimizing the overall cost when planning routes that 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 the version of the VRPTW considered here the edges of the graph belong to different sets. Once the cost of accessing the set is paid all vehicles can access the edges in the set. However, there might still be a price associated with each of the edges used. Note that the price for accessing the 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 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 set. Clearly the ESVRPTW is NP-hard as it is a generalization of the VRPTW problem. We will in this paper present a model for the problem and a Danzig-Wolfe decomposition similar to the classical decomposition of the VRPTW.

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 MIP model for the ESVRPTW and in Section 4 the decomposition of the problem into a Master and subproblem is described. Moreover the solution method and valid inequalities are described. In Section 5 various extensions of the ESVRPTW model are discussed. In Section 6 the test instances are described. Section 7 reports computational results, and finally the paper is concluded in Section 8.

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 Danzig 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. [15] 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 [21] 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. [14] 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],[11], and [13]. Both Jepsen et al. [11] and Baldacci et al. [1] use the valid cuts suggested by Lysgaard et al. [18] to separate candidate sets for branching. Even though the cuts in [18] are implemented for the capacitated vehicle routing problem (CVRP) they may be used for the VRPTW, as the solutions to the VRPTW problem are a subset of the solution to the corresponding CVRP. Jepsen et al. [11] implemented a branch cut and price algorithm with a label-setting bi-directional algorithm for elementary shortest paths developed by Righini and Salani [20]. 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. [14]. Baldacci et al. [1] relaxed the subproblem so that the routes are the so called ng-routes which are not necessarily simple. To find the ng-routes Baldacci et al. [1] developed a dynamic programming algorithm. The ng-routes are used for bounding, and branching is only done whenever enumeration is not possible.

In this paper we formulate the ESVRPTW and solve it using the solution method used by Jepsen et al. [11] on the VRPTW as this method can be easily adapted to solve the ESVRPTW.

3 The Model

The mathematical model is based on the model presented in [11]. Given the following sets:

C	The set of customers
R	The set of edge groups
V	The set vertices representing the customers in C and the depot defined as 0
$\mathbf{A}$	The set of arcs (i,j) in V and $\mathbf{A}_r$ is the set of arcs (i,j) belonging to the group $r \in R$
K	The set of vehicles

The variables are defined as:

x_{ij}^v	Indicator variable indicating if the arc (i, j) is used by vehicle $v \in K$
y_r	Indicator variable which is one if an edge from group $r \in R$ 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 vertex $i \in V$. The demand at the depot is zero
a_i	The availability time for customer $i \in C$
b_i	The required completion time for customer $i \in C$
c_{ij}	The cost of using an arc $(i, j) \in A$
c_r	The cost of accessing the arcs in group $r \in R$

Since the problem is a generalization of the VRPTW the model presented here for the ESVRPTW is the standard VRPTW model presented by Kallehauge in the survey [12], 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 set, however, alternatives to this assumption are discussed in Section 5.

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

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

The objective (1) is the sum of the cost on the edges accessed and the sum of the cost of accessing the sets of the edges accessed. The constraints (2) ensure that if an edge in a set is used then the cost of accessing the set is paid. Note that the integrality of the x_{ij}^v 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) ensure that the vehicles travel a connected path. 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 integer.

3.1 Tightening of the edge set constraints

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

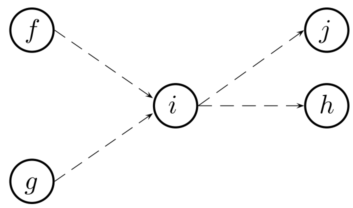


Figure 3: Bound on outgoing edges

From this observation we can construct the constraints:

$$\sum_{v \in K} \sum_{(i,j) \in \mathbf{A_r}} x_{ij}^v \leq y_r \qquad \forall i \in C, \forall r \in R \tag{12}$$

In this case the integrality of the x variables again imposes the integrality of the y variables and the number of constraints in (12) is $|C|$. Note that constraints (12) do not apply to the depot as more than one edge belonging to a group may leave the depot. Therefore the constraints of type (12) cannot entirely replace the constraints (2). However, by formulating a new set of constraints (see (13)) similar to the constraints (12) for edges entering every costumer then the constraints (2) can be replaced by $2|C|$ constraints.

This means that the constraints (2) in the model can be replaced by the constraints:

$$\sum_{v \in K} \sum_{(j,i) \in \mathbf{A_r}} x_{ji}^v \leq y_r \qquad \forall i \in C, \forall r \in R \tag{13}$$

$$\sum_{v \in K} \sum_{(i,j) \in \mathbf{A_r}} x_{ij}^v \leq y_r \qquad \forall i \in C, \forall r \in R \tag{14}$$

These tighter constraints will in the following replace constraints (2) in the model.

4 Solution method

The branch-cut-and-price method has with success been applied to the VRPTW problem and the best results for finding exact solutions to the problem have been produced using this method (see Jepsen et al. [11], Desaulniers et al. [8] and Baldacci et al. [1]). Since the problem ESVRPTW is a generalization of the VRPTW the solution methods for the VRPTW may successfully be applied to the ESVRPTW. Therefore we will apply the BCP algorithm to the VRPTW using cuts for the original formulation of the VRPTW presented by Fukasawa et al. in [10] and by Lysgaard et al. [18] for the CVRP. This corresponds to the algorithm developed by Jepsen et al. [11] for the VRPTW problem. 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 ESVRPTW.

The ESVRPTW can be decomposed into a master and pricing problem using the standard VRPTW Dantzig-Wolfe decomposition where, the pricing problem is to find a elementary shortest path problem with resource constraints.

4.1 Master Problem

The master problem is similar to the standard VRPTW decomposition master problem 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.

$$\textbf{Min:} \sum_{p \in P} \sum_{(i,j) \in \mathbf{A}} c_{ij} \alpha_{ijp} \lambda_p + \sum_{r \in R} c_r y_r \tag{15}$$

$$s.t. \quad \sum_{p \in P} \sum_{(j,i) \in \mathbf{A_r}} \alpha_{jip} \lambda_p \leq y_r \qquad \forall i \in C, \forall r \in R \tag{16}$$

$$\sum_{p \in P} \sum_{(i,j) \in \mathbf{A_r}} \alpha_{ijp} \lambda_p \leq y_r \qquad \forall i \in C, \forall r \in R \tag{17}$$

$$\sum_{p \in P} \sum_{(j,i) \in \mathbf{A}} \alpha_{jip} \lambda_p = 1 \qquad \forall i \in C \tag{18}$$

$$\lambda_p \in \{0, 1\} \qquad \forall\, p \in \mathbf{P} \tag{19}$$

$$y_r \in \{0, 1\} \qquad \forall\ r \in R \tag{20}$$

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 $(i, j) \in A$ is used by the route p and zero otherwise. Constraints (16) and (17) corresponds to the constraints (13) and (14) which ensure that access to the edges used is paid once if an edge is used in one of the selected routes. Constraints (18) 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 (16) and (17) do not change the domain of valid solutions but only affect the value of the solutions.

4.2 Sub problem

The linear relaxation of the master problem can be solved through delayed column generation. The pricing problem is the elementary shortest path problem with resource constraints. Let $\phi'_{ir} \in \mathbb{R}$ be the dual variables of constraints (16) and let $\phi_{ir} \in \mathbb{R}$ be the dual variables of constraints (17). Let $\pi_j \in \mathbb{R}$ be the dual variables of constraint (18) and let $\pi_0 = 0$, $\phi'_{0r} = 0$ and $\phi_{0r} = 0$. Then, the reduced cost for a route in the pricing problem becomes:

$$\bar{c}_p = \sum_{(i,j) \in \mathbf{A}} c_{ij} \alpha_{ijp} - \sum_{(i,j) \in \mathbf{A}} \pi_j \alpha_{ijp} - \sum_{r \in R} \sum_{(j,i) \in A_r} \phi'_{ir} \alpha_{jip} - \sum_{r \in R} \sum_{(i,j) \in A_r} \phi_{ir} \alpha_{ijp} \tag{21}$$

$$= \sum_{(i,j) \in \mathbf{A}} (c_{ij} - \pi_j) \alpha_{ijp} - \sum_{r \in R} \sum_{(i,j) \in A_r} (\phi_{ir} + \phi'_{jr}) \alpha_{ijp} \tag{22}$$

This can be transformed to the elementary shortest path problem with resource constraints (ESPPRC) where each edge (i, j) has the cost $\bar{c}_{ij} = c_{ij} - \pi_j - \sum_{\{r | (i,j) \in A_r\}} (\phi_{ir} + \phi'_{jr})$. The resource constraints included in the elementary shortest path problem are the demand picked up along the route and the time accumulated along the route. The demand of the customers visited by the route must be less than the capacity and the customers must be visited within their time window. The ESPPRC pricing problem can be solved by a label-setting bidirectional shortest path algorithm developed by Righini and Salani [20]. The domination criteria 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 strictly de-/in-creasing, can be used here. However, when introducing the SR cuts which will be described later the objective is no longer additive and the function used is not strictly de-/in-creasing. In [3] Blander Reinhardt and Pisinger cover several different ESPPRC problems with objectives containing functions which are not strictly de-/in-creasing.

4.3 Cuts

After adding route variables to the master problem it is investigated if cuts can be added to the master problem. If the added cuts are valid inequalities derived from the original formulation (2) to (10) 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 in the form of the paths 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. [11]. Jepsen et al. [11] developed a method of handling the reduced cost of a route for the ESPPRC where the objective contains a function which is not strictly in- or decreasing as a result of the reduced cost achieved from the Subset Row cuts.

4.3.1 Valid Inequalities in the original form

Many valid inequalities have been developed for the VRPTW problem. These valid inequalities will also be applicable for ESVRPTW problem as the ESVRPTW problem does not change the set of feasible solutions but only changes the objective function. Valid inequalities for the VRPTW are described in [10], [16] and [18]. 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 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 problem they do not include the time window restrictions to possibly tighten inequalities. The separation algorithm used is that described by Lysgaard et al. [18] and accessible in the framework developed by Lysgaard [17].

4.3.2 Valid Inequalities in the master problem form

In [11], Jepsen et al. developed the Subset-Row (SR) inequalities to generate cuts in the set partitioning formulation of the master problem. The SR inequalities are inspired by the clique and odd hole inequalities for the set-packing problem.

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

$$\text{(Subset-Row:)} \quad \sum_{p \in P} \left\lfloor \frac{1}{k} \sum_{i \in S} \alpha_{ip} \right\rfloor \lambda_p \leq \left\lfloor \frac{|S|}{k} \right\rfloor \tag{23}$$

Where S is a subset of the constraints (18) and $0 < k \leq |S|$ and

$$\left\lfloor \frac{1}{k} \sum_{i \in S} \alpha_{ip} \right\rfloor = \left\lfloor \frac{1}{k} \sum_{i \in S} \sum_{(i,j) \in \delta^+(i)} \alpha_{ijp} \right\rfloor$$

Clearly, if λ_p has a binary value satisfying the customer constraint (18) then the SR inequalities (23) will also be satisfied. However, when solving the linearly relaxed master problem λ_p are relaxed to linear variables between zero and one then there might be solutions where the inequalities (23) are not satisfied. These inequalities are limited to the set of (18) constraints and can therefore easily be introduced in the ESVRPTW problem. Moreover the effect from introducing the SR cuts into the ESVRPTW should be the same as in the VRPTW as the set of feasible solutions do 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. [11] we let the dual variable of a SR inequality be σ we can then formulate the dual cost of a route p as

$\hat{c}_p = \bar{c}_p + \sigma \left\lfloor \left(\sum_{i \in S} \sum_{(i,j) \in \delta^+(i)} \alpha_{ijp} \right) / k \right\rfloor$. Note that the reduced cost σ is not activated before at least k vertices in the set S has 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 4.2, the ESPPRC is solved with a label-setting algorithm using domination 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 vertex 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. We work with the cuts not allowing two or more routes to visit two vertices from a set of three customers, $k = 2$ and $|S| = 3$.

To solve this problem Jepsen et al. [11] modified the domination rule for the cost criteria in the ESPPRC. The modification consists of subtracting the reduced cost σ_q from a label L_i. Where the cost of cut q is included in label i and not included in dominated label L_j so that the domination rule for the cost of two labels at the same vertex becomes:

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

Where Q is the set of SR cuts with sets of three customers where L_i has visited two vertices in the set S and L_j has not and $\sigma_q < 0$. The domination rules for the remaining constraints are kept the same. For further details see Jepsen et al. [11].

4.4 Branch-and-cut-and-price

The branch-cut-and-price algorithm is commonly used for solving integer problems to optimality. Below we describe the algorithm with some of the conditions selected in the method used here included.

The branch-cut-and-price algorithm:

Step 1: Choose an unprocessed node. The node with the lowest lower bound. 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.

Step 3: search for routes with negative reduced cost using heuristic methods. If any found add up to 400 to the master problem and go to step 2. The heuristic used is a simplified version of the ESPPRC algorithm.

Step 4: Solve the pricing problem 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 update the lower bound if the lower bound above the upper bound then remove the node from the unprocessed node list and goto 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 integer 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.

The branching used is described in more detail in the next subsection.

4.5 Branching

For VRPTW the branching is most commonly done on edge variables. We have chosen to do branching on the group variables as well. Branching on the group variables can reduce the depth of the search tree as the edges to branch on are restricted. Moreover the number of group variables is comparably small. The node with the lowest lower bound is chosen for branching. The branching candidates are selected as the groups first and otherwise the edges where they are separated by using the branching strategy presented by Fukasawa et al. [10] where branching occurs on a set of customers $S \subset C$ by having one branch with one vehicle covering the set S. This is represented by constraint $\sum_{v\in K}\sum_{(i,j)\in\delta^+(S)}(x_{ij}^v + x_{ji}^v) = 2$ where $\delta^+(S)$ is the edges leaving the set S. The other branch has at least two vehicles covering the customers in the set S represented by constraint $\sum_{v\in K}\sum_{(i,j)\in\delta^+(S)}(x_{ij}^v + x_{ji}^v) \geq 4$. To separate candidate sets the Lysgaard cut library [17] is used. From preliminary tests it was clear that branching on the group variables tended to improve the solution time for the problem significantly.

5 Closely related formulation and problems

Clearly, there are different versions of the problem where the edges are part of a set. It has been assumed that the edges only belong to one set, however there could be cases where the edges belong to more than one set. If an edge can belong to one set there can be different ways to add the charge. The simple choice would be that the cost must be paid for all the sets an edge belongs to. This problem can be formulated with the constraints (13) and (14) with the only difference that an edge may be included in more than one constraint for each node. Other alternatives are discussed in this section.

5.1 Edges belonging to multiple sets

Another variant could be that for an edge belonging to several sets, the access cost only needs to be paid for one of the sets to which the edge belongs. This can be formulated as:

$$x_{ij} - \sum_{(i,j)\in r\in R} y_r \leq 0 \;\; \forall (i,j) \in \mathbf{A} \tag{24}$$

This constraint is very similar to constraints (2); however, in this case the integrality of the x_{ij} variables does not necessarily imply integrality of the y_r variables. Note, that when replacing constraints (16) and (17) with (24) each edge (i,j) in the ESPPRC sub problem will have cost $c_{ij} - \pi_j - \theta_{ij}$ where θ is the dual variable for the constraints (24). 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.

5.2 Accessing a set of reduced prices

In some cases one may access edges belonging to a set without paying for accessing the set but by paying a more expensive price for using each edge. For instance, instead of buying company access to all freeways in a country, one may be allowed to pay with cash at the barrier to each road. These cash prices are expensive but may be attractive if there is a very limited usage of the edges in the set. This extension is easily handled in our model by duplicating each edge, where one edge belongs to an edge set, and the other edge correspond to cash payment.

$$x_{ij}^{z_r} - \sum_{r\in(i,j)} y_r \leq 0 \;\; \forall (i,j) \in \mathbf{A} \tag{25}$$

where $x_{ij}^{z_r}$ is the edge between i and j which becomes accessible when paying the price for accessing the set r.

In this case each edge $x_{ij}^{z_r}$ in the ESPPRC will have the cost $c_{ij}^{z_r} - \pi_j - \zeta_{ij}$ where ζ_{ij} is the dual variable of the respective constraint of type (25) and the cost of the edges not in sets will simply have the cost $c_{ij} - \pi_j$.

6 Test data

Following the tradition in VRP problems, the test data are based on the Solomon instances [21] making it possible to relate our results to the existing literature. We have generated test instances based on the RC201 to RC204 and C101 to C109 instances by assigning subsets of edges to disjoint groups, and associating a fixed cost with each group. 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 sets**: In these instances, the edges in each 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 sets**: In these instances the selected edges form cheap spanning graphs of a randomly selected subset of vertices. Each subset of vertices consists of half of the total number of vertices. 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.

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

For the **random edge sets** instances, 50% of the edges are assigned to groups with an additional cost. For each combination of Solomon instance and number of groups, two test instances were generated: one case with the costs of an edge set group calculated as $\beta = 5\%$ of the average cost of the edges in the group multiplied by the number of vertices in the set, and another case using the same calculations with $\beta = 10\%$.

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

For the Solomon instances RC201 to RC204, test cases were generated with 15, 20, 30 and 40 customers. For the instances C101 to C109, test cases were generated containing 50 customers using **random edge sets**. We only consider test cases up to 40 customers for the RC201 to RC204 instances, since many instances with 40 customers were not solveable within the given time limit. Instead we have run larger instances with 50 customers for the C101 to C109 instances, as these are know to be easier from the VRPTW literature.

7 Results

The program has been implemented in C++ using the COIN bcp library and CLP as the linear programming solver. The test have been run on a Linux machine with a 64 bit Intel Xeon 2.67 GHz CPU. The edge set constraints have been implemented in the framework by Jepsen et al. [11] provided to us by the authors. On the RC202 - RC204 tests with 20 customers we have tested the effect of running branch-cut-and-price with Lysgaard [17] cuts only, the SR cuts [11] only, both the Lysgaard 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. 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. In Table 1 the rank of a solution is stated

instance			opt solution		solution times				
test	groups	cost β	objective	groups	CPLEX	bcp L+SR	Bp	bcp SR	bcp L
rc201	3	05	4239	3	*0.06	1.22 (3)	1.16(1)	1.22(3)	1.20(2)
rc201	3	10	4912	2	*0.06	0.97(1)	0.99(2)	1.01(3)	1.05(4)
rc201	5	05	4539	4	*0.06	1.09(1)	1.23(4)	1.18(3)	1.16(2)
rc201	5	10	5787	4	*0.1	10.43(3)	7.36(1)	10.83(4)	7.86(2)
rc201	8	05	4878	5	*0.09	1.62(2)	1.52(1)	1.66(3)	1.71(4)
rc201	8	10	5847	3	*0.05	4.880(2)	4.030(1)	5.04(4)	4.14(2)
rc202	3	05	3723	2	473.82	*13.55(1)	14.86(2)	16.85(3)	21.76(4)
rc202	3	10	4371	2	200.65	11.22(3)	*10.17(1)	14.49(4)	10.85(2)
rc202	5	05	4120	3	200.19	10.26(2)	*9.40(1)	11.71(4)	11.51(3)
rc202	5	10	4969	2	96.86	32.47(3)	13.83(2)	56.17(4)	*13.44(1)
rc202	8	05	4166	3	25.54	*9.96(1)	10.13(2)	17.06(4)	11.75(3)
rc202	8	10	5115	4	*8.07	37.30(3)	24.54(1)	45.20(4)	25.13(2)
rc203	3	05	3371	1	-	*29.07(1)	29.19(2)	30.30(3)	54.01(4)
rc203	3	10	3694	1	-	*25.08(1)	28.43(3)	29.38(4)	27.85(2)
rc203	5	05	3635	2	-	*44.34(1)	44.66(2)	47.71(3)	58.24(4)
rc203	5	10	3968	1	3571.92	*24.33(1)	27.76(3)	32.11(4)	26.28(2)
rc203	8	05	3545	2	-	*75.54(1)	81.25(2)	88.53(3)	100.63(4)
rc203	8	10	3954	1	6294.51	*42.05(1)	44.43(3)	48.35(4)	43.32(2)
rc204	3	05	3148	1	-	*130.83(1)	147.04(3)	183.59(4)	136.88(2)
rc204	3	10	3471	1	-	452.81(4)	112.62(2)	303.24 (3)	*89.45(1)
rc204	5	05	3414	2	-	*123.00(1)	205.80(3)	153.72(2)	308.24(4)
rc204	5	10	3797	1	-	315.28(3)	121.67(2)	397.13(4)	*106.00(1)
rc204	8	05	3215	1	-	100.40(3)	*43.06(1)	119.16(4)	54.43(2)
rc204	8	10	3507	1	-	*46.61(1)	61.35(4)	54.26(2)	60.78(3)
Average Rank						1.88	2.04	3.46	2,58

Table 1: RC201-RC204 instances with 20 customers, **random sets**. If the algorithm has not terminated within 7500 seconds it is indicated with "-". The best running time for each instance is marked with a "*". L indicates that the cuts implemented in Lysgaard are used, SR indicates that SR-cuts are used, while SR+L indicates that both families of cuts are used.

in paranthesis 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 7 instances, while the branch-cut-and-price algorithm using both Lysgaard cuts and SR cuts is the fastest for 11 instances. The branch-cut-and-price algorithms using only one of the cut families are only fastest for 3 instances. The ranking average clearly shows that the branch-cut-and-price algorithm using both Lysgaard and SR cuts has the best average rank. Therefore the branch-cut-and-price algorithm used for the tests in Tables 2, 3, 4 and 5 includes the Lysgaard and SR cuts.

instance			opt solution		solution times	
test	groups	cost β	objective	groups	CPLEX	Bcp L+SR
rc201	3	05	7100	3	*1.08	9.30
rc201	3	10	8396	1	*0.7	20.78
rc201	5	05	7173	2	*0.14	13.75
rc201	5	10	8512	1	*0.23	21.60
rc201	8	05	7685	4	*0.41	28.31
rc201	8	10	9031	2	*0.64	65.73
rc202	3	05	5660	2	-	*52.90
rc202	3	10	6430	1	283.68	*167.84
rc202	5	05	5886	2	-	*206.10
rc202	5	10	6831	1	919.49	*253.07
rc202	8	05	5915	3	2897.36	*128.22
rc202	8	10	7398	2	*328.52	3407.25
rc203	3	05	5144	1	-	*104.29
rc203	3	10	5914	1	-	*119.07
rc203	5	05	5429	1	-	*509.70
rc203	5	10	6215	1	-	*1533.37
rc203	8	05	5821	2	-	*870.11
rc203	8	10	6494	0	-	*1491.57
rc204	3	05	4847	1	-	*262.21
rc204	3	10	5587	1	-	*545.37
rc204	5	05	-	-	-	-
rc204	5	10	5743	1	-	*1621.71
rc204	8	05	-	-	-	-
rc204	8	10	6118	0	-	*3307.06

Table 2: RC201-RC204 instances with 30 customers, **random sets**. If the algorithm has not terminated within 7500 seconds it is indicated with "-". The best running time is marked with a "*".

instance			opt solution		solution times	
test	groups	cost β	objective	groups	CPLEX	Bcp L+SR
rc201	3	05	8660	2	*0.47	7.73
rc201	3	10	10730	2	*0.95	28.95
rc201	5	05	9431	3	*1.14	62.16
rc201	5	10	11982	2	*6.63	127.22
rc201	8	05	10064	3	*4.85	152.61
rc201	8	10	11922	2	*2.45	348.53
rc202	3	05	7805	1	-	*226.13
rc202	3	10	8841	1	-	*503.73
rc202	5	05	8518	2	-	*737.95
rc202	5	10	10072	1	-	*3671.59
rc202	8	05	8755	2	-	*4316.18
rc202	8	10	-	-	-	-
rc203	3	05	7098	1	-	*430.43
rc203	3	10	-	-	-	-
rc203	5	05	7120	1	-	*1647.74
rc203	5	10	-	-	-	-
rc203	8	05	-	-	-	-
rc203	8	10	-	-	-	-
rc204	3	05	-	-	-	-
rc204	3	10	5838	0	-	*882.34
rc204	5	05	-	-	-	-
rc204	5	10	5838	0	-	*1848.44
rc204	8	05	-	-	-	-
rc204	8	10	-	-	-	-

Table 3: RC201-RC204 instances with 40 customers, **random sets**. If the algorithm has not terminated within 7500 seconds it is indicated with "-". The best running time is marked with a "*".

Tables 2, 3 and 4 consider test instances with **random sets**. In Table 2 and Table 3 it is seen that the branch-cut-and-price with both Lysgaard and SR cuts often has a significantly reduced running 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. 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 the branch and bound algorithm within the time limit, and 9 instances were not solved by the branch-cut-and-price algorithm within the time limit.

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.

Notice, that most of the instances not solved by CPLEX within the time limit were solved by branch-cut-and-price. Half of the instances were solved in less than 600 seconds (10 minutes).

For the C101-C109 instances with 50 customers shown in Table 4 the results are more mixed. CPLEX is the fastest for easy instances, while the branch-cut-and-price algorithm has some computational overhead which only pays off for the difficult instances. The branch-cut-and-price algorithm solves significantly more instances within the time limit, although there are two instances solved by CPLEX which cannot be solved by branch-cut-and-price within the time limit.

Table 5 contains the test results for instances RC201 to RC204 with **spanning tree sets**. For the RC201 instances, CPLEX solves the instances within a second and considerably faster than the branch-cut-and-price algorithm. For the RC202 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.

Instance			opt solution		Solution Times	
test	groups	cost β	objective	groups	CPLEX	Bcp L+SR
c101	3	05	5683	3	*0.99	37.61
c101	3	10	7052	3	*0.66	162.79
c101	5	05	6749	2	*1.04	574.34
c101	5	10	7725	3	*1.08	135.60
c101	8	05	7125	3	*2.27	1037.11
c101	8	10	8138	2	*1.4	570.75
c102	3	05	5304	2	-	*322.95
c102	3	10	-	-	-	-
c102	5	05	5741	2	-	*1210.31
c102	5	10	-	-	-	-
c102	8	05	6201	1	-	*6932.81
c102	8	10	-	-	-	-
c103	3	05	4801	1	-	*2158.50
c103	3	10	5081	0	-	*535.76
c103	5	05	4979	1	-	*5999.15
c103	5	10	5081	0	-	*252.92
c103	8	05	5081	0	-	*854.73
c103	8	10	5081	0	-	*923.91
c104	3	05	-	-	-	-
c104	3	10	-	-	-	-
c104	5	05	-	-	-	-
c104	5	10	-	-	-	-
c104	8	05	-	-	-	-
c104	8	10	-	-	-	-
c105	3	05	5421	2	*36.90	394.63
c105	3	10	6228	0	*121.33	-
c105	5	05	5838	2	*705.29	1686.57
c105	5	10	6228	0	*124.26	1932.97
c105	8	05	6083	1	*852.82	5477.76
c105	8	10	6228	0	*31.51	2906.03
c106	3	05	5600	2	*2.05	147.36
c106	3	10	6870	1	*3.05	458.43
c106	5	05	6428	2	*5.82	2652.120
c106	5	10	7317	1	*2.65	6803.880
c106	8	05	6896	2	*4.53	1340.30
c106	8	10	7378	0	*1.66	1453.82
c107	3	05	5204	2	*364.35	439.94
c107	3	10	6032	0	*1506.13	-
c107	5	05	5618	1	*2250.54	2945.98
c107	5	10	6032	0	*3385.01	7312.24
c107	8	05	5700	1	*220.37	1164.84
c107	8	10	6032	0	5519.13	*1692.73
c108	3	05	5076	1	-	*1260.62
c108	3	10	-	-	-	-
c108	5	05	5325	-	-	*1124.48
c108	5	10	5747	0	-	*3845.29
c108	8	05	5628	1	-	*2362.47
c108	8	10	5747	0	-	*5474.220
c109	3	05	-	-	-	-
c109	3	10	-	-	-	-
c109	5	05	-	-	-	-
c109	5	10	5022	0	-	*428.11
c109	8	05	5022	0	-	*2730.11
c109	8	10	5022	0	-	*804.76

Table 4: C101-C107 instances with 50 customers, **random sets**. If the algorithm has not terminated within 7500 seconds it is indicated with "-". The best running time is marked with a "*".

8 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 [17] 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 VRPTW problems.

instance			opt solution		solution times	
test	customers	groups	objective	groups	CPLEX	Bcp SR+L
rc201	15	3	2618	1	*0.05	0.75
rc201	15	5	3153	3	*0.04	1.98
rc201	15	8	3474	4	*0.06	1.92
rc202	15	3	2478	2	176.17	*5.03
rc202	15	5	2773	2	33.74	*9.13
rc202	15	8	3154	4	93.06	*40.43
rc203	15	3	2400	0	556.14	*5.37
rc203	15	5	2665	1	268.44	*16.88
rc203	15	8	2982	2	4342.91	*151.65
rc204	15	3	2240	0	-	*6.21
rc204	15	5	2526	1	6698.49	*36.80
rc204	15	8	2859	2	2438.21	*402.83
rc201	20	3	3733	2	*0.02	0.26
rc201	20	5	4302	3	*0.03	1.34
rc201	20	8	4931	3	*0.05	2.42
rc202	20	3	3464	1	107.58	*3.88
rc202	20	5	3862	2	56.45	*1.25
rc202	20	8	4635	4	1276.84	*163.60
rc203	20	3	3042	0	-	*7.09
rc203	20	5	3366	1	-	*136.39
rc203	20	8	4120	3	-	*2986.15
rc204	20	3	2845	0	-	*6.82
rc204	20	5	3301	1	-	*1160.77
rc204	20	8	3790	2	-	*6469.83
rc201	30	3	5599	1	*0.11	4.44
rc201	30	5	6204	2	*0.34	12.57
rc201	30	8	6998	4	*0.24	37.11
rc202	30	3	4832	1	-	*21.95
rc202	30	5	5478	2	-	*99.57
rc202	30	8	6431	5	-	*1482.03
rc203	30	3	4418	1	-	*14.01
rc203	30	5	5191	2	-	*374.77
rc203	30	8	5877	4	-	*4458.23
rc204	30	3	4292	0	-	*24.82
rc204	30	5	4953	2	-	*5606.51
rc204	30	8	-	-	-	-

Table 5: RC201-RC204 instances, **spanning tree sets**. If the algorithm has not terminated within 7500 seconds it is indicated with "-". The best running time is marked with a "*".

Acknowledgments

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

References

[1] R. Baldacci, A. Mingozzi, and R. Roberti. Solving the vehicle routing problem with time windows using new state space relaxation and pricing strategies. Submitted.

[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. H. 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, 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] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

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

[17] J. Lysgaard. Cvrpsep: A package of separation routines for the capacitated vehicle routing problem, 2003.

[18] 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.

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

[20] 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.

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

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 contributions impose a cost for the company while they also give unlimited usage of a set of roads to all vehicles belonging to the company. Different versions for defining the edge sets are discussed and formulated. A MIP-formulation of the problem is presented, and a solution method based on branch-and-price-and-cut is applied to the problem. The computational results show that instances with up to 50 customers can be solved in reasonable time, and that the branch-cut-and-price algorithm generally outperforms CPLEX. It also seems that instances get more difficult when the penalized edge sets form a spanning tree, compared to when they are randomly scattered.

DTU Management Engineering
Department of Management Engineering
Technical University of Denmark

Produktionstorvet
Building 424
DK-2800 Kongens Lyngby
Denmark
Tel. +45 45 25 48 00
Fax +45 45 93 34 35

www.man.dtu.dk