Datalogisk Institut
KØBENHAVNS UNIVERSITET

A Non-Robust Branch-And-Cut-And-Price Algorithm for the Vehicle Routing Problem with Time Windows

Mads Jepsen, Simon Spoorendonk
Bjørn Petersen and David Pisinger

Technical Report no. 06/03
ISSN: 0107-8283

Dept. of Computer Science
University of Copenhagen • Universitetsparken 1
DK-2100 Copenhagen • Denmark

A non-robust Branch-and-Cut-and-Price algorithm for the Vehicle Routing Problem with Time Windows

Mads Jepsen, Bjørn Petersen, Simon Spoorendonk, David Pisinger*
Department of Computer Science (DIKU), University of Copenhagen,
Universitetsparken 1, DK-2100 Copenhagen Ø, Denmark,
mitzi@diku.dk, bjorn@diku.dk, spooren@diku.dk, pisinger@diku.dk

February 2006. Revised: April 2006

This paper presents a non-robust Branch-and-Cut-and-Price algorithm for the Vehicle Routing Problem with Time Windows. The standard Dantzig-Wolfe decomposition leads to a Set Partition Problem as master problem and an Elementary Shortest Path Problem with Resource Constraints as the pricing problem. In a non-robust algorithm the structure of the pricing problem is modified, this includes adding additional constraints or variables to the formulation. The modification of the pricing problem arises with the introduction of the *subset row inequalities* used as cutting planes in the master problem. We show that the subset row inequalities are valid for the Set Packing Problem and correspond to the set of clique and odd hole inequalities which are also known to be valid inequalities for the Set Partition Problem. The pricing problem is solved with a bi-directional label setting algorithm modified to handle the non-robustness. The introduction of the subset row inequalities made it possible to solve 10 previously unsolved instances from the Solomon's benchmark tests.

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 distance possible. In this paper we consider a homogenous fleet, i.e. all vehicles are identical.

The standard Dantzig-Wolfe decomposition of VRPTW is to split the problem into a master problem (a Set Partition Problem) and a pricing problem (an Elementary Shortest Path Problem with Resource Constraints (ESPPRC), where demand and time are the constrained resources). Applying cutting planes in the master problem leads to a Branch-and-Cut-and-Price algorithm (BCP).

Based on an algorithm by Desrochers et al. (1992), Desrosiers et al. (1984) BCP algorithms have been intensively studied since the beginning of the nineties when solving the VRPTW. Kohl et al. (1999) applied subtour elimination constraints, and introduced *two-path* cuts as cutting planes in the master problem of the VRPTW. Cook and Rich (1999) used a generalization of the *two-path* cuts called *k-path* cuts. Common for these BCP algorithms are that all applied cuts are valid inequalities for the VRPTW that have been decomposed into the master problem, hence the expression of the cuts in the master problem does not lead to a modified pricing problem. This leads to a robust BCP algorithm. The topic has been thoroughly surveyed by Barnhart et al. (1998) for the general case. In this paper we extend the framework to include valid inequalities for the master problem leading to a non-robust BCP algorithm. Nemhauser and Park (1991) developed a non-robust BCP algorithm for the Edge Coloring Problem but to our knowledge no non-robust algorithms for VRPTW have been presented.

* DIKU Technical Report 06–03, University of Copenhagen, Universitetsparken 1, DK-2100 Copenhagen Ø, Denmark

Dror (1994) showed that ESPPRC is strongly $\mathcal{NP}$-hard. So far the most successful BCP algorithms have used dynamic programming to solve a relaxation of ESPPRC. Desrosiers et al. (1984) denoted the solution of the relaxed pricing problem a q-route since the solutions were allowed to contain cycles. Later the relaxed pricing problem was denoted the Shortest Path Problem with Resource Constraints (SPPRC). Irnich and Villeneuve (2006) developed an algorithm for the SPPRC with k-cycle elimination (k-cyc-SPPRC) where cycles containing k or less nodes are not permitted. This has shown to significantly improve lower bounds of the master problem compared to the SPPRC.

Beasley and Christofides (1989) were the first to propose solving the ESPPRC directly. Label setting algorithms have been the most popular approach to solve the ESPPRC. Algorithms have been implemented by Dumitrescu (2002) and Feillet et al. (2004), the latter being the first to use a label setting algorithm for ESPPRC in a VRPTW context as they compared lower bounds obtained with elementary and non-elementary pricing problem algorithms. Later both Chabrier (2005) and Danna and Pape (2005) successfully solved several previously unsolved VRPTW instances from the benchmarks of Solomon (1987) using a label setting algorithm for the ESPPRC.

Righini and Salani (2004) developed a label setting algorithm using the idea of Dijkstra's bi-directional 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. In Petersen and Spoorendonk (2006) this result is extended to the label setting algorithms for the k-cyc-SPPRC, however we only consider the elementary case in this paper.

The paper is outlined as follows: In Section 2 we give an overview of the Dantzig-Wolfe decomposition of VRPTW and describe how to calculate the reduced cost of columns when column generation is used to solve the master problem. Furthermore we describe how both robust and non-robust cutting schemes affect the reduced cost of a column when applied to the master problem.

In Section 3 we introduce the subset row inequalities and show that the separation problem is $\mathcal{NP}$-complete. In Section 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 (including bi-directionality) we refer to Righini and Salani (2004) and Petersen and Spoorendonk (2006). This is followed by an algorithmic outline and extensive computational results, using the Solomon benchmark tests, in Section 5. Finally, in Section 6, we have some concluding remarks.

2. Decomposition

Let C be the set of customers, let the set of nodes be $V = C \cup \{0\}$ where $\{0\}$ denotes the depot, 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, each having capacity D, and let d_i be the demand of customer $i \in C$. 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 between $i,j \in V$ and let x_{ijk} be a variable indicating whether vehicle $k \in K$ travels from $i \in V$ to $j \in V$. Last let $\tau_{ij} = c_{ij} + s_i$, be the travel time between $i,j \in V$ plus the service time of customer i. The 3-index flow model from Toth and Vigo (2002) for VRPTW is:

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

$$\text{s.t.} \sum_{k \in K} \sum_{j \in V} x_{ijk} = 1 \qquad \forall i \in C \tag{2}$$

$$\sum_{i \in C} x_{0ik} = \sum_{i \in C} x_{i0k} = 1 \qquad \forall k \in K \tag{3}$$

$$\sum_{j \in V} x_{jik} - \sum_{j \in V} x_{ijk} = 0 \qquad \forall i \in C,\ \forall k \in K \tag{4}$$

$$\sum_{i \in C} \sum_{j \in V} d_i x_{ijk} \leq D \qquad k \in K \tag{5}$$

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

$$x_{0jk} = 1 \Rightarrow a_0 + \tau_{0j} \leq t_{jk} \qquad \forall j \in C,\ \forall k \in K \tag{7}$$

$$x_{ijk} = 1 \Rightarrow t_{ik} + \tau_{ij} \leq t_{jk} \qquad \forall i \in C,\ \forall j \in V,\ \forall k \in K \tag{8}$$

$$t_{ik} \geq 0 \qquad \forall i \in V,\ \forall k \in K \tag{9}$$

$$x_{ijk} \in \{0,1\} \qquad \forall i,j \in V,\ \forall k \in K \tag{10}$$

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) is the flow conservation, while (5) ensures that the capacity of each vehicle is not exceeded. Constraints (6), (7) and (8) ensure that the time windows are satisfied. The last two constraints define the domain of the variables.

The standard Dantzig-Wolfe decomposition of VRPTW leads to the following master problem:

$$\min \sum_{p \in P} \sum_{i,j \in V} c_{ij} \alpha_{ijp} \lambda_p \tag{11}$$

$$\text{s.t} \sum_{p \in P} \sum_{j \in V} \alpha_{ijp} \lambda_p = 1 \qquad \forall i \in C \tag{12}$$

$$\lambda_p \in \{0,1\} \qquad \forall p \in P \tag{13}$$

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 Partition Problem and the LP relaxation may be solved using delayed column generation. Let π be the dual variables of (12) and let $\pi_0 = 0$. Then the reduced cost of a route p is:

$$\hat{c}_p = \sum_{i,j \in V} c_{ij} \alpha_{ijp} - \sum_{i,j \in V} \pi_j \alpha_{ijp} = \sum_{i,j \in V} (c_{ij} - \pi_j) \alpha_{ijp} \tag{14}$$

The pricing problem becomes an ESPPRC where the cost of each edge is:

$$\hat{c}_{ij} = c_{ij} - \pi_j \qquad \forall i,j \in V$$

When applying cuts during column generation we will distinguish between valid inequalities for the VRPTW polytope (2)-(10) and valid inequalities for the Set Partition polytope (12)-(13). Applying inequalities which are valid for (2)-(10) does not change the structure of the pricing problem and has been the approach used in previous robust BCP algorithms for VRPTW.

Consider a valid inequality for the polytope (2)-(10):

$$\sum_{i,j \in V} \beta_{ij} x_{ij} \leq \beta_0 \tag{15}$$

When decomposed into the master problem, inequality (15) is formulated as:

$$\sum_{p \in P} \sum_{i,j \in V} \beta_{ij} \alpha_{ijp} \lambda_p \leq \beta_0 \tag{16}$$

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

$$\begin{aligned} \hat{c}_p &= \sum_{i,j \in V} c_{ij} \alpha_{ijp} - \sum_{i,j \in V} \pi_j \alpha_{ijp} - \mu \sum_{i,j \in V} \beta_{ij} \alpha_{ijp} \\ &= \sum_{i,j \in V} (c_{ij} - \pi_j - \mu \beta_{ij}) \alpha_{ijp} \end{aligned} \tag{17}$$

Compared to (14) an additional coefficient $\mu\beta_{ij}$ is subtracted from the cost of edge (i,j) and the pricing problem remains an ESPPRC, i.e. the BCP algorithm remains robust.

Given a valid inequality for the polytope (12)-(13):

$$\sum_{p\in P}\beta_p\lambda_p \leq \beta_0 \tag{18}$$

Let $\sigma \leq 0$ be the dual variable of (18). The reduced cost of a column p is:

$$\hat{c}_p = \sum_{i,j\in V} c_{ij}\alpha_{ijp} - \sum_{i,j\in V}\pi_j\alpha_{ijp} - \sigma\beta_p \tag{19}$$

In addition to the reduced cost computed for a column p in (14) (or (17)) the cost $-\sigma\beta_p$ must be considered when $\sigma\beta_p \neq 0$. If $\sigma\beta_p = 0$ no additional cost needs to be considered and the pricing problem remains unaltered. To reflect the possible extra cost $-\sigma\beta_p$ it may be necessary to modify the pricing problem by adding constraints or variables, thus leading to a non-robust BCP algorithm. We have developed a solution method that exploits a special structure of the subset row inequalities introduced in the following section.

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 Partition Problem since the Set Partition Problem is a special case of the Set Packing Problem. 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 Partition Problem (Nemhauser and Wolsey (1988)).

Since the master problem previously has been identified as a Set Partition Problem, it would be obvious to go in this direction when looking for valid inequalities for the master problem. However, neither clique nor odd hole inequalities are directly applicable to the master problem when column generation is used due to the impact on the pricing 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:

$$E = \left\{ e(p,q) : \sum_{j\in V}\alpha_{ijp} = 1 \wedge \sum_{j\in V}\alpha_{ijq} = 1,\ i\in C,\ p,q\in P \right\}$$

That is, an edge is present if the two columns p, 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:

$$\sum_{p\in\hat{P}}\lambda_p \leq 1 \tag{20}$$

where $\hat{P} \subseteq P$ are the columns corresponding to the vertices of a clique in G. A cycle visiting an odd number $n \geq 5$ of vertices in G leads to the valid odd hole inequality:

$$\sum_{p\in\hat{P}}\lambda_p \leq \left\lfloor \frac{n}{2} \right\rfloor \tag{21}$$

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 (20) or (21) in the pricing problem as described in Section 2 since there is no specific knowledge of the columns of the master problem when solving the pricing problem.

Inspired by the above inequalities (20) and (21) we introduce the *subset row inequalities* (SR-inequalities). These inequalities are specifically linked to the rows of the Set Packing Problem, hence making it possible to identify the coefficient of a column in a SR-inequality.

DEFINITION 1. Consider the Set Packing polytope

$$X = \{\lambda \in \mathbb{B}^{|P|} : A\lambda \leq 1\} \tag{22}$$

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

$$\sum_{p \in P} \left\lfloor \frac{1}{k} \sum_{i \in S} \alpha_{ip} \right\rfloor \lambda_p \leq \left\lfloor \frac{n}{k} \right\rfloor \tag{23}$$

where $S \subseteq R$, $|S| = n$ and $0 < k \leq n$.

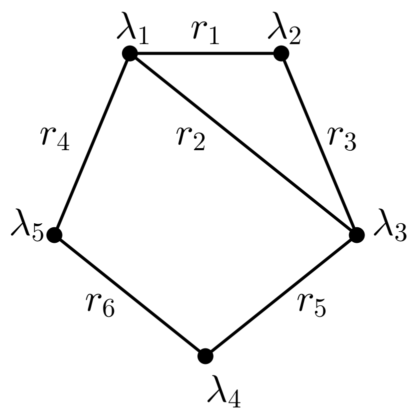
Example 1 illustrates some SR-inequalities derived from the conflict graph of a Set Packing Problem.

EXAMPLE 1.
SR-inequalities derived from the conflict graph of a Set Packing Problem. In the LP-solution to $A\lambda \leq 1$ 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 $S = \{r_1, r_3, r_4, r_5, r_6\}$

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

Set Packing Problem $A\lambda \leq 1$.



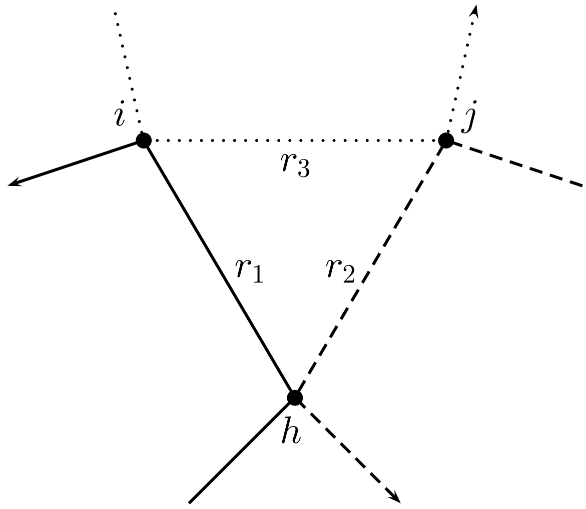
Corresponding conflict graph.

Given a column $p \in P$ we need to have $\sum_{i \in S} \alpha_{ip} \geq k$ to get a non-zero coefficient of λ_p in (23). For the master problem of VRPTW the coefficient matrix can be translated as $\alpha_{ip} = \sum_{j \in V} \alpha_{ijp}$, i.e., α_{ip} is the sum of all the outgoing edges of a customer i. Hence,

$$\frac{1}{k} \sum_{i \in S} \alpha_{ip} = \frac{1}{k} \sum_{i \in S} \sum_{j \in V} \alpha_{ijp}$$

which is only larger than 0 when more than k customers of S are visited on route i. Refer to Example 2 for a violated SR-inequality in a master problem LP solution for a VRPTW.

EXAMPLE 2.
A violated SR-inequality with $n=3$, $k=2$, and $S=\{i,j,h\}$. Let $P=\{r_1,r_2,r_3\}$ be three routes each visiting two of the customers in S. The SR-inequality in our case becomes $\lambda_{r_1}+\lambda_{r_2}+\lambda_{r_3}\leq 1$. In an LP-solution $\lambda_{r_1}=\lambda_{r_2}=\lambda_{r_3}=\frac{1}{2}$, which clearly violates the SR-inequality.



PROPOSITION 1. *The SR-inequalities* (23) *are valid for the Set Packing polytope X.*

Proof. The proof follows directly from Chavtal-Gomory's procedure to construct valid inequalities (Wolsey (1998)). Scale the n inequalities $\sum_{i\in S}\alpha_{ip}\lambda_p\leq 1$ in the set of rows $S\subseteq R$ from (22) with $\frac{1}{k}\geq 0$ and add them:

$$\sum_{p\in P}\frac{1}{k}\sum_{i\in S}\alpha_{ip}\lambda_p\leq\frac{n}{k}$$

Flooring on left side and right side leads to (23). $\square$

Note that, when the coefficient $\left\lfloor\frac{1}{k}\sum_{i\in S}\alpha_{ip}\right\rfloor$ evaluates to 0 or 1 for all $p\in P$, the set of SR-inequalities (23) corresponds to the clique inequalities (20) when $\left\lfloor\frac{n}{k}\right\rfloor=1$ and corresponds to the odd hole inequalities (21) when $n\geq 5$ and $k=2$, i.e., the right hand side of (23) becomes $\left\lfloor\frac{n}{2}\right\rfloor$.

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 $p\in P$, and some fixed values n and k where $1<k\leq n$, find the most violated SR-inequality. Using the binary variable x_i to denote whether $i\in S$ this can be stated as:

$$\max\sum_{p\in P}\left\lfloor\frac{1}{k}\sum_{i\in R}a_{ip}x_i\right\rfloor\lambda_p-\left\lfloor\frac{n}{k}\right\rfloor \tag{24}$$

$$\text{s.t.}\sum_{i\in R}x_i=n \tag{25}$$

$$x_i\in\{0,1\}\qquad\forall i\in R \tag{26}$$

The corresponding decision problem SR-DECISION asks whether

$$\sum_{p\in P}\left\lfloor\frac{1}{k}\sum_{i\in R}a_{ip}x_i\right\rfloor\lambda_p\geq c \tag{27}$$

is feasible subject to (25) and (26), where $1 \le c < n$ and $c \in \mathbb{Z}$. Since we may multiply (27) by any coefficient $\frac{1}{\gamma} > 0$, the coefficient bounds $\lambda_p < 1$ and $c < n$ can be softened to

$$\lambda_p < \frac{1}{\gamma}, \qquad c < \frac{n}{\gamma} \tag{28}$$

This leads to the following proposition:

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

Proof. We will show the stated 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 $C_1, \dots, C_m$ 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 $x_1, \dots, x_n$ be the set of variables which 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., $R = \{1, \dots, 2n+3\}$ and $P = \{1, \dots, m+n+1\}$.

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

We now define matrix A as follows: For $j = 1, \dots, m$ let $a_{ij} = 1$ iff the corresponding literal appears in clause C_j. For $j = 1, \dots, n$ 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 $j = 1, \dots, m+n$ let $a_{2n+1,j} = 0$, while $a_{2n+1,m+n+1} = 1$. For $j = 1, \dots, m+n+1$ 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 (28) for γ sufficiently large. An example of the transformation is illustrated in Example 3.

EXAMPLE 3.
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	…	m	$m+1$	…	…	$m+n$	$m+n+1$
		C_1	…	C_m	x_1	…	…	x_n	
1	x_1	1			1				
2	$\neg x_1$	1		1	1				
	x_2		1			1			
⋮	$\neg x_2$	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 (27) reads

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

which is satisfied if and only if

$$\sum_{i\in R} a_{ip}x_i \geq 3 \qquad \forall p\in P$$

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

$$\sum_{i=1}^{2n} a_{ip}x_i \geq 1 \qquad \forall p = 1,\dots,m+n$$

(i) Assume that there is a feasible assignment of binary values to $x_1,\dots,x_n$ 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. Since exactly n literals are true, we will in this way choose n rows. Since at least one literal is true in each clause, and each column $1,\dots,m$ corresponds to a clause in A we will get a contribution of at least one in each of these columns. Moreover, since exactly one of x_i and $\neg x_i$ is true in ϕ we will get a contribution of exactly one in column $m+1,\dots,m+n$. 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 $m+1,\dots,m+n$ 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, since the rows P' must cover at least one $a_{pi} = 1$ for each column $j = 1,\dots,m$, we see that each clause in ϕ becomes true.

Since the reduction is polynomial, and SR-DECISION obviously is in $\mathcal{NP}$, we have proved the stated. □

Example 4 shows that typical separation problems of SR–inequalities actually possess the properties assumed in the $\mathcal{NP}$-completeness proof.

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), Chabrier (2005), Danna and Pape (2005), Dumitrescu (2002), Feillet et al. (2004), hence in the following we will mainly focus 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 $e\in E$ and resource $r\in R$ three parameters are given: A lower limit $a_r(e)$ on the accumulation of resource r when traversing edge $e\in E$; an upper limit $b_r(e)$ on the accumulation of resource r when traversing edge $e\in E$; and finally an amount $c_r(e)$ of resource r consumed by traversing edge $e\in E$. The objective is to find a minimum cost path p from a source node $s\in V$ to a target node $t\in V$, where the accumulated resources of p

Example 4.
To illustrate that the bounds (28) indeed are realistic consider the case $k=3$. Chose $\gamma = \frac{m+n+1}{\beta}$ where $\beta = \frac{n-2}{3}$ or $\beta = \frac{n-1}{3}$ depending on which of the expressions that evaluates to an integral value. The right hand side of (27) 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

$$\lambda_p \cdot \frac{1}{\gamma} = 1 \cdot \frac{\beta}{m+n+1} \leq 1 \qquad \forall p \in P$$

Hence all bounds are valid according to the separation problem (24)-(26).

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 e, i.e. before $c_r(e)$ has been consumed.

It is noted 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 $v \in V$ and cost $\overline{c}$. Notice that also the cost is considered a resource $\overline{c}$. When considering the pricing problem of VRPTW, the consumptions and upper and lower limits of the resources at each edge e in ESPPRC are:

$$\begin{array}{llll}
a_d(e)=0, & b_d(e)=D-d_j, & c_d(e)=d_j & \forall e=(i,j)\in E \\
a_t(e)=a_i, & b_t(e)=b_i, & c_t(e)=\tau_{ij} & \forall e=(i,j)\in E \\
a_v(e)=0, & b_v(e)=1, & c_v(e)=1 \quad \forall v\in V: v=j, & \forall e=(i,j)\in E \\
a_v(e)=0, & b_v(e)=1, & c_v(e)=0 \quad \forall v\in V: v\neq j, & \forall e=(i,j)\in E \\
a_{\overline{c}}(e)=-\infty, & b_{\overline{c}}(e)=\infty\ , & c_{\overline{c}}(e)=c_{ij} & \forall e=(i,j)\in E
\end{array}$$

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

$\overline{v}(L)$ The node L belongs to, i.e. the current end-node of the partial path represented by L.
$r(L)$ The accumulated consumption of resource $r \in R$.

A feasible extension $\epsilon \in \mathcal{E}(L)$ of a label L is a partial path starting in a node $\overline{v}(L) \in V$ and ending in the target node t, that does not violate any resources when concatenated with the partial path represented by label 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 is below the lower limit at a given edge e, 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 related by an edge $e=(u,v)$, i.e. L_u is extended and becomes L_v, where $\overline{v}(L_u)=u$ and $\overline{v}(L_v)=v$ must satisfy

$$r(L_u) \leq b_r(e), \qquad \forall r \in R \qquad (29)$$
$$r(L_v) = \max\{r(L_u)+c_r(e), a_r(e)\}, \qquad \forall r \in R \qquad (30)$$

Here (29) demands that each label L_u satisfies the upper limit of resource r corresponding to edge $e=(u,v)$, while (30) states that resource r at label L_v corresponds to the resource consumption at label L_u plus the amount consumed by traversing edge e, respecting the lower limitat edge e.

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 which will not lead to an optimal solution.

Definition 2. A label L_i dominates label L_j if

$$\overline{v}(L_i)=\overline{v}(L_j) \tag{31}$$
$$\overline{c}(L_i)\leq\overline{c}(L_j) \tag{32}$$
$$\mathcal{E}(L_j)\subseteq\mathcal{E}(L_i) \tag{33}$$

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

To determine if (33) holds can be quite cumbersome, as the straightforward definition demands that we calculate all extension of the two labels, therefore a sufficient criteria for (33) is sought which can be computed faster. If label L_i has consumed less 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 **(Sufficient condition)**. *Label L_i dominates label L_j if:*

$$\overline{v}(L_i)=\overline{v}(L_j) \tag{34}$$
$$r(L_i)\leq r(L_j) \qquad \forall r\in R \tag{35}$$

Proof. We check Definition 2. Equation (31) follows directly from (34) and (32) follows from (35) with $r=\overline{c}$, i.e. the cost resource. Since all resources (including node visits) $r\in R$ in L_i are consumed less or equal to the consumption in L_j all paths feasible for L_j must also be possible for L_i, i.e. $\mathcal{E}(L_j)\subseteq\mathcal{E}(L_i)$ and (33) holds. □

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 in order to determine if some extensions are impossible. Or in other words: update the node resources in an eager fashion instead of a lazy. The following definition is a generalization of (Feillet et al. 2004, Definition 3).

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

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

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 $\overline{v}(L)\in V$, and $v(L)=0$ otherwise.

This leads to the following dominance criteria introduced by Feillet et al. (2004).

Proposition 4 **(Sufficient condition)**. *Label L_i dominates label L_j if:*

$$\overline{v}(L_i)=\overline{v}(L_j) \tag{36}$$
$$r(L_i)\leq r(L_j) \qquad \forall r\in R \tag{37}$$

and node resources are set according to Definition 3.

Proof. We check Definition 2. Constraints (31) and (32) checked as in the proof of Proposition 3. The remaining concern is if (33) holds for L_i and L_j.

The proof is by contradiction. Assume that (36) and (37) are satisfied but that (33) is not satisfied. Then there must exist an extension $\epsilon \in \mathcal{E}(L_j) \backslash \mathcal{E}(L_i)$, i.e. ϵ is feasible for L_j but not for L_i. Let L^u denote the label that is obtained with $\overline{v}(L^u) = v_u$ after L has recursively been extended through ϵ, let $v_1, \dots, v_{h-1}, v_h, \dots$ be the nodes on ϵ and let v_h be the first node on ϵ preventing the extension of L_i^{h-1}. There are only two conditions where this can happen:

1) $v_h(L_i^{h-1}) = 1$
2) $\exists r \in R, \qquad r(L_i^{h-1}) + l_r(v_{h-1}, v_h) > b_r(h)$

Since L_j can be extended with ϵ the equivalent conditions for L_j^{h-1} are:

1) $v_h(L_j^{h-1}) = 0$
2) $r(L_j^{h-1}) + c_r(v_{h-1}, v_h) \leq b_r(h), \qquad \forall r \in R$

Since all resources are consumed equally on ϵ until v_{h-1} for both L_i and L_j the above conditions contradicts that (36) and (37) are satisfied. Hence, $\mathcal{E}(L_j) \backslash \mathcal{E}(L_i) = \emptyset$ which implies $\mathcal{E}(L_j) \subseteq \mathcal{E}(L_i)$ and (33) holds. That is, Definition 2 holds and L_i dominates L_j. □

Using Proposition 4 as a dominance criteria is a relaxation of the dominance criteria of Definition 2 since only a subset of labels satisfying (31), (32) and (33) satisfies (36) and (37).

4.1. Solving the modified pricing problem

Consider some valid SR-inequality on the form (23)

$$\sum_{p \in P} \left\lfloor \frac{1}{k} \sum_{i \in S} \alpha_{ip} \right\rfloor \lambda_p = \sum_{p \in P} \frac{1}{k} \left\lfloor \sum_{i,j \in V} \alpha_{ijp} \right\rfloor \lambda_p \leq \left\lfloor \frac{n}{k} \right\rfloor \tag{38}$$

where $S \subseteq R$, $|S| = n$ and $0 < k \leq n$. Let $\sigma \leq 0$ be the corresponding dual variable when solving the master problem to LP-optimality. From (19) the reduced cost of a column in the VRPTW master problem is:

$$\hat{c}_p = \sum_{e(i,j) \in E} c_{ij} \alpha_{ijp} - \sum_{e(i,j) \in E \backslash \{\delta^+(0)\}} \pi_j \alpha_{ijp} - \sigma \left\lfloor \frac{\sum_{i \in S} \sum_{j \in V} \alpha_{ijp}}{k} \right\rfloor \tag{39}$$

We will analyze how the label setting algorithm for ESPPRC can be modified to handle this additional cost.

For an ESPPRC the last term in (39) can be interpreted as: A penalty $-\sigma$ must be paid for each k visits to the nodes in S. Let $E(L)$ be the edges and $V(L)$ be the nodes visited on the partial path of label L. The cost of a label L can then be expressed as:

$$c(L) = \sum_{e(i,j) \in E(L)} c_{ij} - \sum_{i \in V(L) \backslash \{0\}} \pi_i - \sigma \left\lfloor \frac{|S \cap V(L)|}{k} \right\rfloor$$

This invalidates the previous dominance criteria of Proposition 4 since the penalty is depending on both visited and unvisited nodes, i.e. if label L_i dominates label L_j regarding Proposition 4, then $\overline{c}(L_j) \leq \overline{c}(L_i)$ could be true later due to penalties paid by L_i but not by L_j which clearly makes it invalid to dominate L_j.

It is sufficient to consider if label L_i is closer to paying a penalty than label L_j. For a label L let

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

be the number of visits made to S since the last penalty was paid for visiting k nodes in S. Recall $\mathcal{E}(L)$ as the set of feasible extensions from the label L to the target node t and note that when label L_i dominates label L_j their common extensions are $\mathcal{E}(L_j)$ due to (33). The following cost dominance criteria is obtained for a single SR-inequality:

PROPOSITION 5. *If $\mathcal{T}(L_i) \le \mathcal{T}(L_j)$ label L_i may dominate label L_j disregarding any additional penalties.*

Proof. Consider any common extension $\epsilon \in \mathcal{E}(L_j)$. The comparison of the number of future penalties for the two labels are:

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

since $\mathcal{T}(L_i) \le \mathcal{T}(L_j)$. Hence L_j will always pay at least the same penalties as L_i using their common extensions and the current cost of L_i can be used during dominance. □

PROPOSITION 6. *If $\mathcal{T}(L_i) > \mathcal{T}(L_j)$ label L_i may dominate label L_j if a temporary penalty $-\sigma$ is added to the cost of L_i.*

Proof. Consider any common extension $\epsilon \in \mathcal{E}(L_j)$. The comparison of the number of future penalties for the two labels are:

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

since $0 \le \mathcal{T}(L_j) < \mathcal{T}(L_i) \le k-1$. Applying an additional penalty to L_i gives:

$$1 + \left\lfloor \frac{|S \cap \mathcal{E}(L_j)| + \mathcal{T}(L_i)}{k} \right\rfloor \le \left\lfloor \frac{|S \cap \mathcal{E}(L_j)| + \mathcal{T}(L_j)}{k} \right\rfloor$$

Adding one additional penalty $-\sigma$ to the cost of L_i during dominance ensures that L_j always pays at least the same penalties as L_i using their common extensions. □

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

The new dominance criteria is as follows:

PROPOSITION 7. *Let Q be the set of all SR-inequalities with dual variables σ where*

$$\sigma_q < 0 \wedge \mathcal{T}_q(L_i) > \mathcal{T}_q(L_j) \qquad q \in Q$$

then label L_i dominates label L_j if:

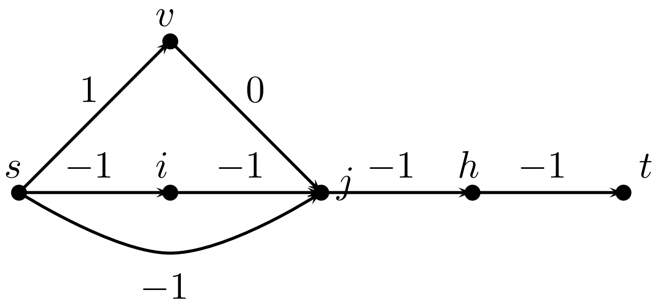
$$\overline{v}(L_i) = \overline{v}(L_j) \tag{40}$$
$$\overline{c}(L_i) - \sum_{q \in Q} \sigma_q \le \overline{c}(L_j) \tag{41}$$
$$r(L_i) \le r(L_j) \qquad \forall r \in R \backslash \{c\} \tag{42}$$

and node resources are set according to Definition 3.

Proof. The validity of (41) follows directly from Propositions 5 and 6. The validity of (40) and (42) is equivalent to the proof of Proposition 4. □

The new dominace criteria is illustrated in Example 5.

EXAMPLE 5.
Consider the instance of ESPPRC depicted in the figure. No resources have influence on the length of any paths.



Let $S = \{i, j, h\}$ be the nodes (customers) of a SR-inequality with $n = 3$ and $k = 2$ and let the dual cost be $\sigma = -2$.
After a few iterations, the label setting algorithm has produced three labels in node j:

L	$V(L)$	$\bar{c}(L)$	$V(L)$	$\mathcal{T}_S(L)$
L_1	j	0	$\{s, i, j\}$	0
L_2	j	−1	$\{s, j\}$	1
L_3	j	1	$\{s, v, j\}$	1

i.e. the label L contains information on the current node $V(L)$, the cost $\bar{c}(L)$, the visited nodes $V(L) = \{s, \dots\}$ and the number of visits $\mathcal{T}_S(L)$ to S since the last penalty was paid.
Although the path visiting nodes s, i and j costs −2 label L_1 pays the penalty of $-\sigma = 2$ since two nodes in the SR-inequality were visited. Normally L_2 would dominate L_1 since it is cheaper, but since $\mathcal{T}_S(L_2) > \mathcal{T}_S(L_1)$ a temporary penalty of 2 must be paid according to (41), hence L_2 is not cheaper than L_1 and does not dominate L_1. Had the path of L_2 been one unit cheaper, it would have dominated L_1 even though a penalty for the unvisited SR-inequality was paid.
L_3 is dominated by L_1 since the cost of L_1 is smaller than that of L_3 although a penalty for the SR-inequality was paid by L_1 and not L_3. Note that L_3 is also dominated by L_2.

Extending the remaining labels L_1 and L_2 to node h gives:

L	$V(L)$	$\bar{c}(L)$	$V(L)$	$\mathcal{T}_S(L)$
L_4	u	−1	$\{s, i, j, h\}$	1
L_5	u	0	$\{s, j, h\}$	0

Although the parent labels of L_4 and L_5 could not dominate each other, it is now possible for L_4 to dominate L_5 even though L_4 is closer to paying the penalty of the SR-inequality. However, it was previously observed that if enough nodes of a SR-inequality could not be reached to trigger a penalty, it could be disregarded which is the case here since only t can be reached from h. Finally label L_4 is extended to t and the shortest path becomes $\{s, i, j, h, t\}$ with a reduced cost of −2.

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:

1. Choose an unprocessed branch node. If the lower bound is above the upper bound, then fathom branch node.
2. Solve the LP master problem.
3. Solve the pricing problem heuristically. If any columns with negative reduced cost is found, then add them to the master problem and go back to Step 2.
4. Solve the pricing problem to optimality. Update lower bound. If the lower bound is above the upper bound, then fathom branch node. If columns are found, then add them to the master problem and go to Step 2.
5. Separate SR-inequalities. If any violated cuts are found, then add them to the master problem and go to Step 2.
6. If the LP solution is fractional, then add the children to the unprocessed branch node and mark the current node as processed.

We allow a maximum of 400 variables and 50 cuts to be generated in each of steps 4, 5, and 6 respectfully. The pricing problem heuristic is based on the label setting algorithm but a simpler heuristic dominance criteria 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 are done with a complete enumeration of all inequalities with $n = 3$ and $k = 2$. Let B be the set of basis 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. (2005): 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.,

$$\sum_{k \in K} \sum_{i \in S} \sum_{j \in V \setminus S} (x_{ijk} + x_{jik}) = 2$$

and

$$\sum_{k \in K} \sum_{i \in S} \sum_{j \in V \setminus S} (x_{ijk} + x_{jik}) \geq 4$$

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 five times faster than that of Irnich and Villeneuve.

Table 1 Comparison of computer speed.

Author(s)	CPU	SpecINT	SpecCFP	Normalized
Irnich and Villeneuve	P3 600 MHz*	295	204	1.0
Chabrier	P4 1.5 GHz	526	606	2.4
Jepsen et al.	P4 3.0 GHz	1152	1201	4.9

Based on CPU2000 benchmarks from SPEC (2005). (*) benchmarks are given for P3 650 MHz since no benchmarks were available for P3 600. The normalized value is an average of SpecINT and SpecCFP.

A comparison of running times is shown in Table 2. We regard the Solomon instances that were closed by either Irnich and Villeneuve or Chabrier and have been solved by us. Hence we consider these instances to be among the hardest solvable of the Solomon benchmarks.

Table 2 Comparison of running time for instances recently solved by at least two of the references Irnich and Villeneuve (2006), Chabrier (2005) and Jepsen et al..

	Irnich and Villeneuve	Chabrier	Jepsen et al.			
Instance	Time (s)	Time (s)	Time (s)	Speedup		
R104.100	268106.0	-	**32343.9**	1.7	/	-
RC104.100	986809.0	-	**65806.8**	3.1	/	-
RC107.100	42770.7	-	**153.8**	56.8	/	-
RC108.100	71263.0	-	**3365.0**	4.3	/	-
R203.50	**217.1**	3320.9	50.8	0.9	/	32.0
R204.25	123.1	171.6	**7.5**	3.3	/	11.1
R205.50	585.7	531.0	**15.5**	7.7	/	16.8
R206.50	22455.3	4656.1	**190.9**	24.0	/	11.9
R208.25	321.9	741.5	***2.9**	22.7	/	125.2
R209.50	142.4	195.4	**16.6**	1.8	/	5.8
R210.50	11551.4	65638.6	***332.7**	7.1	/	96.6
R211.50	**21323.0**	-	10543.8	0.4	/	-
RC202.50	241.6	**13.0**	*10.7	4.6	/	0.6
RC202.100	124018.0	19636.5	**312.6**	194.3	/	30.8
RC203.25	1876.0	5.1	***0.7**	531.7	/	3.5
RC203.50	54229.2	4481.5	***190.9**	58.2	/	11.4
RC204.25	-	13.0	***2.0**	-	/	3.2
RC205.50	52.6	**10.6**	*5.9	1.8	/	0.9
RC205.100	13295.9	15151.7	**221.2**	12.3	/	33.5
RC206.50	469.1	**9.4**	*8.2	12.4	/	0.6
RC207.50	-	71.1	***21.5**	-	/	1.6
RC208.25	-	33785.3	**78.4**	-	/	211.1

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

As indicated by the table our algorithm outperforms those of Irnich and Villeneuve and Chabrier for 17 out of 22 instances. Seven of these instances were solved without any SR-inequalities. In this case, the faster running time is mainly due to the bi-directional 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 (e.g., instances R210.50, RC202.100, RC205.100, and RC208.25).

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 only report results on what we consider the hard instances.

Table 3 Comparing root lower bounds on Solomon instances closed by either Irnich and Villeneuve (2006), Chabrier (2005) or Jepsen et al..

		Irnich and Villeneuve		Jepsen et al.	
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	1132.3	3	1114.4	1101.9	1129.9
RC106.100	1372.7	4	1343.1	1318.8	1367.3
RC107.100	1207.8	4	1195.4	1183.4	**1207.8**
RC108.100	1114.2	3	1100.5	1073.5	**1114.2**
R202.100	1029.6	0	933.5	1022.3	1027.3
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	1092.3	3	1055.0	1088.1	**1092.3**
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	1154.0	3	1130.5	1147.7	**1154.0**
RC206.50	610.0	4	597.1	**610.0**	**610.0**
RC206.100	1051.1	3	1017.0	1038.6	**1051.1**
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**

LB by Irnich and Villeneuve 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 **boldface** indicates lower bounds equal to the upper bound. Instances in **boldface** are the Solomon instances closed by Jepsen et al.

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 8 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.

5.3. Closed Solomon's instances

Table 4 gives an overview of how many instances were solved for each class of the Solomon instances. We were able to close 10 previously unsolved instances. We did not succeed to solve four previously solved instances (R204.50, C204.50, C204.100, and RC204.50).

Table 4 Summery of solved Solomona instances.

		25 customers		50 customers		100 customers	
Class	No.	Prev.	Jepsen et al.	Prev.	Jepsen et al.	Prev	Jepsen et al.
R1	12	12	12	12	12	10	12
C1	9	9	9	9	9	9	9
RC1	8	8	8	8	8	7	8
R2	11	11	11	9	9	1	4
C2	8	8	8	8	7	8	7
RC2	8	8	8	7	7	3	5
Summary	56	56	56	53	52	38	45

No. is the number of instances in that class, and for 25, 50 and 100 customers the two columns refers to the number of instances previously solved to optimality and the number of instances solved to optimality by Jepsen et al.

Information on all solved Solomon instances can be found in Tables 6–8 in Appendix A. Furthermore Table 5 provides detailed information of the instances closed by Jepsen et al.. The solutions can be found in Tables 9–18 in Appendix B.

Table 5 Instances closed by Jepsen et al..

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	5911.71	5796.04	77.36	5911.74
R112.100	948.6	946.7	10	9	351	55573.68	199907.03	1598.63	202803.94
RC106.100	1372.7	1367.3	12	37	1035	298.92	4461.64	5214.08	15891.55
R202.100	1029.6	1027.3	8	13	514	974.51	730.04	4810.47	8282.38
R203.100	870.8	870.8	6	1	447	54187.15	48474.45	3973.42	54187.40
R207.50	575.5	575.5	3	1	107	34406.92	34282.47	118.69	34406.96
R209.100	854.8	854.4	5	3	337	31547.45	74779.58	2978.42	78560.47
RC203.100	923.7	923.7	5	1	402	14917.18	13873.53	1025.65	14917.36
RC206.100	1051.1	1051.1	7	1	179	339.63	159.33	171.34	339.69
RC208.50	476.7	476.7	3	1	138	1639.35	1605.40	31.70	1639.40

UB is the optimal solution found by us, *LB* is 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, $Time_{root}$ is the time solving the root node, $Time_{var}$ is time spent solving the pricing problem, $Time_{LP}$ is the time spent solving LP problems, and *Time* is the total time.

6. Concluding Remarks

The introduction of the SR-inequalities significantly improved the results of the BCP algorithm. This made it possible to solve 10 previously unsolved instances from the Solomon's 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 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 in order to solve large VRPTW instances, since 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 only needs 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 non-additive 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 non-additive constraints can be found in Reinhardt and Pisinger (2006). The development of algorithms which efficiently handle non-additive constraints is important to increase the number of valid inequalities which can be handled in a non-robust BCP algorithm.

Appendix A: Results on Solomon's instances

This appendix contains detailed information about solved Solomon benchmark 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 follows. The columns are the lower bound in the rot node, the number of branch tree nodes and the total running time. A (-) means that the instance was not solved and a (d.o.) means no cuts were generated, hence the results are identical with the first columns. The last two columns are the optimal upper bound and a reference to the authors who were the first to solve that instance. The author legend is:

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

Table 6 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			d.o.	617.1	KDMSS
R102	546.4	3	0.13	547.1	1	0.09	547.1	KDMSS
R103	454.6	1	0.11			d.o.	454.6	KDMSS
R104	416.9	1	0.12			d.o.	416.9	KDMSS
R105	530.5	1	0.02			d.o.	530.5	KDMSS
R106	457.3	5	0.29	465.4	1	0.10	465.4	KDMSS
R107	424.3	1	0.12			d.o.	424.3	KDMSS
R108	396.9	3	0.31	397.3	1	0.24	397.3	KDMSS
R109	441.3	1	0.06			d.o.	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			d.o.	191.3	KDMSS
C102	190.3	1	0.53			d.o.	190.3	KDMSS
C103	190.3	1	0.80			d.o.	190.3	KDMSS
C104	186.9	1	3.29			d.o.	186.9	KDMSS
C105	191.3	1	0.17			d.o.	191.3	KDMSS
C106	191.3	1	0.14			d.o.	191.3	KDMSS
C107	191.3	1	0.20			d.o.	191.3	KDMSS
C108	191.3	1	0.37			d.o.	191.3	KDMSS
C109	191.3	1	0.62			d.o.	191.3	KDMSS
RC101	406.7	5	0.20	461.1	1	0.09	461.1	KDMSS
RC102	351.8	1	0.05			d.o.	351.8	KDMSS
RC103	332.8	1	0.19			d.o.	332.8	KDMSS
RC104	306.6	1	0.52			d.o.	306.6	KDMSS
RC105	411.3	1	0.06			d.o.	411.3	KDMSS
RC106	345.5	1	0.10			d.o.	345.5	KDMSS
RC107	298.3	1	0.29			d.o.	298.3	KDMSS
RC108	294.5	1	0.67			d.o.	294.5	KDMSS
R201	460.1	3	0.44	463.3	1	0.27	463.3	CR+KLM
R202	410.5	1	0.61			d.o.	410.5	CR+KLM
R203	391.4	1	0.80			d.o.	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			d.o.	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			d.o.	214.7	CR+L
C202	214.7	1	3.00			d.o.	214.7	CR+L
C203	214.7	1	3.02			d.o.	214.7	CR+L
C204	213.1	1	7.00			d.o.	213.1	CR+KLM
C205	214.7	1	1.10			d.o.	214.7	CR+L
C206	214.7	1	1.75			d.o.	214.7	CR+L
C207	214.5	1	2.70			d.o.	214.5	CR+L
C208	214.5	1	1.85			d.o.	214.5	CR+L
RC201	360.2	1	0.25			d.o.	360.2	CR+L
RC202	338.0	1	0.58			d.o.	338.0	CR+KLM
RC203	326.9	1	0.72			d.o.	326.9	IV+C
RC204	299.7	1	1.95			d.o.	299.7	C
RC205	338.0	1	0.62			d.o.	338.0	L+KLM
RC206	324.0	1	0.87			d.o.	324.0	KLM
RC207	298.3	1	0.88			d.o.	298.3	KLM
RC208	269.1	1	78.42			d.o.	269.1	C

Table 7 Instances with 50 customers.

	with ESPPRC			with ESPPRC and SR				
Instance	LB	Tree	Time (s)	LB	Tree	Time (s)	UB	Ref.
R101	1043.4	3	0.14	1044.0	1	0.09	1044.0	KDMSS
R102	909.0	1	0.27			d.o.	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	1723.29	607.4	23	1601.68	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			d.o.	362.4	KDMSS
C102	361.4	1	1.59			d.o.	361.4	KDMSS
C103	361.4	1	6.06			d.o.	361.4	KDMSS
C104	358.0	1	1564.88			d.o.	358.0	KDMSS
C105	362.4	1	0.49			d.o.	362.4	KDMSS
C106	362.4	1	0.69			d.o.	362.4	KDMSS
C107	362.4	1	0.97			d.o.	362.4	KDMSS
C108	362.4	1	1.55			d.o.	362.4	KDMSS
C109	362.4	1	3.62			d.o.	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			d.o.	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			d.o.	791.9	CR+KLM
R202	698.5	1	9.88			d.o.	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	15400.44	575.5	1	34406.96	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	18545.61	645.6	IV+C
R211	528.7	31	44644.89	535.5	1	10543.81	535.5	IV+DLP
C201	360.2	1	42.07			d.o.	360.2	CR+L
C202	360.2	1	67.05			d.o.	360.2	CR+KLM
C203	359.8	1	214.88			d.o.	359.8	CR+KLM
C204			-			-	350.1	KLM
C205	359.8	1	64.18			d.o.	359.8	CR+KLM
C206	359.8	1	38.91			d.o.	359.8	CR+KLM
C207	359.6	1	72.81			d.o.	359.6	CR+KLM
C208	350.5	1	55.79			d.o.	350.5	CR+KLM
RC201	684.8	1	3.00			d.o.	684.8	L+KLM
RC202	613.6	1	10.69			d.o.	613.6	IV+C
RC203	555.3	1	190.88			d.o.	555.3	IV+C
RC204			-			-	442.2	DLP
RC205	630.2	1	5.88			d.o.	630.2	IV+C
RC206	610.0	1	8.17			d.o.	610.0	IV+C
RC207	558.6	1	21.53			d.o.	558.6	C
RC208			-	476.7	1	1639.40	476.7	**JPSP**

Table 8 Instances with 100 customers.

	with ESPPRC			with ESPPRC and SR				
Instance	LB	Tree	Time (s)	LB	Tree	Time (s)	UB	Ref.
R101	1631.2	57	20.08	1634.0	3	1.87	1637.7	KDMSS
R102	1466.6	1	4.39			d.o.	1466.6	KDMSS
R103	1206.8	19	55.78	1208.7	1	23.85	1208.7	CR+L
R104			-	971.3	3	32343.92	971.5	IV
R105	1346.2	113	126.96	1355.2	5	43.12	1355.3	KDMSS
R106	1227.0	147	511.07	1234.6	1	75.42	1234.6	CR+KLM
R107			-	1064.3	3	1310.30	1064.6	CR+KLM
R108			-	932.1	1	5911.74	932.1	**JPSP**
R109			-	1144.1	19	1432.41	1146.9	CR+KLM
R110			-	1068.0	3	1068.31	1068.0	CR+KLM
R111			-	1045.9	39	83931.48	1048.7	CR+KLM
R112			-	946.7	9	202803.94	948.6	**JPSP**
C101	827.3	1	3.02			d.o.	827.3	KDMSS
C102	827.3	1	12.92			d.o.	827.3	KDMSS
C103	826.3	1	33.89			d.o.	826.3	KDMSS
C104	822.9	1	4113.09			d.o.	822.9	KDMSS
C105	827.3	1	5.34			d.o.	827.3	KDMSS
C106	827.3	1	7.15			d.o.	827.3	KDMSS
C107	827.3	1	6.55			d.o.	827.3	KDMSS
C108	827.3	1	14.46			d.o.	827.3	KDMSS
C109	827.3	1	20.53			d.o.	827.3	KDMSS
RC101	1584.1	59	56.62	1619.8	1	12.39	1619.8	KDMSS
RC102			-	1457.4	1	76.69	1457.4	CR+KLM
RC103			-	1257.7	3	2705.78	1258.0	CR+KLM
RC104			-	1129.9	7	65806.79	1132.3	IV
RC105	1472.0	191	309.83	1513.7	1	26.73	1513.7	KDMSS
RC106			-	1367.3	37	15891.55	1372.7	**JPSP**
RC107			-	1207.8	1	153.80	1207.8	IV
RC108			-	1114.2	1	3365.00	1114.2	IV
R201			-	1143.2	1	139.03	1143.2	KLM
R202			-	1027.3	13	8282.38	1029.6	**JPSP**
R203			-	870.8	1	54187.40	870.8	**JPSP**
R204			-			-	-	-
R205			-			-	-	-
R206			-			-	-	-
R207			-			-	-	-
R208			-			-	-	-
R209			-	854.8	3	78560.47	854.8	**JPSP**
R210			-			-	-	-
R211			-			-	-	-
C201	589.1	1	203.34			d.o.	589.1	CR+KLM
C202	589.1	1	3483.15			d.o.	589.1	CR+KLM
C203	588.7	1	13070.71			d.o.	588.7	KLM
C204			-			-	588.1	IV
C205	586.4	1	416.56			d.o.	586.4	CR+KLM
C206	586.0	1	594.92			d.o.	586.0	CR+KLM
C207	585.8	1	1240.97			d.o.	585.8	CR+KLM
C208	585.8	1	555.27			d.o.	585.8	KLM
RC201			-	1261.7	3	229.27	1261.8	KLM
RC202			-	1092.3	1	312.57	1092.3	IV+C
RC203	922.6	11	34063.95	923.7	1	14917.36	923.7	**JPSP**
RC204			-			-	-	-
RC205			-	1154.0	1	221.24	1154.0	IV+C
RC206			-	1051.1	1	339.69	1051.1	**JPSP**
RC207			-			-	-	-
RC208			-			-	-	-

Appendix B: Solutions of closed Solomon instances

Table 9 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

The left column is the cost of the routes and the total cost. The right column is a comma separated list of the customers visited on the routes and the total number of routes in the last row.

Table 10 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 11 Solution of RC106.100.

Cost	Route
133.4	65, 52, 87, 59, 75, 97, 58, 74
86.5	61, 81, 94, 67, 93, 96, 54
126.0	11, 12, 14, 47, 15, 16, 9, 10, 13, 17
103.4	82, 99, 86, 57, 22, 49, 20, 24
109.4	2, 45, 5, 8, 7, 6, 46, 4, 3, 1,100
105.6	92, 95, 63, 85, 76, 51, 84, 56, 66
131.6	42, 44, 39, 40, 36, 38, 41, 43, 37, 35
127.9	62, 33, 28, 26, 27, 34, 50, 91, 80
130.8	83, 64, 19, 23, 21, 18, 48, 25, 77
160.6	72, 71, 31, 29, 30, 32, 89
91.6	69, 98, 88, 78, 73, 79, 60
65.9	90, 53, 55, 70, 68
1372.7	12

Table 12 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
1029.6	8

Table 13 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 14 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 15 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 16 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 17 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
1051.1	7

Table 18 Solution of RC208.50.

Cost	Route
97.0	12, 14, 47, 17, 16, 15, 13, 9, 11, 10
163.0	2, 6, 7, 8, 46, 5, 3, 45, 4, 1 , 43, 44, 42, 38, 37, 35, 36, 40, 39, 41
216.7	22, 19, 18, 21, 23, 25, 48, 49, 24, 20 , 30, 31, 29, 27, 28, 26, 33, 32, 34, 50
476.7	3

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. *Operations Research* **46** 316–329.

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

Chabrier, A. 2005. Vehicle routing problem with elementary shortest path based column generation. In press: Computers and Operations Research.

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. Tech. rep., Computational and Applied Mathematics, Rice University, Houston, Texas, USA.

Danna, E., C. Le Pape. 2005. *Accelerating branch-and-price with local search: A case study on the vehicle routing problem with time windows*, chap. 3 in Column Generation, G. Desaulniers, J. Desrosiers, and M. M. Solomon (editors). Kluwer Academic Publishers, 90–130.

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

Desrosiers, J., F. Soumis, M. Desrochers. 1984. Routing with time windows by column generation. *Networks* **14** 545–565.

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

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

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., J. Lysgaard, M. Poggi de Aragão, M. Reis, E. Uchoa, R.F. Werneck. 2005. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. Forthcoming in: Mathematical Programming.

Irnich, S., D. Villeneuve. 2006. The shortest path problem with resource constraints and k-cycle elimination for $k \geq 3$. Forthcoming in: INFORMS Journal of Computing.

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, Denmark.

Kallehauge, B., J. Larsen, O.B.G. Madsen. 2000. Lagrangean duality and non-differentiable optimization applied on routing with time windows - experimental results. Tech. Rep. Internal report IMM-REP-2000-8, 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 Science* **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.

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

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

Nemhauser, G., S. Park. 1991. A polyhedral approach to edge coloring. *Operations Research Letters* **10** 315–322.

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

Petersen, B., S. Spoorendonk. 2006. Label setting algorithms for variants of the shortest path problem with resource constraints. DIKU University of Copenhagen, Denmark, Working paper.

Reinhardt, L. Blander, D. Pisinger. 2006. Multi-objective non-additive shortest path. Submitted.

Righini, G., M. Salani. 2004. Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Tech. Rep. 66, Note del Polo - Ricerca, Dipartimento di Tecnologie dell'Informazione, Universitá degli studi di Milano. Submitted to Discrete Applied Mathematics.

Solomon, M. M. 1987. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research* **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.

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