Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Subset-Row Inequalities Applied to the Vehicle-Routing Problem with Time Windows

Mads Jepsen, Bjørn Petersen, Simon Spoorendonk, David Pisinger,

OPERATIONS RESEARCH
Vol. 56, No. 2, March–April 2008, pp. 497–511
ISSN 0030-364X | EISSN 1526-5463 | 08 | 5602 | 0497

DOI 10.1287/opre.1070.0449

Subset-Row Inequalities Applied to the Vehicle-Routing Problem with Time Windows

Mads Jepsen, Bjørn Petersen, Simon Spoorendonk, David Pisinger
Department of Computer Science (DIKU), University of Copenhagen, DK-2100 Copenhagen Ø, Denmark
{mitzi@diku.dk, bjorn@diku.dk, spooren@diku.dk, pisinger@diku.dk}

This paper presents a branch-and-cut-and-price algorithm for the vehicle-routing problem with time windows. The standard Dantzig-Wolfe decomposition of the arc flow formulation leads to a set-partitioning problem as the master problem and an elementary shortest-path problem with resource constraints as the pricing problem. We introduce the subset-row inequalities, which are Chvatal-Gomory rank-1 cuts based on a subset of the constraints in the master problem. Applying a subset-row inequality in the master problem increases the complexity of the label-setting algorithm used to solve the pricing problem because an additional resource is added for each inequality. We propose a modified dominance criterion that makes it possible to dominate more labels by exploiting the step-like structure of the objective function of the pricing problem. Computational experiments have been performed on the Solomon benchmarks where we were able to close several instances. The results show that applying subset-row inequalities in the master problem significantly improves the lower bound and, in many cases, makes it possible to prove optimality in the root node.

Subject classifications: transportation: vehicle routing; programming: integer.
Area of review: Transportation.
History: Received March 2006; revisions received August 2006, October 2006; accepted November 2006. Published online in *Articles in Advance* December 20, 2007.

1. Introduction

The vehicle-routing problem with time windows (VRPTW) can be described as follows: A set of customers, each with a demand, needs to be serviced by a number of vehicles all starting and ending at a central depot. Each customer must be visited exactly once within a given time window, and the capacity of the vehicles must not be exceeded. The objective is to service all customers traveling the least possible distance. In this paper, we consider a homogenous fleet, i.e., all vehicles are identical.

The standard Dantzig-Wolfe decomposition of the arc flow formulation of the VRPTW is to split the problem into a master problem (a set-partitioning problem) and a pricing problem (an elementary shortest-path problem with resource constraints (ESPPRC), where capacity and time are the constrained resources). A restricted master problem can be solved with delayed column generation and embedded in a branch-and-bound framework to ensure integrality. Applying cutting planes either in the master or the pricing problem leads to a branch-and-cut-and-price algorithm (BCP).

Kohl et al. (1999) implemented a successful BCP algorithm for the VRPTW by applying subtour elimination constraints and *two-path cuts*. Cook and Rich (1999) generalized the two-path cuts to the *k-path cuts*. Common for these BCP algorithms is that all applied cuts are valid inequalities for the VRPTW, i.e., the original arc flow formulation, and contain a structure making it possible to handle values of the dual variables in the pricing problem without increasing the complexity of the problem. Fukasawa et al. (2006) refer to this as a *robust approach* in their paper, where a range of valid inequalities for the capacitated vehicle-routing problem are used in a BCP algorithm. The topic of column generation and BCP algorithms has been surveyed by Barnhart et al. (1998) and Lubbecke and Desrosiers (2005).

Dror (1994) showed that the ESPPRC is strongly $\mathcal{NP}$-hard, hence a relaxation of the ESPPRC was used as a pricing problem in earlier BCP approaches for the VRPTW. The relaxed pricing problem where nonelementary paths are allowed is denoted the shortest-path problem with resource constraints (SPPRC) and can be solved in pseudo-polynomial time using a label-setting algorithm, which was initially done by Desrochers (1986). To improve lower bounds of the master problem, Desrochers et al. (1992) used two-cycle elimination, which was later extended by Irnich and Villeneuve (2006) to k-cycle elimination (k-cyc-SPPRC) where cycles containing k or less nodes are not permitted.

Beasley and Christofides (1989) proposed to solve the ESPPRC using Lagrangian relaxation. However, recently label-setting algorithms have become the most popular approach to solve the ESPPRC; see, e.g., Dumitrescu (2002) and Feillet et al. (2004). When solving the ESPPRC with a label-setting algorithm, a binary resource for each node is added, which increases the complexity of the algorithm compared to solving the SPPRC or the k-cyc-SPPRC. Righini and Salani (2004) developed a label-setting algorithm using the idea of Dijkstra's bidirectional shortest-path algorithm that expands both forward and backward from the depot and connects routes in the middle, thereby potentially reducing the running time of the algorithm. Furthermore, Righini and Salani (2005) and Boland et al. (2006) proposed a decremental state space algorithm that iteratively solves a SPPRC by applying resources that force nodes to be visited at most once. Recently, Chabrier (2005), Danna and Le Pape (2005), and Salani (2005) successfully solved several previously unsolved instances of the VRPTW from the benchmarks of Solomon (1987) using a label-setting algorithm for the ESPPRC.

In this paper, we extend the BCP framework to include valid inequalities for the master problem, more specifically by applying the subset-row (SR) inequalities to the set-partitioning master problem. Nemhauser and Park (1991) developed a similar BCP algorithm for the edge-coloring problem, but to our knowledge no such algorithms for the VRPTW have been presented. Applying the SR inequalities leads to an increased complexity of the pricing problem because each inequality is represented by an additional resource. To improve the performance of the label-setting algorithm, we introduce a modified dominance criterion that handles the reduced cost calculation in a reasonable way. Moreover, the SR inequalities potentially provide better lower bounds and smaller branch trees.

This paper is organized as follows. In §2, we give an overview of the Dantzig-Wolfe decomposition of the VRPTW and describe how to calculate the reduced cost of columns when column generation is used. In §3, we introduce the SR inequalities and show that the separation problem is $\mathcal{NP}$-complete. In §4, we review the basics of a label-setting algorithm for solving the ESPPRC and show how to handle the modified pricing problem in the same label-setting algorithm. For details regarding label-setting algorithms (including bidirectionality), we refer to Desaulniers et al. (1998), Irnich and Desaulniers (2005), Irnich (2006), and Righini and Salani (2004). An algorithmic outline and computational results, using the Solomon benchmark instances, are presented in §5. Section 6 concludes the paper.

2. Decomposition

Let C be the set of customers; let the set of nodes be $V = C \cup \{o, o'\}$, where $\{o\}$ denotes the depot at the start of the routes and $\{o'\}$ denotes the depot at the end; and let $E = \{(i, j): i, j \in V, i \neq j\}$ be the edges between the nodes. Let K be the set of vehicles with $|K|$ unbounded, each vehicle having capacity D, and let d_i be the demand of customer $i \in C$ and $d_o = d_{o'} = 0$. Let a_i be the beginning and b_i be the end of the time window for node $i \in V$. Let s_i be the service time for $i \in V$ and let t_{ik} be the time vehicle $k \in K$ visits node $i \in V$, if k visits i. Let c_{ij} be the travel cost on edge $(i, j) \in E$ and let x_{ijk} be a variable indicating whether vehicle $k \in K$ traverses edge $(i, j) \in E$. Last, let $\tau_{ij} = c_{ij} + s_i > 0$ be the travel time on edge $(i, j) \in E$ plus the service time of customer i. The three-index flow model (Toth and Vigo 2002) for the VRPTW is

$$\min \sum_{k\in K} \sum_{(i,j)\in E} c_{ij} x_{ijk} \tag{1}$$

$$\text{s.t.} \sum_{k\in K} \sum_{(i,j)\in\delta^+(i)} x_{ijk} = 1 \quad \forall i \in C, \tag{2}$$

$$\sum_{(i,j)\in\delta^+(o)} x_{ijk} = \sum_{(i,j)\in\delta^-(o')} x_{ijk} = 1 \quad \forall k \in K, \tag{3}$$

$$\sum_{(j,i)\in\delta^-(i)} x_{jik} - \sum_{(i,j)\in\delta^+(i)} x_{ijk} = 0 \quad \forall i \in C, \forall k \in K, \tag{4}$$

$$\sum_{(i,j)\in E} d_i x_{ijk} \leqslant D \quad k \in K, \tag{5}$$

$$a_i \leqslant t_{ik} \leqslant b_i \quad \forall i \in V, \forall k \in K, \tag{6}$$

$$x_{ijk}(t_{ik} + \tau_{ij}) \leqslant t_{jk} \quad \forall (i,j) \in E, \forall k \in K, \tag{7}$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i,j) \in E, \forall k \in K. \tag{8}$$

Here (2) ensures that every customer $i \in C$ is visited, while (3) ensures that each route starts and ends in the depot. Constraint (4) maintains flow conservation, while (5) ensures that the capacity of each vehicle is not exceeded. Constraints (6) and (7) ensure that the time windows are satisfied. Note that (7) together with the assumption that $\tau_{ij} > 0$ for all $(i, j) \in E$ eliminates subtours. The last constraints define the domain of the arc flow variables. Note that a zero-cost edge $x_{oo'k}$ between the start and end depot must be present for all vehicles for (3) to hold if not all vehicles are used.

The standard Dantzig-Wolfe decomposition of the VRPTW (see, e.g., Desrochers et al. 1992), leads to the following master problem:

$$\min \sum_{p\in P} \sum_{(i,j)\in E} c_{ij} \alpha_{ijp} \lambda_p \tag{9}$$

$$\text{s.t.} \sum_{p\in P} \sum_{(i,j)\in\delta^+(i)} \alpha_{ijp} \lambda_p = 1 \quad \forall i \in C, \tag{10}$$

$$\lambda_p \in \{0, 1\} \quad \forall p \in P, \tag{11}$$

where P is the set of all feasible routes, the binary constant α_{ijp} is one if and only if edge (i, j) is used by route $p \in P$, and the binary variable λ_p indicates whether route p is used. The master problem can be recognized as a set-partitioning problem, and the LP relaxation can be solved using delayed column generation. Let $\pi \in \mathbb{R}$ be the dual variables of (10) and let $\pi_0 = 0$. Then, the reduced cost of a route p is

$$\bar{c}_p = \sum_{(i,j)\in E} c_{ij} \alpha_{ijp} - \sum_{(i,j)\in E} \pi_j \alpha_{ijp} = \sum_{(i,j)\in E} (c_{ij} - \pi_j) \alpha_{ijp}. \tag{12}$$

The pricing problem becomes an ESPPRC where the cost of each edge is $\bar{c}_{ij} = c_{ij} - \pi_j$ for all edges $(i, j) \in E$. When applying cuts during column generation, we will distinguish between valid inequalities for the VRPTW constraints (2)–(8) and valid inequalities for the set-partitioning constraints (10) and (11).

Consider a valid inequality for the VRPTW constraints (2)–(8) in terms of the arc flow variables x:

$$\sum_{k\in K}\sum_{(i,j)\in E}\beta_{ij}x_{ijk}\leqslant \beta_0. \tag{13}$$

When decomposed into the master problem, inequality (13) is reformulated as

$$\sum_{p\in P}\sum_{(i,j)\in E}\beta_{ij}\alpha_{ijp}\lambda_p\leqslant \beta_0. \tag{14}$$

Let $\mu\leqslant 0$ be the dual variable of (14). The reduced cost of a column p is then

$$\begin{aligned}\bar{c}_p &= \sum_{(i,j)\in E}c_{ij}\alpha_{ijp}-\sum_{(i,j)\in E}\pi_j\alpha_{ijp}-\mu\sum_{(i,j)\in E}\beta_{ij}\alpha_{ijp}\\ &=\sum_{(i,j)\in E}(c_{ij}-\pi_j-\mu\beta_{ij})\alpha_{ijp}. \end{aligned}\tag{15}$$

Compared to (12), an additional coefficient $\mu\beta_{ij}$ is subtracted from the cost of edge (i, j), and the complexity of the pricing problem remains unchanged if we use the edge costs $\bar{c}_{ij}=c_{ij}-\pi_j-\mu\beta_{ij}$.

Now, consider adding a valid inequality for the set-partitioning master problem (9)–(11) that cannot be written as a linear combination of the arc flow variables:

$$\sum_{p\in P}\beta_p\lambda_p\leqslant \beta_0. \tag{16}$$

Let $\sigma\leqslant 0$ be the dual variable of (16). The reduced cost of a column p is

$$\hat{c}_p=\bar{c}_p-\sigma\beta_p=\sum_{(i,j)\in E}\bar{c}_{ij}\alpha_{ijp}-\sigma\beta_p. \tag{17}$$

In addition to the reduced cost computed for a column p in (15), the cost $-\sigma\beta_p$ must be considered. To reflect the possible extra cost $-\sigma\beta_p$, it might be necessary to modify the pricing problem by adding constraints or variables, thereby increasing its complexity.

3. Subset-Row Inequalities

The set of valid inequalities for the set-packing problem is a subset of the set of valid inequalities for the set-partitioning problem because the latter problem is a special case of the former. Two well-known valid inequalities for the set-packing problem are the clique and the odd-hole inequalities, where the first is known to be facet defining for the set-partitioning problem (Nemhauser and Wolsey 1988).

Because the master problem is a set-partitioning problem, it would be obvious to go in this direction when looking for valid inequalities for the master problem. Consider the separation of a clique or an odd-hole inequality. The undirected conflict graph $G'(P, E')$ is defined as follows: Each column is a vertex in G' and the edge set is given as

$$E'=\left\{(p,q)\colon \sum_{(i,j)\in\delta^+(i)}\alpha_{ijp}=1\wedge\sum_{(i,j)\in\delta^+(i)}\alpha_{ijq}=1,\quad i\in C,\ p,q\in P,\ p\neq q\right\}.$$

That is, an edge is present if the two columns p and q have coefficient one in the same row. In a VRPTW context it reads: Two routes are conflicting if they are visiting the same customer. A clique in G' leads to the valid clique inequality

$$\sum_{p\in\hat{P}}\lambda_p\leqslant 1, \tag{18}$$

where $\hat{P}\subseteq P$ are the columns corresponding to the vertices of a clique in G'. A cycle visiting an odd number of vertices P in G' leads to the valid odd-hole inequality

$$\sum_{p\in\hat{P}}\lambda_p\leqslant\left\lfloor\frac{|\hat{P}|}{2}\right\rfloor, \tag{19}$$

where $\hat{P}\subseteq P$ are the columns corresponding to the vertices visited on the cycle in G'. However, when column generation is applied, it is not obvious how to reflect the reduced cost of (18) or (19) in the pricing problem because there is no specific knowledge of the columns of the master problem when solving the pricing problem.

Inspired by the above inequalities (18) and (19), we introduce the *subset-row inequalities* (SR inequalities). These inequalities are specifically linked to the rows (rather than the columns) of the set-packing problem, hence it is possible to identify the coefficient of a column in an SR inequality.

Definition 1. Consider the set-packing structure

$$X=\{\lambda\in\mathbb{B}^{|P|}\colon A\lambda\leqslant 1\} \tag{20}$$

with the set of rows M and columns P, and a $|M|\times|P|$ binary coefficient matrix A. The SR inequality is defined as

$$\sum_{p\in P}\left\lfloor\frac{1}{k}\sum_{i\in S}\alpha_{ip}\right\rfloor\lambda_p\leqslant\left\lfloor\frac{|S|}{k}\right\rfloor, \tag{21}$$

where $S\subseteq M$ and $0<k\leqslant|S|$.

Example 1 illustrates some SR inequalities derived from the conflict graph of a set-packing problem.

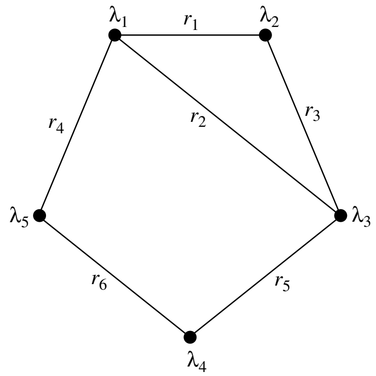
Example 1. SR inequalities derived from the conflict graph of a set-packing problem. In the LP-solution to $A\lambda\leqslant 1$, all λ variables are $\frac{1}{2}$, which results in two violated SR inequalities:

- With $n=3$ and $k=2$ due to variables λ_1, λ_2, and λ_3 giving the set of rows $S=\{r_1, r_2, r_3\}$.
- With $n=5$ and $k=2$ due to variables λ_1, λ_2, λ_3, λ_4, and λ_5 giving the set of rows $S=\{r_1, r_3, r_4, r_5, r_6\}$.

Set-Packing Problem $A\lambda \leqslant 1$.

	λ_1	λ_2	λ_3	λ_4	λ_5		
r_1	1	1				$\leqslant$	1
r_2	1		1			$\leqslant$	1
r_3		1	1			$\leqslant$	1
r_4	1				1	$\leqslant$	1
r_5			1	1		$\leqslant$	1
r_6				1	1	$\leqslant$	1

Corresponding conflict graph.



Given a column $p \in P$, we need to have $\sum_{i\in S} \alpha_{ip} \geqslant k$ to get a nonzero coefficient of λ_p in (21). For the master problem of VRPTW, the coefficient matrix can be translated as $\alpha_{ip} = \sum_{(i,j)\in\delta^+(i)} \alpha_{ijp}$; i.e., α_{ip} is the sum of all the outgoing edges of a customer i. Hence,

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

which is only one or larger when k or more customers of S are visited on route p.

Proposition 1. *The SR inequalities* (21) *are valid for the set-packing structure X.*

Proof. The proof follows directly from Chavtal-Gomory's procedure to construct valid inequalities (Wolsey 1998). Scale the $|S|$ inequalities $\sum_{p\in P}\alpha_{ip}\lambda_p \leqslant 1$ for each row $i \in S \subseteq M$ from (20) with $1/k \geqslant 0$ and add them:

$$\sum_{p\in P}\frac{1}{k}\sum_{i\in S}\alpha_{ip}\lambda_p \leqslant \frac{|S|}{k}.$$

Flooring on the left side and the right side leads to (21). □

Observe that when the coefficient $\lfloor (1/K)\sum_{i\in S}\alpha_{ip}\rfloor$ evaluates to zero or one for all $p \in P$ and the right side $\lfloor |S|/k\rfloor = 1$, then the set of SR inequalities (21) is a subset of the clique inequalities (18).

From Definition 1, it is clear that the SR inequalities are Chvatal-Gomory rank-1 cuts; see Chvatal (1973). Eisenbrand (1999) has shown that the separation problem is $\mathcal{NP}$-complete for general Chvatal-Gomory rank-1 cuts. However, in some special cases polynomial time separation is possible, e.g., the maximally violated mod-k cuts for a fixed k by Caprara et al. (2000). Because the SR inequalities are another special case, the separation problem will be investigated further.

3.1. Separation of Subset-Row Inequalities

The separation problem of SR inequalities is defined as follows: Given the current LP-solution λ where $\lambda_p < 1$ for all $p \in P$, let n be the size of S. For some fixed values n and k where $1 < k \leqslant n$, find the most violated SR inequality. Using the binary variable x_i to denote whether $i \in S$, this can be stated as

$$\max \quad \sum_{p\in P}\left\lfloor \frac{1}{k}\sum_{i\in M} a_{ip}x_i\right\rfloor \lambda_p - \left\lfloor \frac{n}{k}\right\rfloor \tag{22}$$

$$\text{s.t.} \quad \sum_{i\in M} x_i = n, \tag{23}$$

$$x_i \in \{0, 1\} \quad \forall\, i \in M. \tag{24}$$

The corresponding decision problem SR-DECISION asks whether

$$\sum_{p\in P}\left\lfloor \frac{1}{k}\sum_{i\in M} a_{ip}x_i\right\rfloor \lambda_p \geqslant c \tag{25}$$

is feasible subject to (23) and (24), where $1 \leqslant c < n$ and $c \in \mathbb{Z}$. Because we can multiply (25) by any coefficient $1/\gamma > 0$, the coefficient bounds $\lambda_p < 1$ and $c < n$ can be softened to

$$\lambda_p < \frac{1}{\gamma}, \qquad c < \frac{n}{\gamma}. \tag{26}$$

This leads to the following proposition.

Proposition 2. *The separation problem SR-DECISION is $\mathcal{NP}$-complete.*

Proof. We will show the statement by reduction from 3-conjunctive normal form satisfiability (3CNF-SAT). Given an expression ϕ written in three-conjunctive normal form, the 3CNF-SAT problem asks whether there is an assignment of binary values to the variables such that ϕ evaluates to true. An expression is in three-conjunctive normal form when it consists of a collection of disjunctive clauses $C_1, \ldots, C_m$ of literals, where a literal is a variable x_i or a negated variable $\neg x_i$, and each clause contains exactly three literals.

Let $x_1, \ldots, x_n$ be the set of variables that occurs in the clause ϕ. We transform the 3CNF-SAT instance to a SR-DECISION instance by constructing a matrix $A = (a_{ij})$ with $2n + 3$ rows and $m + n + 1$ columns, i.e., $M = \{1, \ldots, 2n+3\}$ and $P = \{1, \ldots, m+n+1\}$.

The rows $1, \ldots, 2n$ of matrix A correspond to literals $x_1, \neg x_1, x_2, \neg x_2, \ldots, x_n, \neg x_n$, while columns $j = 1, \ldots, m$ correspond to clauses $C_1, \ldots, C_m$, and columns $j = m+1, \ldots, m+n$ correspond to variables $x_1, \ldots, x_n$.

We now define matrix A as follows: For $j = 1, \ldots, m$, let $a_{ij} = 1$ iff the corresponding literal appears in clause C_j.

For $j = 1, \ldots, n$, let $a_{i,\,j+m} = 1$ iff the corresponding literal is x_j or $\neg x_j$. For $j = m+n+1$, let $a_{ij} = 0$. The last three rows of A are defined as follows: For $j = 1, \ldots, m+n$, let $a_{2n+1,\,j} = 0$, while $a_{2n+1,\,m+n+1} = 1$. For $j = 1, \ldots, m+n+1$, let $a_{2n+2,\,j} = a_{2n+3,\,j} = 1$. Finally, we set $k = 3$, $\lambda_p = 1$ for all $p \in P$ and $c = m+n+1$. Note that all coefficients are within the bounds (26) for γ sufficiently large. An example of the transformation is illustrated in Example 2.

Example 2. Illustration of the transformation 3CNF-SAT to SR-DECISION. Given the 3CNF-SAT expression

$$\phi = (x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4),$$

the matrix $A = (a_{ij})$ becomes

		1	$\cdots$	m	$m+1$	$\cdots$	$\cdots$	$m+n$	$m+n+1$
		C_1	$\cdots$	C_m	x_1	$\cdots$	$\cdots$	x_n	
1	x_1	1			1				
2	$\neg x_1$	1		1	1				
	x_2		1			1			
$\vdots$	$\neg x_2$	1				1			
	x_3		1				1		
	$\neg x_3$			1			1		
	x_4		1					1	
$2n$	$\neg x_4$			1				1	
$2n+1$									1
$2n+2$		1	1	1	1	1	1	1	1
$2n+3$		1	1	1	1	1	1	1	1

while we set $k = 3$, $\lambda_p = 1$ for $p \in P$ and $c = 8$.

With the chosen constants, the SR-DECISION problem (25) reads

$$\sum_{p \in P} \left\lfloor \frac{1}{3} \sum_{i \in M} a_{ip} x_i \right\rfloor \geqslant m+n+1 = |P|,$$

which is satisfied if and only if

$$\sum_{i \in M} a_{ip} x_i \geqslant 3 \quad \forall\, p \in P.$$

Because the last three rows of A always must be chosen, it is equivalent to

$$\sum_{i=1}^{2n} a_{ip} x_i \geqslant 1 \quad \forall\, p = 1, \ldots, m+n.$$

(i) Assume that there is a feasible assignment of binary values to $x_1, \ldots, x_n$ such that ϕ evaluates to true in the 3CNF-SAT instance. In the corresponding SR-DECISION problem, choose row i if and only if the corresponding literal is true in ϕ. Because exactly n literals are true, we will in this way choose n rows. Because at least one literal is true in each clause and each column $1, \ldots, m$ corresponds to a clause in A, we will get a contribution of at least one in each of these columns. Moreover, because exactly one of x_i and $\neg x_i$ is true in ϕ, we will get a contribution of exactly one in column $m+1, \ldots, m+n$. Hence, the corresponding SR-DECISION problem is true.

(ii) Assume, on the other hand, that SR-DECISION is true. Let $P' \subseteq P$ be the set of rows corresponding to the solution. By assumption, $|P'| = n$. First, we notice that exactly one of the rows corresponding to the literals x_i and $\neg x_i$ is chosen. This follows from the fact that we have n columns $m+1, \ldots, m+n$, which needs to be covered by n rows, and each row covers exactly one column. For each literal in ϕ, let x_i or $\neg x_i$ be true if the corresponding row was chosen in SR-DECISION. Each variable will be well defined due to the above argument. Moreover, because the rows P' must cover at least one $a_{pi} = 1$ for each column $j = 1, \ldots, m$, we see that each clause in ϕ becomes true.

Because the reduction is polynomial, and SR-DECISION obviously is in $\mathcal{NP}$, we have proved the statement. $\square$

Example 3 shows that typical separation problems of SR inequalities actually possess the properties assumed in the $\mathcal{NP}$-completeness proof.

Example 3. To illustrate that the bounds (26) indeed are realistic, consider the case $k = 3$. Choose $\gamma = (m+n+1)/\beta$ where $\beta = (n-2)/3$ or $\beta = (n-1)/3$, depending on which of the expressions that evaluates to an integral value. The right side of (25) evaluates to

$$c \cdot \frac{1}{\gamma} = (m+n+1) \cdot \frac{\beta}{m+n+1} = \beta,$$

where an integral value of β gives

$$\beta = \left\lfloor \frac{n}{3} \right\rfloor < n.$$

The value of λ gives

$$\lambda_p \cdot \frac{1}{\gamma} = 1 \cdot \frac{\beta}{m+n+1} \leqslant 1 \quad \forall\, p \in P.$$

Hence, all bounds are valid according to the separation problem (22)–(24).

4. Label-Setting Algorithm

When solving the pricing problem, it is noted that finding a route with negative reduced cost corresponds to finding a negative cost path starting and ending at the depot, i.e., an ESPPRC. Our ESPPRC algorithm is based on standard label-setting techniques presented by, e.g., Beasley and Christofides (1989), Dumitrescu (2002), Feillet et al. (2004), Chabrier (2005), Danna and Le Pape (2005); hence, in the following we focus mainly on the dominance criteria used for handling the modifications stemming from the SR inequalities of the master problem.

The ESPPRC can be formally defined as: Given a weighted directed graph $G(V, E)$ with nodes V and edges E, and a set of resources R. For each edge $(i, j) \in E$ and resource $r \in R$, three parameters are given: A lower limit $a_r(i, j)$ on the accumulation of resource r when traversing edge $(i, j) \in E$; an upper limit $b_r(i, j)$ on the accumulation of resource r when traversing edge $(i,j)\in E$; and finally, an amount $c_r(i,j)$ of resource r consumed by traversing edge $(i,j)\in E$. The objective is to find a minimum cost path p from a source node $o\in V$ to a target node $o'\in V$, where the accumulated resources of p satisfy the limits for all resources $r\in R$. Without loss of generality, we assume that the limits must be satisfied at the start of each edge (i,j), i.e., before $c_r(i,j)$ has been consumed.

Remark that equivalent upper and lower limits and consumptions on the nodes can be "pushed" onto the edges, e.g., the ingoing edges of the node.

For the pricing problem of VRPTW, the resources are demand d, time t, a binary visit-counter for each customer $v\in C$, and reduced cost $\bar{c}$. Note that also the reduced cost is considered a resource. When considering the pricing problem of VRPTW, the consumptions and upper and lower limits of the resources at each edge (i,j) in ESPPRC are

$$a_d(i,j)=0,\quad b_d(i,j)=D-d_j,\quad c_d(i,j)=d_j\quad \forall (i,j)\in E,$$

$$a_t(i,j)=a_i,\quad b_t(i,j)=b_i,\quad c_t(i,j)=\tau_{ij}\quad \forall (i,j)\in E,$$

$$a_v(i,j)=0,\quad b_v(i,j)=1,\quad c_v(i,j)=1\qquad \forall v\in V\colon v=j,\ \forall (i,j)\in E,$$

$$a_v(i,j)=0,\quad b_v(i,j)=1,\quad c_v(i,j)=0\qquad \forall v\in V\colon v\neq j,\ \forall (i,j)\in E,$$

$$a_{\bar{c}}(i,j)=-\infty,\quad b_{\bar{c}}(i,j)=\infty,\quad c_{\bar{c}}(i,j)=\bar{c}_{ij}\quad \forall (i,j)\in E.$$

In the label-setting algorithm, labels at node v represent partial paths from o to v. The following attributes for a label L are considered:

- $\bar{v}(L)$ current end-node of the partial path represented by L.
- $\bar{c}(L)$ sum of the reduced cost along path L.
- $r(L)$ accumulated consumption of resource $r\in R$ along path L.

A feasible extension $\epsilon\in\mathscr{E}(L)$ of a label L is a partial path starting in a node $\bar{v}(L)\in V$ and ending in the target node o', that does not violate any resources when concatenated with the partial path represented by L.

In the following, it is assumed that all resources are bounded strongly from above, and weakly from below. This means that if the current resource accumulation of a label is below the lower limit on a given edge, it is allowed to fill up the resource to the lower limit, e.g., waiting for a time window to open. This means that two consecutive labels L_u and L_v that are related by an edge (u,v), i.e., L_u is extended and creates L_v, where $\bar{v}(L_u)=u$ and $\bar{v}(L_v)=v$, must satisfy

$$r(L_v)\leqslant b_r(u,v)\quad \forall r\in R, \tag{27}$$

$$r(L_v)=\max\{r(L_u)+c_r(u,v),\,a_r(u,v)\}\quad \forall r\in R. \tag{28}$$

Here (27) demands that each label L_u satisfies the upper limit $b_r(u,v)$ of resource r corresponding to edge (u,v), while (28) states that resource r at label L_v corresponds to the resource consumption at label L_u plus the amount consumed by traversing edge (u,v), respecting the lower limit $a_r(u,v)$ on edge (u,v).

A simple enumeration algorithm could be used to produce all these labels, but to limit the number of labels considered, dominance rules are introduced to fathom labels that do not lead to an optimal solution.

Definition 2. A label L_i dominates label L_j if

$$\bar{v}(L_i)=\bar{v}(L_j), \tag{29}$$

$$\bar{c}(L_i)\leqslant\bar{c}(L_j), \tag{30}$$

$$\mathscr{E}(L_j)\subseteq\mathscr{E}(L_i). \tag{31}$$

In other words, the paths corresponding to labels L_i and L_j should end at the same node $\bar{v}(L_i)=\bar{v}(L_j)\in V$, the path corresponding to label L_i should cost no more than the path corresponding to label L_j, and any feasible extension of L_j is also a feasible extension of L_i.

Feillet et al. (2004) suggested to consider the set of nodes that cannot be reached from a label L_i and compare the set with the unreachable nodes of a label L_j to determine if some extensions are impossible. In other words, update the node resources in an eager fashion instead of a lazy one. The following definition is a generalization of Feillet et al. (2004, Definition 3).

Definition 3. Given a start node $o\in V$, a label L, and a node $u\in V$ where $\bar{v}(L)=u$, a node $v\in V$ is considered *unreachable* if v has already been visited on the path from o to u or if a resource window is violated, e.g.,

$$\exists r\in R\quad r(L)+l_r(u,v)>b_r(v),$$

where $l_r(u,v)$ is a lower bound on the consumption of resource r on all feasible paths from u to v. The *node resources* are then given as $v(L)=1$ indicates that node $v\in V$ is unreachable from node $\bar{v}(L)\in V$, and $v(L)=0$ otherwise.

Determining if (31) holds can be quite cumbersome because the straightforward definition demands that we calculate all extensions of the two labels. Therefore, a sufficient criterion for (31) is sought that can be computed faster. If label L_i has consumed fewer resources than label L_j, then no resources are limiting the possibilities of extending L_i compared to L_j; hence, the following proposition can be used as a relaxed version of the dominance criteria.

Proposition 3 (Desaulniers et al. 1998). *If all resource extension functions are nondecreasing, then label L_i dominates label L_j if*

$$\bar{v}(L_i)=\bar{v}(L_j), \tag{32}$$

$$\bar{c}(L_i)\leqslant\bar{c}(L_j), \tag{33}$$

$$r(L_i)\leqslant r(L_j)\quad \forall r\in R. \tag{34}$$

Using Proposition 3 as a dominance criterion is a relaxation of the dominance criteria of Definition 2 because only a subset of labels satisfying (29), (30), and (31) satisfies inequalities (32), (33), and (34).

4.1. Solving the Modified Pricing Problem

Consider some valid SR inequality of the form (21),

$$\sum_{p\in P}\left\lfloor \frac{1}{k}\sum_{i\in S}\alpha_{ip}\right\rfloor \lambda_p \leqslant \left\lfloor \frac{|S|}{k}\right\rfloor,$$

where $S\subseteq M$ and $0<k\leqslant |S|$. Let $\sigma\leqslant 0$ be the corresponding dual variable when solving the master problem to LP-optimality. From (17), the reduced cost of a column in the VRPTW master problem is

$$\begin{aligned}\hat{c}_p &= \bar{c}_p - \sigma\left\lfloor \frac{\sum_{i\in S}\sum_{(i,j)\in\delta^+(i)}\alpha_{ijp}}{k}\right\rfloor\\ &= \sum_{(i,j)\in E}\bar{c}_{ij}\alpha_{ijp} - \sigma\left\lfloor \frac{\sum_{i\in S}\sum_{(i,j)\in\delta^+(i)}\alpha_{ijp}}{k}\right\rfloor. \end{aligned}\tag{35}$$

We analyze how this additional cost can be handled in the label-setting algorithm for ESPPRC.

Let $V(L)$ be the nodes visited on the partial path of label L. The cost of a label L can then be expressed as

$$\hat{c}(L)=\bar{c}(L)-\sigma\left\lfloor \frac{|S\cap V(L)|}{k}\right\rfloor. \tag{36}$$

A new resource m can be used to compute the coefficient of penalty σ for label L, i.e., $m(L)=|S\cap V(L)|$, the number of customers involved in the cut. Note that the consumption of resource m is one for each outgoing edge of the involved customers. Therefore, the usual dominance criteria of Proposition 3 can be used. Note that in case L_i dominates L_j, $\bar{c}(L_i)\leqslant \bar{c}(L_j)$ and $m(L_i)\leqslant m(L_j)$, so $\hat{c}(L_i)\leqslant \hat{c}(L_j)$ because $-\sigma>0$. Hence, the penalty term must be considered only on the last edge to the target node to compute the reduced cost $\hat{c}(L)$ of path L. However, further labels can be eliminated by exploiting the structure of (36).

For a label L, let

$$\mathcal{T}(L)=|S\cap V(L)| \bmod k$$

be the number of visits made to S since the last penalty was paid for visiting k nodes in S. Recall $\mathscr{E}(L)$ as the set of feasible extensions from the label L to the target node o' and note that when label L_i dominates label L_j, their common extensions are $\mathscr{E}(L_j)$ due to (31). The following cost dominance criterion is obtained for a single SR inequality.

Proposition 4. *If $\mathcal{T}(L_i)\leqslant\mathcal{T}(L_j)$, $\bar{v}(L_i)=\bar{v}(L_j)$, $\hat{c}(L_i)\leqslant \hat{c}(L_j)$, and $r(L_i)\leqslant r(L_j)$ $\forall\, r\in R$, then label L_i dominates label L_j.*

Proof. Consider any common extension $\epsilon\in\mathscr{E}(L_j)$. Because $\mathcal{T}(L_i)\leqslant\mathcal{T}(L_j)$, the relation between the number of future penalties for the two labels when concatenated with ϵ is

$$\left\lfloor \frac{|S\cap\epsilon|+\mathcal{T}(L_i)}{k}\right\rfloor \leqslant \left\lfloor \frac{|S\cap\epsilon|+\mathcal{T}(L_j)}{k}\right\rfloor.$$

This leads to the following relation between the costs:

$$\begin{aligned}\hat{c}(L_i+\epsilon)&=\hat{c}(L_i)+\bar{c}(\epsilon)-\sigma\left\lfloor \frac{|S\cap\epsilon|+\mathcal{T}(L_i)}{k}\right\rfloor\\ &\leqslant \hat{c}(L_j+\epsilon)=\hat{c}(L_j)+\bar{c}(\epsilon)-\sigma\left\lfloor \frac{|S\cap\epsilon|+\mathcal{T}(L_j)}{k}\right\rfloor.\end{aligned}$$

Hence, label L_i dominates label L_j. □

Proposition 5. *If $\mathcal{T}(L_i)>\mathcal{T}(L_j)$, $\bar{v}(L_i)=\bar{v}(L_j)$, $\hat{c}(L_i)-\sigma\leqslant \hat{c}(L_j)$, and $r(L_i)\leqslant r(L_j)$ $\forall\, r\in R$, then label L_i dominates label L_j.*

Proof. Consider any common extension $\epsilon\in\mathscr{E}(L_j)$. Because $\mathcal{T}(L_i)>\mathcal{T}(L_j)$, the relation between the number of future penalties for the two labels when concatenated with ϵ is

$$\left\lfloor \frac{|S\cap\epsilon|+\mathcal{T}(L_i)}{k}\right\rfloor \geqslant \left\lfloor \frac{|S\cap\epsilon|+\mathcal{T}(L_j)}{k}\right\rfloor. \tag{37}$$

Because $0\leqslant\mathcal{T}(L_j)<\mathcal{T}(L_i)\leqslant k$, it is clear that the left side of (37) is at most one unit larger than the right side, i.e., label L_i will pay the penalty at most one more time than label L_j. Hence,

$$\left\lfloor \frac{|S\cap\epsilon|+\mathcal{T}(L_i)}{k}\right\rfloor -1\leqslant \left\lfloor \frac{|S\cap\epsilon|+\mathcal{T}(L_j)}{k}\right\rfloor.$$

That is, the additional cost of extending L_i with ϵ is at most $-\sigma$ more than extending L_j with ϵ. This leads to the following relation between the costs:

$$\begin{aligned}\hat{c}(L_i+\epsilon)&=\hat{c}(L_i)+\bar{c}(\epsilon)-\sigma\left\lfloor \frac{|S\cap\epsilon|+\mathcal{T}(L_i)}{k}\right\rfloor\\ &=\hat{c}(L_i)-\sigma+\bar{c}(\epsilon)-\sigma\left(\left\lfloor \frac{|S\cap\epsilon|+\mathcal{T}(L_i)}{k}\right\rfloor-1\right)\\ &\leqslant \hat{c}(L_j)+\bar{c}(\epsilon)-\sigma\left\lfloor \frac{|S\cap\epsilon|+\mathcal{T}(L_j)}{k}\right\rfloor\\ &=\hat{c}(L_j+\epsilon).\end{aligned}$$

Hence, label L_i dominates label L_j. □

Observe that if $\mathcal{T}(L_i)+|S\cap\epsilon|<k$ for all $\epsilon\in\mathscr{E}(L_j)$, it is not possible to visit S enough times to trigger a penalty, i.e., the temporary penalty to the cost of L_i can be disregarded.

In the case of several SR inequalities, the new dominance criterion is as follows.

Proposition 6. *Let $Q=\{q\colon\ \sigma_q<0\wedge\mathcal{T}_q(L_i)>\mathcal{T}_q(L_j)\}$. Then, label L_i dominates label L_j if*

$$\bar{v}(L_i)=\bar{v}(L_j), \tag{38}$$

$$\hat{c}(L_i)-\sum_{q\in Q}\sigma_q\leqslant \hat{c}(L_j), \tag{39}$$

$$r(L_i)\leqslant r(L_j)\quad \forall\, r\in R. \tag{40}$$

PROOF. The validity of (39) follows directly from Propositions 4 and 5. The validity of (38) and (40) follows from Proposition 3. □

5. Computational Results

The BCP algorithm has been implemented using the BCP framework and the open-source linear programming solver CLP, both parts of the framework COIN (2005). All tests are run on an Intel® Pentium® 4 3.0-GHz PC with 4 GB of memory.

The benchmarks of Solomon (1987) follow a naming convention of `DTm.n`. The distribution `D` can be R, C, and RC, where the C instances have a clustered distribution of customers, the R instances have a random distribution of customers, and the RC instances are a mix of clustered and randomly distributed customers. The time window `T` is either 1 or 2, where instances of type 1 have tighter time windows than instances of type 2. The instance number is given by `m` and the number of customers is given by `n`.

The outline of the BCP algorithm presented in this paper is as follows.

Step 1. Choose an unprocessed branch node. If the lower bound is above the upper bound, then fathom the branch node.

Step 2. Solve the LP master problem.

Step 3. Solve the pricing problem heuristically. If columns with negative reduced cost have been found, then add them to the master problem and go back to Step 2.

Step 4. Solve the pricing problem to optimality. Update the lower bound. If the lower bound is above the upper bound, then fathom the branch node. If some new columns have been found, then add them to the master problem and go to Step 2.

Step 5. Separate the SR inequalities. If any violated cuts are found, then add them to the master problem and go to Step 2.

Step 6. If the LP solution is fractional, then branch and add the children to the set of unprocessed branch nodes. Mark the current node as processed and go to Step 1.

We allow a maximum of 400 variables and 50 cuts to be generated in each of Steps 3, 4, and 5. The pricing-problem heuristic is based on the label-setting algorithm, but a simpler heuristic dominance criterion is used. If a label L_i dominates L_j on *cost*, *demand*, and *time*, it is regarded as dominated and L_j is discarded. That is, no concern is taken to the node resources. The separation of SR inequalities is done with a complete enumeration of all inequalities with $|S| = 3$ and $k = 2$. Let B be the set of basic variables in the current LP solution and C be the set of customers; then the separation can be done in $O(|C|^3|B|)$. Preliminary tests showed that SR inequalities with different values of n and k seldom appeared in the VRPTW instances, hence no separation of these inequalities was done.

The branch tree is explored with a best-bound search strategy, i.e., the node with the lowest lower bound is chosen first, breaking ties based on the LP result of the strong branching. We have adapted the branching rule used by Fukasawa et al. (2006): For a subset of customers $S \subset C$, the number of vehicles to visit that set is either two or greater than or equal to four, i.e.,

$$\sum_{k\in K} \sum_{(i,\,j)\in\delta^+(S)} (x_{ijk} + x_{jik}) = 2$$

and

$$\sum_{k\in K} \sum_{(i,\,j)\in\delta^+(S)} (x_{ijk} + x_{jik}) \geqslant 4.$$

We are using the cut library of Lysgaard (2003) to separate candidate sets for branching, which is an implementation of the heuristic methods described in Lysgaard et al. (2004).

5.1. Running Times

To give a fair comparison between running times of our algorithm and the two most recent algorithms presented by Irnich and Villeneuve (2006) and Chabrier (2005), the CPU speed is taken into account. This is done according to the CPU2000 benchmarks reported by the Standard Performance Evaluation Corporation (SPEC 2005). Table 1 gives the integer and floating point benchmark scores and a normalized value; e.g., our computations were carried out on a computer approximately twice as fast as that of Chabrier (2005).

A comparison of running times is shown in Table 2. To save space we report results only on what we consider hard instances, e.g., the Solomon instances that were closed by either Irnich and Villeneuve (2006) or Chabrier (2005) and by us.

Our algorithm outperforms those of Irnich and Villeneuve (2006) and Chabrier (2005) for 17 out of 22 instances. Seven of these instances were solved without any SR inequalities. In these cases, the faster running times were probably due to the bidirectional label-setting algorithm.

With the introduction of SR inequalities, our algorithm becomes competitive with the algorithm based on solving k-cyc-SPPRC (e.g., instances R104.100, RC104.100, RC107.100, RC108.100, and R211.50) and clearly outperforms the ESPPRC-based algorithm on the harder instances

Table 1. Comparison of computer speed.

Author(s)	CPU	SpecINT	SpecCFP	Normalized
Irnich and Villeneuve (2006)	P3 600 MHz*	295	204	0.23
Chabrier (2005)	P4 1.5 GHz	526	606	0.52
Jepsen et al. (this paper)	P4 3.0 GHz	1,099	1,077	1.00

Notes. Based on CPU2000 benchmarks from SPEC (2005).

*Benchmarks are given for P3 650 MHz because no benchmarks were available for P3 600. The normalized value is an average of SpecINT and SpecCFP.

Table 2. Comparison of running times.

Instance	Irnich and Villeneuve (2006) Time (s)	Chabrier (2005) Time (s)	Jepsen et al. (this paper) Time (s)	Speedup
R104.100	268,106.0	—	**32,343.9**	1.9/—
RC104.100	986,809.0	—	**65,806.8**	3.4/—
RC107.100	42,770.7	—	**153.8**	64.0/—
RC108.100	71,263.0	—	**3,365.0**	4.9/—
R203.50	**217.1**	3,320.9	50.8	1.0/34.0
R204.25	123.1	171.6	**7.5**	3.8/11.9
R205.50	585.7	531.0	**15.5**	8.6/17.8
R206.50	22,455.3	4,656.1	**190.9**	27.1/12.7
R208.25	321.9	741.5	****2.9**	25.5/133.0
R209.50	142.4	195.4	**16.6**	2.0/6.1
R210.50	11,551.4	65,638.6	****332.7**	8.0/102.6
R211.50	**21,323.0**	—	10,543.8	0.5/—
RC202.50	241.6	**13.0**	*10.7	5.2/0.6
RC202.100	124,018.0	19,636.5	**312.6**	91.2/32.7
RC203.25	1,876.0	5.1	****0.7**	616.4/3.8
RC203.50	54,229.2	4,481.5	****190.9**	65.3/12.2
RC204.25	—	13.0	****2.0**	—/3.4
RC205.50	52.6	**10.6**	*5.9	2.1/0.9
RC205.100	13,295.9	15,151.7	**221.2**	13.8/35.6
RC206.50	469.1	**9.4**	*8.2	13.2/0.6
RC207.50	—	71.1	****21.5**	—/1.7
RC208.25	—	33,785.3	**78.4**	—/224.1

Notes. Speedup is calculated based on the normalized values in Table 1 and are versus Irnich and Villeneuve (2006) and Chabrier (2005), respectively. Results with (*) are based on an algorithm without the SR inequalities. Results in bold indicate the fastest algorithm after normalization. (—) indicates that no running times were provided by the author(s) or that the instance was not solved.

(e.g., instances R210.50, RC202.100, RC205.100, and RC208.25). In some cases, when solving the C1 and C2 instances, the BCP algorithm tails off, leading to slow solution times or no solution at all. However, this must be seen in light of a simple implementation and with no use of other cutting planes than the SR inequalities.

5.2. Comparing Lower Bounds in the Root Node

Table 3 reports the lower bounds obtained in the root node of the master problem with and without SR inequalities and with best bounds obtained by Irnich and Villeneuve (2006) using k-cyc-SPPRC. Again, we report results only on what we consider the hard instances from Table 2 plus the instances closed by us.

As seen, the lower bounds obtained with SR inequalities are improved quite significantly for most of the instances. Moreover, in most cases the problems are solved without branching. Out of the 32 instances considered, the gap was closed in the root node in eight instances due to the ESPPRC and in an additional 16 instances due to the SR inequalities. However, one needs to take into account that the running time of solving the root node is increased due to the increased difficulty of the pricing problems.

Table 3. Comparison of root lower bounds.

		Irnich and Villeneuve (2006)		Jepsen et al. (this paper)	
Instance	UB	k	LB	LB(1)	LB(2)
R104.100	971.5	3	955.8	956.9	971.3
R108.100	932.1	4	913.9	913.6	**932.1**
R112.100	948.6	3	925.9	926.8	946.7
RC104.100	1,132.3	3	1,114.4	1,101.9	1,129.9
RC106.100	1,372.7	4	1,343.1	1,318.8	1,367.3
RC107.100	1,207.8	4	1,195.4	1,183.4	**1,207.8**
RC108.100	1,114.2	3	1,100.5	1,073.5	**1,114.2**
R202.100	1,029.6	0	933.5	1,022.3	1,027.3
R203.50	605.3	4	598.6	598.6	**605.3**
R203.100	870.8	2	847.1	867.0	**870.8**
R204.25	355.0	4	349.1	350.5	**355.0**
R205.50	690.1	4	682.8	682.9	**690.1**
R206.50	632.4	4	621.3	626.4	**632.4**
R207.50	575.5	4	557.4	564.1	**575.5**
R208.25	328.2	4	327.1	**328.2**	**328.2**
R209.50	600.6	4	599.9	599.9	**600.6**
R209.100	854.8	3	834.4	841.5	854.4
R210.50	645.6	4	633.1	636.1	645.3
R211.50	535.5	4	526.0	528.7	**535.5**
RC202.50	613.6	4	604.5	**613.6**	**613.6**
RC202.100	1,092.3	3	1,055.0	1,088.1	**1,092.3**
RC203.25	326.9	4	297.7	**326.9**	**326.9**
RC203.50	555.3	4	530.0	**555.3**	**555.3**
RC203.100	923.7	0	693.7	922.6	**923.7**
RC204.25	299.7	4	266.3	**299.7**	**299.7**
RC205.50	630.2	4	**630.2**	**630.2**	**630.2**
RC205.100	1,154.0	3	1,130.5	1,147.7	**1,154.0**
RC206.50	610.0	4	597.1	**610.0**	**610.0**
RC206.100	1,051.1	3	1,017.0	1,038.6	**1,051.1**
RC207.50	558.6	4	504.9	**558.6**	**558.6**
RC208.25	269.1	4	238.3	**269.1**	**269.1**
RC208.50	476.7	3	422.3	472.3	**476.7**

Notes. LB by Irnich and Villeneuve (2006) is the best lower bound obtained with k-cyc-SPPRC and valid inequalities, LB(1) is with ESPPRC, and LB(2) is with ESPPRC and SR inequalities. Lower bounds in bold indicate lower bounds equal to the upper bound. Instances in bold are the Solomon instances closed by us.

5.3. Closed Solomon Instances

Table 4 gives an overview of how many instances were solved for each class of the Solomon instances. We were able to close eight previously unsolved instances. We did not succeed in solving four previously solved instances (R204.50, C204.50, C204.100, and RC204.50).

Information on all solved Solomon instances can be found in Tables A.1–A.3 in Appendix A. Furthermore, Table 5 provides detailed information of the instances closed in this paper. The solutions can be found in Tables B.1–B.8 in Appendix B.

6. Concluding Remarks

The introduction of the SR inequalities significantly improved the results of the BCP algorithm. This made it

Table 4. Summary of solved Solomon instances.

		25 customers		50 customers		100 customers	
Class	No.	Prev.	Jepsen et al. (this paper)	Prev.	Jepsen et al. (this paper)	Prev	Jepsen et al. (this paper)
R1	12	12	12	12	12	10	12
C1	9	9	9	9	9	9	9
RC1	8	8	8	8	8	8	8
R2	11	11	11	9	9	1	4
C2	8	8	8	8	7	8	7
RC2	8	8	8	8	7	3	5
Summary	56	56	56	55	52	39	45

Notes. No. is the number of instances in that class, and for 25, 50, and 100 customers the two columns refer to the number of instances previously solved to optimality and the number of instances solved to optimality by us.

possible to solve eight previously unsolved instances from the Solomon benchmarks.

Except for four cases (R204.50, C204.50, and C204.100 solved with k-cyc-SPPRC by Irnich and Villeneuve 2006, and RC204.50 solved by Danna and Le Pape 2005), our BCP algorithm is competitive and in most cases superior to earlier algorithms within this field. With minor modifications in the definition of the conflict graph, the SR inequalities can be applied to the k-cyc-SPPRC algorithm using the same cost-modified dominance criteria as described in this paper. Preliminary results by Jepsen et al. (2005) have shown that the lower bounds obtained in a BCP algorithm for VRPTW using the k-cyc-SPPRC algorithm and SR inequalities are almost as good as those obtained using the approach presented in this paper. This seems to be a promising direction of research to solve large VRPTW instances because the ESPPRC algorithm is considerably slower than the k-cyc-SPPRC algorithm when the number of customers increases.

Moreover, we note that the SR inequalities can be applied to any set-packing problem. That is, they can be used in BCP algorithms for other problems with a set-packing problem master problem. One needs only to consider how the dual variables of the SR inequalities are handled in the pricing problems; however, this is not necessarily trivial and must be investigated for the individual pricing problems.

Adding SR inequalities to the master problem means that the pricing problem becomes a shortest-path problem with nonadditive nondecreasing constraints or objective function. By modifying the dominance criteria, we have shown that this is tractable in a label-setting algorithm. A further discussion of shortest-path problems with various nonadditive constraints can be found in Pisinger and Reinhardt (2006). The development of algorithms that efficiently handle nonadditive constraints is important to increase the number of valid inequalities that can be handled.

Appendix A. Results on Solomon Instances

This appendix contains detailed information about solved Solomon instances. The first column of the tables is the instance name, then three columns for the branch-and-cut-and-price algorithm with ESPPRC, and with ESPPRC and SR inequalities follow. The columns are the lower bound in the root node, the number of branch tree nodes, and the total running time. A (—) means that the instance was not solved. The last two columns are the optimal upper bound and a reference to the authors who were the first to solve that instance—disregarding Desrochers et al. (1992), who solved many of the instances with a different calculation of the travel times, making it hard to compare with later solutions. The author legend is:

C: Chabrier (2005)
CR: Cook and Rich (1999)
DLP: Danna and Le Pape (2005)
IV: Irnich and Villeneuve (2006)
JPSP: Jepsen et al. (this paper)
KDMSS: Kohl et al. (1999)
KLM: Kallehauge et al. (2000)
L: Larsen (1999)
S: Salani (2005)

Table 5. Instances closed by Jepsen et al. (this paper).

Instance	UB	LB	Vehicles	Tree	LP	$\text{Time}_{\text{root}}$ (s)	Time_{var} (s)	Time_{LP} (s)	Time (s)
R108.100	932.1	932.1	10	1	132	5,911.71	5,796.04	77.36	5,911.74
R112.100	948.6	946.7	10	9	351	55,573.68	199,907.03	1,598.63	202,803.94
R202.100	1,029.6	1,027.3	8	13	514	974.51	730.04	4,810.47	8,282.38
R203.100	870.8	870.8	6	1	447	54,187.15	48,474.45	3,973.42	54,187.40
R207.50	575.5	575.5	3	1	107	34,406.92	34,282.47	118.69	34,406.96
R209.100	854.8	854.4	5	3	337	31,547.45	74,779.58	2,978.42	78,560.47
RC203.100	923.7	923.7	5	1	402	14,917.18	13,873.53	1,025.65	14,917.36
RC206.100	1,051.1	1,051.1	7	1	179	339.63	159.33	171.34	339.69

Notes. UB is the optimal solution found by us, LB is the lower bound at the root node, Vehicles is the number of vehicles in the solution, Tree is the number of branch nodes, LP is the number of LP iterations, $\text{Time}_{\text{root}}$ is the time solving the root node, Time_{var} is the time spent solving the pricing problem, Time_{LP} is the time spent solving LP problems, and Time is the total time.

Table A.1. Instances with 25 customers.

	With ESPPRC			With ESPPRC and SR				
Instance	LB	Tree	Time (s)	LB	Tree	Time (s)	UB	Ref.
R101	617.1	1	0.02	617.1	1	0.02	617.1	KDMSS
R102	546.4	3	0.13	547.1	1	0.09	547.1	KDMSS
R103	454.6	1	0.11	454.6	1	0.11	454.6	KDMSS
R104	416.9	1	0.12	416.9	1	0.12	416.9	KDMSS
R105	530.5	1	0.02	530.5	1	0.02	530.5	KDMSS
R106	457.3	5	0.29	465.4	1	0.10	465.4	KDMSS
R107	424.3	1	0.12	424.3	1	0.12	424.3	KDMSS
R108	396.9	3	0.31	397.3	1	0.24	397.3	KDMSS
R109	441.3	1	0.06	441.3	1	0.06	441.3	KDMSS
R110	438.4	17	1.16	444.1	3	0.29	444.1	KDMSS
R111	427.3	3	0.23	428.8	1	0.13	428.8	KDMSS
R112	387.1	13	1.19	393.0	1	0.52	393.0	KDMSS
C101	191.3	1	0.13	191.3	1	0.13	191.3	KDMSS
C102	190.3	1	0.53	190.3	1	0.53	190.3	KDMSS
C103	190.3	1	0.80	190.3	1	0.80	190.3	KDMSS
C104	186.9	1	3.29	186.9	1	3.29	186.9	KDMSS
C105	191.3	1	0.17	191.3	1	0.17	191.3	KDMSS
C106	191.3	1	0.14	191.3	1	0.14	191.3	KDMSS
C107	191.3	1	0.20	191.3	1	0.20	191.3	KDMSS
C108	191.3	1	0.37	191.3	1	0.37	191.3	KDMSS
C109	191.3	1	0.62	191.3	1	0.62	191.3	KDMSS
RC101	406.7	5	0.20	461.1	1	0.09	461.1	KDMSS
RC102	351.8	1	0.05	351.8	1	0.05	351.8	KDMSS
RC103	332.8	1	0.19	332.8	1	0.19	332.8	KDMSS
RC104	306.6	1	0.52	306.6	1	0.52	306.6	KDMSS
RC105	411.3	1	0.06	411.3	1	0.06	411.3	KDMSS
RC106	345.5	1	0.10	345.5	1	0.10	345.5	KDMSS
RC107	298.3	1	0.29	298.3	1	0.29	298.3	KDMSS
RC108	294.5	1	0.67	294.5	1	0.67	294.5	KDMSS
R201	460.1	3	0.44	463.3	1	0.27	463.3	CR + KLM
R202	410.5	1	0.61	410.5	1	0.61	410.5	CR + KLM
R203	391.4	1	0.80	391.4	1	0.80	391.4	CR + KLM
R204	350.5	19	18.40	355.0	1	7.51	355.0	IV + C
R205	390.6	3	1.62	393.0	1	1.06	393.0	CR + KLM
R206	373.6	3	1.67	374.4	1	0.93	374.4	CR + KLM
R207	360.1	5	4.03	361.6	1	1.39	361.6	KLM
R208	328.2	1	2.87	328.2	1	2.87	328.2	IV + C
R209	364.1	9	4.99	370.7	1	2.26	370.7	KLM
R210	404.2	3	1.52	404.6	1	1.04	404.6	CR + KLM
R211	341.4	29	38.17	350.9	1	22.62	350.9	KLM
C201	214.7	1	0.84	214.7	1	0.84	214.7	CR + L
C202	214.7	1	3.00	214.7	1	3.00	214.7	CR + L
C203	214.7	1	3.02	214.7	1	3.02	214.7	CR + L
C204	213.1	1	7.00	213.1	1	7.00	213.1	CR + KLM
C205	214.7	1	1.10	214.7	1	1.10	214.7	CR + L
C206	214.7	1	1.75	214.7	1	1.75	214.7	CR + L
C207	214.5	1	2.70	214.5	1	2.70	214.5	CR + L
C208	214.5	1	1.85	214.5	1	1.85	214.5	CR + L
RC201	360.2	1	0.25	360.2	1	0.25	360.2	CR + L
RC202	338.0	1	0.58	338.0	1	0.58	338.0	CR + KLM
RC203	326.9	1	0.72	326.9	1	0.72	326.9	IV + C
RC204	299.7	1	1.95	299.7	1	1.95	299.7	C
RC205	338.0	1	0.62	338.0	1	0.62	338.0	L + KLM
RC206	324.0	1	0.87	324.0	1	0.87	324.0	KLM
RC207	298.3	1	0.88	298.3	1	0.88	298.3	KLM
RC208	269.1	1	78.42	269.1	1	78.42	269.1	C

Table A.2. Instances with 50 customers.

	With ESPPRC			With ESPPRC and SR				
Instance	LB	Tree	Time (s)	LB	Tree	Time (s)	UB	Ref.
R101	1,043.4	3	0.14	1,044.0	1	0.09	1,044.0	KDMSS
R102	909.0	1	0.27	909.0	1	0.27	909.0	KDMSS
R103	769.3	13	4.98	772.9	1	2.02	772.9	KDMSS
R104	619.1	21	33.29	625.4	1	6.73	625.4	KDMSS
R105	892.2	29	2.78	893.7	5	1.15	899.3	KDMSS
R106	791.4	5	1.41	793.0	1	0.83	793.0	KDMSS
R107	707.3	11	5.56	711.1	1	4.76	711.1	KDMSS
R108	594.7	789	1,723.29	607.4	23	1,601.68	617.7	CR + KLM
R109	775.4	77	20.11	783.3	7	11.54	786.8	KDMSS
R110	695.1	9	3.38	697.0	1	1.46	697.0	KDMSS
R111	696.3	41	19.21	707.2	1	3.67	707.2	CR + KLM
R112	614.9	165	169.26	630.2	1	35.67	630.2	CR + KLM
C101	362.4	1	0.47	362.4	1	0.47	362.4	KDMSS
C102	361.4	1	1.59	361.4	1	1.59	361.4	KDMSS
C103	361.4	1	6.06	361.4	1	6.06	361.4	KDMSS
C104	358.0	1	1,564.88	358.0	1	1,564.88	358.0	KDMSS
C105	362.4	1	0.49	362.4	1	0.49	362.4	KDMSS
C106	362.4	1	0.69	362.4	1	0.69	362.4	KDMSS
C107	362.4	1	0.97	362.4	1	0.97	362.4	KDMSS
C108	362.4	1	1.55	362.4	1	1.55	362.4	KDMSS
C109	362.4	1	3.62	362.4	1	3.62	362.4	KDMSS
RC101	850.1	39	5.60	944.0	1	2.12	944.0	KDMSS
RC102	721.9	127	60.41	822.5	1	8.68	822.5	KDMSS
RC103	645.3	9	8.56	710.9	1	40.05	710.9	KDMSS
RC104	545.8	1	5.71	545.8	1	5.71	545.8	KDMSS
RC105	761.6	21	7.22	855.3	1	4.31	855.3	KDMSS
RC106	664.5	11	3.35	723.2	1	3.88	723.2	KDMSS
RC107	603.6	7	4.60	642.7	1	4.49	642.7	KDMSS
RC108	541.2	5	15.88	594.8	5	260.95	598.1	KDMSS
R201	791.9	1	4.97	791.9	1	4.97	791.9	CR + KLM
R202	698.5	1	9.88	698.5	1	9.88	698.5	CR + KLM
R203	598.6	25	355.99	605.3	1	50.80	605.3	IV + C
R204			—			—	506.4	IV
R205	682.9	35	118.12	690.1	1	15.45	690.1	IV + C
R206	626.4	47	288.00	632.4	1	190.86	632.4	IV + C
R207	564.1	141	15,400.44	575.5	1	34,406.96	575.5	**JPSP**
R208			—			—	—	—
R209	599.9	3	24.45	600.6	1	16.63	600.6	IV + C
R210	636.1	49	332.70	645.3	3	18,545.61	645.6	IV + C
R211	528.7	31	44,644.89	535.5	1	10,543.81	535.5	IV + DLP
C201	360.2	1	42.07	360.2	1	42.07	360.2	CR + L
C202	360.2	1	67.05	360.2	1	67.05	360.2	CR + KLM
C203	359.8	1	214.88	359.8	1	214.88	359.8	CR + KLM
C204			—			—	350.1	KLM
C205	359.8	1	64.18	359.8	1	64.18	359.8	CR + KLM
C206	359.8	1	38.91	359.8	1	38.91	359.8	CR + KLM
C207	359.6	1	72.81	359.6	1	72.81	359.6	CR + KLM
C208	350.5	1	55.79	350.5	1	55.79	350.5	CR + KLM
RC201	684.8	1	3.00	684.8	1	3.00	684.8	L + KLM
RC202	613.6	1	10.69	613.6	1	10.69	613.6	IV + C
RC203	555.3	1	190.88	555.3	1	190.88	555.3	IV + C
RC204			—			—	442.2	DLP
RC205	630.2	1	5.88	630.2	1	5.88	630.2	IV + C
RC206	610.0	1	8.17	610.0	1	8.17	610.0	IV + C
RC207	558.6	1	21.53	558.6	1	21.53	558.6	C
RC208			—	476.7	1	1,639.40	476.7	S

Table A.3. Instances with 100 customers.

	With ESPPRC			With ESPPRC and SR				
Instance	LB	Tree	Time (s)	LB	Tree	Time (s)	UB	Ref.
R101	1,631.2	57	20.08	1,634.0	3	1.87	1,637.7	KDMSS
R102	1,466.6	1	4.39	1,466.6	1	4.39	1,466.6	KDMSS
R103	1,206.8	19	55.78	1,208.7	1	23.85	1,208.7	CR + L
R104			—	971.3	3	32,343.92	971.5	IV
R105	1,346.2	113	126.96	1,355.2	5	43.12	1,355.3	KDMSS
R106	1,227.0	147	511.07	1,234.6	1	75.42	1,234.6	CR + KLM
R107			—	1,064.3	3	1,310.30	1,064.6	CR + KLM
R108			—	932.1	1	5,911.74	932.1	**JPSP**
R109			—	1,144.1	19	1,432.41	1,146.9	CR + KLM
R110			—	1,068.0	3	1,068.31	1,068.0	CR + KLM
R111			—	1,045.9	39	83,931.48	1,048.7	CR + KLM
R112			—	946.7	9	202,803.94	948.6	**JPSP**
C101	827.3	1	3.02	827.3	1	3.02	827.3	KDMSS
C102	827.3	1	12.92	827.3	1	12.92	827.3	KDMSS
C103	826.3	1	33.89	826.3	1	33.89	826.3	KDMSS
C104	822.9	1	4,113.09	822.9	1	4,113.09	822.9	KDMSS
C105	827.3	1	5.34	827.3	1	5.34	827.3	KDMSS
C106	827.3	1	7.15	827.3	1	7.15	827.3	KDMSS
C107	827.3	1	6.55	827.3	1	6.55	827.3	KDMSS
C108	827.3	1	14.46	827.3	1	14.46	827.3	KDMSS
C109	827.3	1	20.53	827.3	1	20.53	827.3	KDMSS
RC101	1,584.1	59	56.62	1,619.8	1	12.39	1,619.8	KDMSS
RC102			—	1,457.4	1	76.69	1,457.4	CR + KLM
RC103			—	1,257.7	3	2,705.78	1,258.0	CR + KLM
RC104			—	1,129.9	7	65,806.79	1,132.3	IV
RC105	1,472.0	191	309.83	1,513.7	1	26.73	1,513.7	KDMSS
RC106			—	1,367.3	37	15,891.55	1,372.7	S
RC107			—	1,207.8	1	153.80	1,207.8	IV
RC108			—	1,114.2	1	3,365.00	1,114.2	IV
R201			—	1,143.2	1	139.03	1,143.2	KLM
R202			—	1,027.3	13	8,282.38	1,029.6	**JPSP**
R203			—	870.8	1	54,187.40	870.8	**JPSP**
R204			—			—	—	—
R205			—			—	—	—
R206			—			—	—	—
R207			—			—	—	—
R208			—			—	—	—
R209			—	854.8	3	78,560.47	854.8	**JPSP**
R210			—			—	—	—
R211			—			—	—	—
C201	589.1	1	203.34	589.1	1	203.34	589.1	CR + KLM
C202	589.1	1	3,483.15	589.1	1	3,483.15	589.1	CR + KLM
C203	588.7	1	13,070.71	588.7	1	13,070.71	588.7	KLM
C204			—			—	588.1	IV
C205	586.4	1	416.56	586.4	1	416.56	586.4	CR + KLM
C206	586.0	1	594.92	586.0	1	594.92	586.0	CR + KLM
C207	585.8	1	1,240.97	585.8	1	1,240.97	585.8	CR + KLM
C208	585.8	1	555.27	585.8	1	555.27	585.8	KLM
RC201			—	1,261.7	3	229.27	1,261.8	KLM
RC202			—	1,092.3	1	312.57	1,092.3	IV + C
RC203	922.6	11	34,063.95	923.7	1	14,917.36	923.7	**JPSP**
RC204			—			—	—	—
RC205			—	1,154.0	1	221.24	1,154.0	IV + C
RC206			—	1,051.1	1	339.69	1,051.1	**JPSP**
RC207			—			—	—	—
RC208			—			—	—	—

Appendix B. Solutions of Closed Solomon Instances

Table B.1. Solution of R108.100.

Cost	Route
8.8	53
119.2	70, 30, 20, 66, 65, 71, 35, 34, 78, 77, 28
105.4	92, 98, 91, 44, 14, 38, 86, 16, 61, 85,100, 37
84.1	2, 57, 15, 43, 42, 87, 97, 95, 94, 13, 58
106.5	73, 22, 41, 23, 67, 39, 56, 75, 74, 72, 21, 40
114.6	52, 88, 62, 19, 11, 64, 63, 90, 32, 10, 31
78.4	6, 96, 59, 99, 93, 5, 84, 17, 45, 83, 60, 89
107.3	26, 12, 80, 68, 29, 24, 55, 4, 25, 54
93.2	27, 69, 76, 3, 79, 9, 51, 81, 33, 50, 1
114.6	18, 7, 82, 8, 46, 36, 49, 47, 48
932.1	10

Notes. The left column is the cost of the routes and the total cost. The right column is a comma-separated list indicating the customers visited on the routes in the order of visit and the total number of routes.

Table B.2. Solution of R112.100.

Cost	Route
78.1	6, 94, 95, 87, 42, 43, 15, 57, 58
115.8	2, 41, 22, 75, 56, 23, 67, 39, 25, 55, 54
117.4	28, 76, 79, 78, 34, 35, 71, 65, 66, 20, 1
128.2	31, 62, 19, 11, 63, 64, 49, 36, 47, 48
62.8	53, 40, 21, 73, 74, 72, 4, 26
98.0	52, 88, 7, 82, 8, 46, 45, 17, 84, 5, 89
76.4	12, 80, 68, 24, 29, 3, 77, 50
100.5	61, 16, 86, 38, 14, 44, 91,100, 37, 59, 96
67.6	18, 83, 60, 99, 93, 85, 98, 92, 97, 13
103.8	27, 69, 33, 81, 9, 51, 30, 32, 90, 10, 70
948.6	10

Table B.3. Solution of R202.100.

Cost	Route
8.8	53
93.6	52, 62, 63, 90, 10, 32, 70
177.2	83, 45, 82, 48, 47, 36, 19, 11, 64, 49, 46, 17, 5, 60, 89
223.8	50, 33, 65, 71, 29, 76, 3, 79, 78, 81, 9, 51, 20, 66, 35, 34, 68, 77
140.2	27, 69, 1, 30, 31, 88, 7, 18, 8, 84, 86, 91, 100, 37, 98, 93, 59, 94
67.1	40, 73, 41, 22, 74, 2, 58
148.9	28, 26, 21, 72, 75, 39, 67, 23, 56, 4, 54, 55, 25, 24, 80, 12
170.0	95, 92, 42, 15, 14, 38, 44, 16, 61, 85, 99, 96, 6, 87, 57, 43, 97, 13
1,029.6	8

Table B.4. Solution of R203.100.

Cost	Route
24.2	53, 40, 58
142.1	27, 69, 1, 76, 3, 79, 78, 81, 9, 66, 71, 35, 34, 29, 68, 77, 28
187.3	89, 18, 45, 46, 36, 47, 48, 19, 11, 62, 88, 7, 82, 8, 83, 60, 5, 84, 17, 61, 91, 100, 37, 98, 93, 59, 94
183.3	95, 92, 97, 42, 15, 43, 14, 44, 38, 86, 16, 85, 99, 96, 6, 87, 57, 41, 22, 74, 73, 2, 13
190.3	50, 33, 51, 71, 65, 20, 30, 32, 90, 63, 64, 49, 10, 70, 31, 52
143.6	26, 21, 72, 75, 39, 67, 23, 56, 4, 55, 25, 54, 24, 80, 12
870.8	6

Table B.5. Solution of R207.50.

Cost	Route
202.5	27, 31, 7, 48, 47, 36, 46, 45, 8, 18, 6, 37, 44, 14, 38, 16, 17, 5, 13
130.5	2, 42, 43, 15, 23, 39, 22, 41, 21, 40
242.5	28, 12, 3, 33, 50, 1, 30, 11, 49, 19, 10, 32, 20, 9, 35, 34, 29, 24, 25, 4, 26
575.5	3

Table B.6. Solution of R209.100.

Cost	Route
146.8	52, 7, 82, 83, 18, 6, 94, 13, 87, 57, 15, 43, 42, 97, 92, 37, 100, 91, 93, 96
198.7	95, 99, 59, 98, 85, 5, 84, 61, 16, 44, 14, 38, 86, 17, 45, 8, 46, 36, 49, 48, 60, 89
205.9	27, 69, 31, 88, 62, 47, 19, 11, 64, 63, 90, 30, 51, 71, 9, 81, 33, 79, 3, 77, 68, 80, 24, 54, 26
157.6	28, 12, 76, 29, 78, 34, 35, 65, 66, 20, 32, 10, 70, 1, 50
145.8	40, 2, 73, 21, 72, 75, 23, 67, 39, 25, 55, 4, 56, 74, 22, 41, 58, 53
854.8	5

Table B.7. Solution of RC203.100.

Cost	Route
139.4	81, 54, 72, 37, 36, 39, 42, 44, 41, 38, 40, 35, 43, 61, 68
172.8	90, 65, 83, 64, 85, 63, 89, 76, 23, 21, 48, 18, 19, 49, 22, 20, 51, 84, 56, 66
241.4	69, 98, 88, 53, 82, 99, 52, 86, 87, 9, 10, 47, 17, 13, 74, 59, 97, 75, 58, 77, 25, 24, 57
211.0	1, 3, 5, 45, 60, 12, 11, 15, 16, 14, 78, 73, 79, 7, 6, 8, 46, 4, 2, 55,100, 70
159.1	91, 92, 95, 62, 33, 32, 30, 27, 26, 28, 29, 31, 34, 50, 67, 94, 93, 71, 96, 80
923.7	5

Table B.8. Solution of RC206.100.

Cost	Route
8.4	90
186.6	81, 94, 67, 84, 85, 51, 76, 89, 48, 25, 77, 58, 74
168.6	92, 71, 72, 42, 39, 38, 36, 40, 44, 43, 41, 37, 35, 54, 93, 96
180.9	65, 83, 64, 95, 62, 63, 33, 30, 31, 29, 27, 28, 26, 32, 34, 50, 56, 91, 80
189.6	61, 2, 45, 5, 8, 7, 79, 73, 78, 53, 88, 6, 46, 4, 3, 1, 100, 70, 68
120.9	82, 99, 52, 86, 57, 23, 21, 18, 19, 49, 20, 22, 24, 66
196.1	69, 98, 12, 14, 47, 16, 15, 11, 59, 75, 97, 87, 9, 13, 10, 17, 60, 55
1,051.1	7

Acknowledgments

The authors thank Jacques Desrosiers, HEC, Montreal, Canada, and the anonymous referees for valuable suggestions and corrections.

References

Barnhart, C., E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, P. H. Vance. 1998. Branch-and-price: Column generation for solving huge integer programs. *Oper. Res.* **46** 316–329.

Beasley, J. E., N. Christofides. 1989. An algorithm for the resource constrained shortest path problem. *Networks* **19** 379–394.

Boland, N., J. Dethridge, I. Dumitrescu. 2006. Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Oper. Res. Lett.* **34**(1) 58–68.

Caprara, A., M. Fischetti, A. N. Letchford. 2000. On the separation of maximally violated mod-k cuts. *Math. Programming* **87**(1) 37–56.

Chabrier, A. 2005. Vehicle routing problem with elementary shortest path based column generation. *Comput. Oper. Res.* **23**(10) 2972–2990.

Chvatal, V. 1973. Edmonds polytopes and hierarchy of combinatorial problems. *Discrete Math.* **4** 305–337.

COIN. 2005. COIN—COmputational INfrastructure for Operations Research. www.coin-or.org.

Cook, W., J. L. Rich. 1999. A parallel cutting plane algorithm for the vehicle routing problem with time windows. Technical Report TR99-04, Computational and Applied Mathematics, Rice University, Houston, TX.

Danna, E., C. Le Pape. 2005. Accelerating branch-and-price with local search: A case study on the vehicle routing problem with time windows. G. Desaulniers, J. Desrosiers, M. M. Solomon, eds. *Column Generation*, Chapter 3. Springer, New York, 90–130.

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

Desrochers, M. 1986. La fabrication d'horaires de travail pour les conducteurs d'autobus par une méthode de génération de colonnes. Ph.D. thesis, Université de Montréal, Montréal, Quebec, Canada.

Desrochers, M., J. Desrosiers, M. Solomon. 1992. A new optimization algorithm for the vehicle routing problem with time windows. *Oper. Res.* **40** 342–354.

Dror, M. 1994. Note on the complexity of the shortest path models for column generation in VRPTW. *Oper. Res.* **42** 977–979.

Dumitrescu, I. 2002. Constrained path and cycle problems. Ph.D. thesis, Department of Mathematics and Statistics, University of Melbourne, Melbourne, Australia.

Eisenbrand, F. 1999. Note on the membership problem for the elementary closure of a polyhedron. *Combinatorica* **19**(2) 297–300.

Feillet, D., P. Dejax, M. Gendreau, C. Gueguen. 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks* **44**(3) 216–229.

Fukasawa, R., H. Longo, J. Lysgaard, M. Poggi de Aragão, M. Reis, E. Uchoa, R. F. Werneck. 2006. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Math. Programming* **106**(3) 491–511.

Irnich, S. 2006. Resource extension functions: Properties, inversion, and generalization to segments. Technical Report 2006-01, Deutsche Post Endowed Chair of Optimization of Distribution Networks, RWTH Aachen University, Aachen, Germany. www.dpor.rwth-aachen.de.

Irnich, S., G. Desaulniers. 2005. Shortest path problems with resource constraints. G. Desaulniers, J. Desrosiers, M. M. Solomon, eds. *Column Generation*, Chapter 2. Springer, New York, 33–65.

Irnich, S., D. Villeneuve. 2006. The shortest path problem with resource constraints and k-cycle elimination for $k \geqslant 3$. *INFORMS J. Comput.* **18**(3) 391–406.

Jepsen, M., B. Petersen, S. Spoorendonk. 2005. A branch-and-cut-and-price framework for the VRP applied on CVRP and VRPTW. Master's thesis, DIKU, University of Copenhagen, Copenhagen, Denmark.

Kallehauge, B., J. Larsen, O. B. G. Madsen. 2000. Lagrangean duality and non-differentiable optimization applied on routing with time windows—Experimental results. Technical Report IMM-REP-2001-9, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.

Kohl, N., J. Desrosiers, O. B. G. Madsen, M. M. Solomon, F. Soumis. 1999. 2-path cuts for the vehicle routing problem with time windows. *Transportation Sci.* **33**(1) 101–116.

Larsen, J. 1999. Parallelization of the vehicle routing problem with time windows. Ph.D. thesis, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.

Lubbecke, M. E., J. Desrosiers. 2005. Selected topics in column generation. *Oper. Res.* **53**(6) 1007–1023.

Lysgaard, J. 2003. CVRPSEP: A package of separation routines for the capacitated vehicle routing problem. www.asb.dk/∼lys.

Lysgaard, J., A. N. Letchford, R. W. Eglese. 2004. A new branch-and-cut algorithm for the capacitated vehicle problem. *Math. Programming* **100**(2) 423–445.

Nemhauser, G., S. Park. 1991. A polyhedral approach to edge coloring. *Oper. Res. Lett.* **10** 315–322.

Nemhauser, G. L., L. A. Wolsey. 1988. *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc., New York.

Pisinger, D., L. B. Reinhardt. 2006. Multi-objective non-additive shortest path. Working paper, DIKU, University of Copenhagen, Copenhagen, Denmark.

Righini, G., M. Salani. 2004. Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Technical Report 66, Note del Polo—Ricerca, Dipartimento di Tecnologie dell'Informazione, Università degli Studi di Milano, Milano, Italy.

Righini, G., M. Salani. 2005. New dynamic programming algorithms for the resource constrained shortest path problem. Technical Report 69, Note del Polo—Ricerca, Dipartimento di Tecnologie dell'Informazione, Università degli Studi di Milano, Milano, Italy.

Salani, M. 2005. Branch-and-price algorithms for vehicle routing problems. Ph.D. thesis, Università degli Studi di Milano, Facoltà di Scienza Matematiche, Fisiche e Naturali, Dipartimento di Tecnologie dell'Informazione, Milano, Italy.

Solomon, M. M. 1987. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Oper. Res.* **35** 234–265.

SPEC. 2005. Standard Performance Evaluation Corporation. www.spec.org.

Toth, P., D. Vigo. 2002. *The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications*, Vol. 9. SIAM, Philadelphia, PA.

Wolsey, L. A. 1998. *Integer Programming*. John Wiley & Sons, Inc., New York.