Clique Inequalities applied to the Vehicle Routing Problem with Time Windows

Simon Spoorendonk* Guy Desaulniers⋆

*DTU Management Engineering, Technical University of Denmark
Produktionstorvet, DTU - Building 424, 2800 Kgs. Lyngby, Denmark

⋆Ecole Polytechnique de Montréal and GERAD
C.P. 6079, Succ. Centre-ville, Montréal, Canada, H3C 3A7

Abstract

This work presents an exact branch-cut-and-price (BCP) algorithm for the vehicle routing problem with time windows (VRPTW) where the well-known clique inequalities are used as cutting planes defined on the set partitioning master problem variables. It is shown how these cutting planes affect the dominance criterion applied in the pricing algorithm, which is a labeling algorithm for solving the elementary shortest path problem with resource constraints (ESPPRC). To our knowledge, this is a first attempt at incorporating, for the VRPTW, a set of valid inequalities specialized for the set partitioning polytope. Computational results show that the use of clique inequalities improves the lower bound at the root node and reduces the number of nodes in the search tree.

Keywords: *vehicle routing problem with time windows, Dantzig-Wolfe decomposition, cutting planes*

1 Introduction

Given an unlimited number of identical vehicles with a specified capacity, and a set of customers, each with a demand to fulfill and a delivery time window, the VRPTW consists of finding a set of routes starting and ending at a central depot such that each customer is visited exactly once within its time window, the capacity of each vehicle is respected, and the total length of the routes is minimized.

Because the ESPPRC is strongly $\mathcal{NP}$-hard, branch-and-price algorithms, where the (relaxed) pricing problem allows cycles, have been widely used. Two research trends have been followed to reduce the possibly large integrality gaps. In the first trend, the pricing problem is strengthened, see e.g., Irnich and Villeneuve [6], Feillet et al. [4]. In the second research trend, cutting planes are generated to tighten the LP-relaxed master problem. An example is the k-path cuts, see e.g., Desaulniers et al. [3], that can be defined using arc-flow variables and whose treatment only implies a modification of the arc costs in the pricing problem. Recently, Jepsen et al. [7] proposed subset row inequalities that are directly defined on the master problem variables. These inequalities form a subset of the Chvátal-Gomory rank 1 inequalities for the set partitioning polytope. Later, Petersen et al. [10] generalized this work to include all Chvátal-Gomory rank 1 cuts. These inequalities significantly reduce integrality gaps. However, they complexify the solution of the pricing problem. In a very recent paper, [1] used a set partitioning formulation for the capacitated vehicle routing problem and applied clique inequalities. The pricing problem is solved by an extensive enumeration scheme based on dynamic programming. This procedure performed well on small, tightly constrained instances, but had memory issues when the number of feasible candidate paths was too large.

In this paper, we follow this second research trend by studying the use of clique inequalities for the VRPTW. As opposed to the work of Baldacci et al. [1], we propose to consider the dual values of the clique inequalities directly into the labeling algorithm to avoid an extensive enumeration of candidate paths. We believe that such an alternative must be sought to derive a scalable BCP algorithm that can solve the most difficult VRPTW instances (e.g., those allowing more than 50 customers per route). To handle the dual values of these inequalities in the labeling algorithm, we introduce an approximate representation of a clique and develop a modified dominance criterion. To assess the usefulness of the clique cuts, we performed computational experiments on a subset of the well-known VRPTW instances of Solomon [11]. The paper is organized as follows. In Section 2, the VRPTW is formulated mathematically while, in Section 3, a basic BCP algorithm is described. Section 4 describes the approximate representation of the cliques and they are applied. Computational results are reported in Section 5, and concluding remarks are made in Section 6.

2 Problem Formulation

In this section, we present the set partitioning formulation for the VRPTW. Consider a network $G = (N, A)$, where N and A are the node and arc sets, respectively. The node set N contains one node for each customer $i \in C$ and two nodes, o and o', representing the depot at the start and the end of the routes, respectively. Thus, $N = C \cup \{o, o'\}$. Each customer $i \in C$ has a demand d_i, an associated service time s_i, and a time window $[a_i, b_i]$ in which it should be visited. To simplify notation, we set $d_o = d_{o'} = s_o = s_{o'} = 0$ and $[a_o, b_o] = [a_{o'}, b_{o'}] = [0, H]$, where H is the length of the planning horizon. The arc set A is given by $A = \{(i,j) : i, j \in N,\ i \neq j, a_i + \tau_{ij} \leq b_j, d_i + d_j \leq D\} \setminus \{(o', o)\}$, where D is the vehicle capacity and τ_{ij} is the travel time between locations i and j plus the service time s_i at i. The last two conditions ensure that the same vehicle can visit consecutively locations i and j according to the time and load restrictions. Denote by c_{ij} the travel cost along arc $(i,j) \in A$. The set partitioning formulation of the VRPTW is given as:

$$\min \sum_{p \in P} \Big(\sum_{(i,j) \in p} c_{ij} \Big) \lambda_p \tag{1}$$

$$\text{s.t.} \sum_{p \in P} \alpha_{ip} \lambda_p = 1 \qquad \forall i \in C \tag{2}$$

$$\lambda_p \quad \text{binary} \qquad \forall p \in P \tag{3}$$

where P is the set of all feasible $o - o'$ paths (routes) in G, that is, the time windows of the customers are obeyed along the path and the vehicle capacity is not exceeded. The binary parameter α_{ip} is equal to one if and only if node $i \in N$ is used in route $p \in P$, and the binary variable λ_p indicates whether or not route $p \in P$ is selected in the solution. Constraints (2) ensure that each customer $i \in C$ is visited by exactly one vehicle. Generally, this model contains a huge number of variables and can be solved using a BCP method.

3 Branch-Cut-and-Price

As mentioned in the introduction, a BCP method consists of a column generation method embedded in a branch-and-cut method (see Desaulniers et al. [2], Lübbecke and Desrosiers [8]). For the VRPTW, each variable λ_p, $p \in P$, is associated with a feasible route in G. The reduced cost $\bar{c}_p$ of such a variable is: $\bar{c}_p = \sum_{(i,j) \in p} c_{ij} - \sum_{(i,j) \in p} \pi_j = \sum_{(i,j) \in p} (c_{ij} - \pi_j)$ where $\pi_j \in \mathbb{R}$ for all $j \in C$ are the dual values of (2) and $\pi_{o'} = 0$. To determine if there exists negative reduced cost variables, one can find the shortest feasible path from o to o' in G, where the cost of each arc $(i,j) \in A$ is $\bar{c}_{ij} = c_{ij} - \pi_j$. Feasibility is enforced by resource constraints. A resource is a quantity that accumulates along a path and is restricted at each node to take a value within a prespecified resource interval, called a *resource window*. For the

VRPTW, the set of resources $\mathcal{R}$ contains a loading resource to ensure that vehicle capacity is satisfied, a time resource to respect the customer time windows, and a binary resource for each customer $i \in C$ to ensure path elementarity. The pricing problem thus corresponds to an ESPPRC, and can be solved using a labeling algorithm, see e.g., Feillet et al. [4]. In such a dynamic programming method, labels represent partial paths that are extended (using so-called *extension functions*) in all feasible directions from the source node o. Each label L (a vector with $|\mathcal{R}| + 1$ components) stores the cost of the partial path $T_{cost}(L)$ and the current value $T_r(L)$ of each resource $r \in \mathcal{R}$. To avoid enumerating all feasible paths in G, only Pareto-optimal labels are kept during the execution of the algorithm. When using non-decreasing extension functions as it is the case for the VRPTW, the label dominance criterion can be stated as follows.

Proposition 1 (Desaulniers et al. [2])**.** Let L and L' be two labels representing partial paths ending at the same node. Label L dominates label L' (which can be discarded) if

$$T_{cost}(L) \leq T_{cost}(L') \tag{4}$$

$$T_r(L) \leq T_r(L') \qquad \forall r \in \mathcal{R}. \tag{5}$$

When equality holds for all label components, one of the two labels must be kept.

Valid inequalities that can be defined as linear combinations of the variables in the original arc-flow formulation can be rewritten in terms of the master problem variables λ_p. In this case, the cuts do not pose a problem for the dominance criterion, since only the arc costs are affected in the pricing problem. The subset row inequalities introduced by Jepsen et al. [7] are, however, a set of valid inequalities directly defined on the master problem variables, and they give rise to an additional resource in the pricing problem per cut. Jepsen et al. [7] presented an improved dominance criterion based on the observation, that the additional resources only affects the cost of the path and not the feasibility. As shown by their computational results, the use of these inequalities can significantly improve the lower bound computed at the root node. Petersen et al. [10] extended this work and showed how any Chvátal-Gomory rank 1 cut can be applied to the VRPTW.

4 Clique Inequalities

Clique inequalities are well-known valid inequalities for the set partitioning polytopes, see e.g., Nemhauser and Wolsey [9]. A clique is defined on a *conflict* graph $\mathcal{G} = (V, E)$ that is undirected. Its vertex set V contains one vertex for each route $p \in P$, that is, $V = P$. Its edge set E contains an edge between two vertices p and q of V if routes p and q have at least one customer in common (p and q are said to be *in conflict* or *conflicting*), that is, $E = \{(p,q) : p, q \in P, p \neq q, \exists\, i \in C \text{ such that } \alpha_{ip} = \alpha_{iq} = 1\}$. Thus, an edge (p, q) identifies a conflict between p and q for which the variables λ_p and λ_q cannot both take value 1 in a feasible integer solution. A clique W is a maximal subset of vertices of V such that $(p, q) \in E$ for all pairs $p, q \in W$, $p \neq q$. Given a clique $W \subseteq P$ in $\mathcal{G}$, the corresponding clique inequality is expressed as:

$$\sum_{p \in W} \lambda_p \leq 1. \tag{6}$$

The separation of the most-violated clique inequality is strongly $\mathcal{NP}$-hard. In this paper we use a separation heuristic proposed by Hoffman and Padberg [5]. Next, we propose a representation of the clique inequalities which will facilitate their treatment in the labeling algorithm used for solving the pricing problem. In a BCP context, such a representation is important to easily determine in which cliques a newly generated route belongs to (and should therefore be added to). If generated routes were not added to existing cliques, one would experience a repeated regeneration of already existing routes (except for the entries in the cliques) and a separation of similar cliques (now including the newest routes). Last, we introduce the additional resources required in the ESPPRC pricing problem for handling these inequalities and develop a new dominance criterion for the labeling algorithm.

4.1 Clique Representation

Let $\boldsymbol{\alpha}_q = \left(\alpha_{iq}\right)_{i \in C}$ for all $q \in P$. Consider a clique W and a newly generated route $p \in P \setminus W$. To verify if p is in conflict with all routes in W using a complete representation of W, one has to check if $\boldsymbol{\alpha}_p$ is orthogonal with $\boldsymbol{\alpha}_q$ for each route q in W. This CC check (for clique conflict check) requires $O(|W||C|)$ operations. Because the size of $|W|$ can become very large (worst case $O(|P|)$) and the CC check is performed repeatedly within the labeling algorithm, we propose to reduce the (practical) complexity of the CC check by using an approximate representation that considers only a subset of the rows in C and a subset of the routes in W. To describe this representation, let us start with preliminary notation and definitions. Let $idRows(p) \subseteq C$ be the set of customers visited in route $p \in P$.

Definition 1. Let i be a customer in C, and p and q be two routes in P. The constraint (2) indexed by i (more briefly, row i) is said to be a *conflicting row for p and q* if both routes visit customer i, that is, if $i \in \big(idRows(p) \cap idRows(q)\big)$.

Definition 2. Let W be a clique. The *set $\chi(W) \subseteq C$ of conflicting rows of* W is given by $\chi(W) = \bigcup_{p,q \in W : p \neq q} \big(idRows(p) \cap idRows(q)\big)$. In other words, row $i \in C$ belongs to $\chi(W)$ if i is conflicting for at least one pair of distinct routes p and q in W.

Let W be a clique and denote by $idConfPairs_i(W) \subseteq W \times W$ the set of pairs of routes p and q in W for which row $i \in C$ is conflicting.

Definition 3. Let W be a clique. A *minimal subset $\chi_{min}(W)$ of conflicting rows of* W is a subset of $\chi(W)$ such that, i) $\bigcup_{i \in \chi_{min}(W)} idConfPairs_i(W) = \{(p,q) : p,q \in W, p \neq q\}$, and ii) $idConfPairs_i(W) \setminus \Big(\bigcup_{j \in \chi_{min}(W)\setminus\{i\}} idConfPairs_j(W)\Big) \neq \emptyset, \forall i \in \chi_{min}(W)$. Condition i) ensures that all pairs of routes in W is conflicting with respect to at least one row in subset $\chi_{min}(W)$ (that is, W would still be a clique if the set of constraints (2) was restricted to the subset of indices in $\chi_{min}(W)$), while condition ii) guarantees the minimality of this subset (that is, the first condition would not hold if any row is removed from the subset).

Finding a minimal subset of minimal cardinality is $\mathcal{NP}$-hard: it corresponds to a set covering problem where each pair of routes in the clique defines a covering constraint, and each constraint (2) a binary variable with a cost coefficient of 1. We rather resort to a heuristic procedure that starts from the set of conflicting rows $\chi(W)$ and sequentially removes rows from it until satisfying condition ii) in Definition 3. The proposed approximate representation is based on the following observation.

Observation 1. Let W be a clique and $\chi_{min}(W)$ a minimal subset of its conflicting rows. Let p be a newly generated route. If p is in conflict with all routes in W when considering only the rows in $\chi_{min}(W)$, then p is also in conflict with all these routes when all rows $i \in C$ are considered.

Observation 1 provides a sufficient condition to identify a route that can be used to enlarge a clique. Given a clique W, the number of conflicting rows in a minimal subset $\chi_{min}(W)$ is at most $\sum_{i=1}^{|W|-1} i = \frac{|W|(|W|-1)}{2}$. Consequently, when $|C| > |W|^2$, considering only a minimal subset of conflicting rows when performing the CC check can clearly accelerate this operation. An approximate representation of a clique W based on a minimal subset $\chi_{min}(W)$ is now available. It is composed of the vectors $\boldsymbol{v}_p(\chi_{min}(W))$, $\forall p \in W$, where $\boldsymbol{v}_p(\chi_{min}(W))$ is the sub-vector of the column $\boldsymbol{\alpha}_p$ restricted to the rows in $\chi_{min}(W)$. This representation can be made more compact by noticing that a newly generated route is in conflict with a route $p \in W$ if it is also conflicting with a route $q \in W$, $q \neq p$, such that $\boldsymbol{v}_q(\chi_{min}(W)) \leq \boldsymbol{v}_p(\chi_{min}(W))$. It is easy to prove that there are no pairs of routes p and q in W such that $\boldsymbol{v}_q(\chi_{min}(W)) < \boldsymbol{v}_p(\chi_{min}(W))$ (otherwise, $\chi_{min}(W)$ would not be a minimal subset of conflicting rows of W). Consequently, one can eliminate from the representation all vectors $\boldsymbol{v}_p(\chi_{min}(W))$, except one, that are equal. For a clique W, we denote by $\Gamma(W)$ the resulting set of routes used in the representation. This representation, called a key set, is defined as follows.

Definition 4. Given a clique W and a minimal set $\chi_{min}(W)$ of its conflicting rows, the *key set* of W is composed of the sub-vectors $\boldsymbol{v}_p(\chi_{min}(W))$, $\forall p \in \Gamma(W)$.

We propose to add in a clique W only the routes p for which $\boldsymbol{v}_p(\chi_{min}(W)) \geq \boldsymbol{v}_q(\chi_{min}(W))$ for at least one route $q \in \Gamma(W)$. This restriction, called the *clique admissibility rule*, guarantees that all added routes conflict with each other (since they all contain at least one of the key set sub-vectors which all conflict), and allows to always keep the same key set as newly generated routes are added to the clique. A route p satisfying the admissibility rule of a clique W is said to be *admissible to enlarge clique* W.

4.2 Modified Pricing Problem and Labeling Algorithm

Let Ω be the set of cliques, and let ζ_W be the non-positive dual variable associated with the clique inequality (6) for $W \in \Omega$. The reduced costs of the variables λ_p now depend on the dual values of these inequalities, that is, $\bar{c}_p = \sum_{(i,j)\in p}(c_{ij} - \pi_j) - \sum_{W\in\Omega:p\in W} \zeta_W$. For each clique $W \in \Omega$, $|\chi_{min}(W)|+1$ binary resources are defined and corresponding components are added to each label. For a label representing a partial path p, the first $|\chi_{min}(W)|$ of these resource values indicate, until it is proven that p can enlarge W, whether or not each customer of the minimal subset $\chi_{min}(W)$ has been visited along p. Thus, they represent the sub-vector $\boldsymbol{v}_p(\chi_{min}(W))$ and are compared in the CC check to the key set vectors of W. When the CC check is positive (that is, p is admissible to enlarge W), these resource values are set to 0 and never change in the subsequent label extensions. Furthermore, the dual value ζ_W is subtracted from the label (reduced) cost component. For a label L, these resource components are denoted $T^W_{cust_\ell}(L)$, $\ell \in \chi_{min}(W)$. The other additional resource component, denoted $T^W_{inadm}(L)$, simply indicates whether or not p is inadmissible to enlarge the clique: it is equal to 1 if p is inadmissible, and 0 otherwise. The set of all these new resources for all cliques in Ω is denoted $\mathcal{Q}$. With these additional resources, the labeling algorithm can be applied for solving exactly the modified pricing problem if the dominance criterion of Proposition 1 considers the set $\mathcal{R} \cup \mathcal{Q}$ instead of only $\mathcal{R}$. The dominance criterion we propose improves on the criterion of Proposition 1 and is stated in the following proposition.

Proposition 2. Let L and L' be two labels representing partial paths ending at the same node. Label L dominates label L' (which can be discarded) if

$$T_{cost}(L) - \sum_{W\in\Omega_{LL'}} \zeta_W \leq T_{cost}(L') \tag{7}$$

$$T_r(L) \leq T_r(L') \qquad \forall r \in \mathcal{R}, \tag{8}$$

where $\Omega_{LL'} = \{W \in \Omega : T^W_{inadm}(L) = 1 \text{ and } \big(T^W_{inadm}(L') = 0 \text{ or } \exists\, \ell \in \chi_{min}(W) \text{ such that } T^W_{cust_\ell}(L) > T^W_{cust_\ell}(L')\big)\}$ is the set of cliques W for which the penalty ζ_W could be paid in a feasible extension of L along an arc sequence, while it would not be paid when extending similarly L' (especially if the penalty was already paid, i.e., if $T^W_{inadm}(L') = 0$).

This can be proved by considering the maximum cost between labels L and L' when extended with all feasible extensions of L. The proof is much similar to the ones presented for the improved dominance criteria by Jepsen et al. [7] and Petersen et al. [10].

5 Computational Experiments

We compare the results of two different BCP algorithms. The first method, denoted SR, only adds subset row cuts. Specifically, we apply the 3/2-subset row cuts used in Jepsen et al. [7] and Desaulniers et al. [3] that corresponds to clique inequalities with a key set of exactly three sub-vectors, each with two non-zero entries. For these cuts the representation as subset row inequalities is superior to the representation as clique inequalities. For this method, we used the code of Desaulniers et al. [3]. The second method, denoted CLIQUE, is the same as the first one except that clique inequalities are added when no violated subset row inequalities are found. The experiments were conducted on a subset (R1 and RC1) of the instances of Solomon [11]. For the remaining instance sets (C1, C2, R2, and R2) either no cuts at all are separated, the instances could not be solved, or almost no clique inequalities are separated after

		no cuts	SR				CLIQUE				
Instance	UB	LB	CPU	BB	LB	gap cl.	CPU	BB	LB	gap cl.	gap cl. SR
R101	1637.7	1631.2	9	15	1634.0	43.1 %	9	15	1634.0	43.1 %	0 %
R102	1466.6	**1466.6**	3	0	***id***	-	*id*	*id*	***id***	-	-
R103	1208.7	1206.8	20	1	**1208.7**	100.0 %	*id*	*id*	***id***	*id*	-
R104	971.5	956.9	2332	11	970.5	93.1 %	3990	5	971.3	98.6 %	80.0 %
R105	1355.3	1346.2	28	3	1355.2	98.9 %	35	3	**1355.3**	100.0 %	100.0 %
R106	1234.6	1227.0	88	3	1234.4	97.4 %	104	3	1234.4	97.4 %	0 %
R107	1064.6	1053.3	347	5	1063.3	88.5 %	479	3	1063.9	93.8 %	46.2 %
R108	932.1	913.6	1527	3	932.0	99.5 %	1249	1	**932.1**	100.0 %	100.0 %
R109	1146.9	1134.3	942	47	1144.0	77.0 %	2373	91	1144.2	78.6 %	6.9 %
R110	1068.0	1055.6	420	5	1067.0	91.9 %	533	5	1067.6	96.8 %	60.0 %
R111	1048.7	1034.8	5246	95	1045.3	75.5 %	13667	91	1045.9	80.0 %	17.6 %
R112	948.6	926.8	17484	21	945.8	87.2 %	55226	11	946.9	92.2 %	39.3 %
RC101	1619.8	1584.1	28	1	**1619.8**	100.0 %	*id*	*id*	***id***	*id*	-
RC102	1457.4	1406.3	194	1	**1457.4**	100.0 %	*id*	*id*	***id***	*id*	-
RC103	1258.0	1225.6	1055	5	1257.5	98.5 %	961	3	1257.7	99.1 %	40.0 %
RC104	1132.3	1101.9	14294	25	1129.7	91.4 %	14343	15	1129.9	92.1 %	7.7 %
RC105	1513.7	1472.0	26	1	**1513.7**	100.0 %	*id*	*id*	***id***	*id*	-
RC106	1372.7	1318.8	5240	87	1367.5	90.4 %	4279	47	1367.5	90.4 %	0 %
RC107	1207.8	1183.4	229	1	**1207.8**	100.0 %	*id*	*id*	***id***	*id*	-
RC108	1114.2	1073.5	2203	1	**1114.2**	100.0 %	*id*	*id*	***id***	*id*	-

Table 1: Experimental results. "UB" is the upper bound, "LB" is the lower bound, "CPU" is the time in seconds, and "BB" is the number of branch nodes. The "gap cl." columns gives the gap of the no cuts method closed by the cuts (SR and CLIQUE). The "gap cl. SR" column specifies the gap of the SR method closed by the clique cuts. In these columns, a "-" appears when the corresponding LB column contains "id".

the addition of the subset row inequalities. Due to the minor impact of the clique inequalities on these instances, detailed results have been left out. Experiments were performed on a 2.2 GHz 64-bit Dual Core AMD Opteron processor 275 with 12 GB of memory. The results are reported in Table 1. These results indicate that the clique cuts are only useful in the root node for 13 of the 20 instances. Out of these 13 instances, the clique cuts succeeded to improve the lower bound in 10 cases (closing the gap in 2 cases). The average gap closed by the clique cuts is 38% when compared to the SR method. For 8 of the 13 instances the number of branch-and-bound nodes was reduced, but the computational times are, in general, longer with the CLIQUE method.

6 Conclusion

This paper is a step towards adding special purpose cuts, considered clique inequalities, for the master problem in a BCP algorithm for the VRPTW. In particular, it proposed an approximate representation of a clique and a modified dominance criterion for the ESPPRC pricing problem. The computational results showed that, less branch-and-bound nodes are explored, but, that the computational times are generally slower. We think that the proposed methodology can be improved to yield much better results, e.g., with better separation heuristics and policies. Such developments will be the subject of future research.

Acknowledgements

This research is partly funded by the Danish Research Council for Technology and Production Sciences part of the Danish Councils for Independent Research (project 274-08-0353).

References

[1] R. Baldacci, N. Christofides, and A. Mingozzi, *An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts*, Mathematical Programming **115** (2008) pp. 351–385.

[2] G. Desaulniers, J. Desrosiers, J. Ioachim, I. M. Solomon, F. Soumis, and D. Villeneuve, *A unified framework for deterministic time constrained vehicle routing and crew scheduling problems*, in *Fleet Management and Logistics*, T. G. Crainic and G. Laporte, eds., Kluwer Academic Publishers, Dordrecht, 1998, pp. 57–93.

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

[4] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen, *An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems*, Networks **44** (2004) pp. 216–229.

[5] K. L. Hoffman and M. Padberg, *Solving airline crew scheduling problems by branch-and-cut*, Management Science **39** (1993) pp. 657–682.

[6] S. Irnich and D. Villeneuve, *The shortest path problem with resource constraints and k-cycle elimination for* $k \geq 3$, INFORMS Journal on Computing **18** (2006) pp. 391–406.

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

[8] M. E. Lübbecke and J. Desrosiers, *Selected topics in column generation*, Operations Research **53** (2005) pp. 1007–1023.

[9] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley & Sons, Inc., 1988.

[10] B. Petersen, D. Pisinger, and S. Spoorendonk, *Chvátal-Gomory rank-1 cuts used in a Dantzig-Wolfe decomposition of the vehicle routing problem with time windows*, in *The Vehicle Routing Problem: Latest Advances and New Challenges*, B. Golden, R. Raghavan, and E. Wasil, eds., Springer, 2008, pp. 397–420.

[11] M. M. Solomon, *Algorithms for the vehicle routing and scheduling problem with time window constraints*, Operations Research **35** (1987) pp. 234–265.