Branch-Cut-and-Price for the Vehicle Routing Problem with Time Windows and Convex Node Costs

Qie He[a], Stefan Irnich[*,b], Yongjia Song[c]

[a] *Department of Industrial and Systems Engineering, University of Minnesota, 111 Church Street SE, Minneapolis, MN 55455, United States.*
[b] *Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz, Jakob-Welder-Weg 9, D-55128 Mainz, Germany.*
[c] *Department of Statistical Sciences and Operations Research, Virginia Commonwealth University, 1015 Floyd Avenue, Richmond, VA 23284, United States.*

Abstract

Two critical yet frequently conflicting objectives for logistics and transportation service companies are improving customer satisfaction and reducing transportation cost. In particular, given a network of customer requests with preferred service times, it is very challenging to find vehicle routes and service schedules simultaneously that respect all operating constraints and minimize the total transportation and customers' inconvenience costs. In this paper, we introduce the Vehicle Routing Problem with Time Windows and Convex Node Costs (VRPTW-CNC), in which we model each customer's inconvenience cost as a convex function of the service start time at that customer. The VRPTW-CNC combines and extends both the standard vehicle routing problem with time windows and some previous results on the optimal service scheduling problem over a fixed route. We propose a branch-cut-and-price algorithm to solve the VRPTW-CNC with general convex inconvenience cost functions. To solve the pricing problem, our labeling algorithm only generates labels that possibly lead to optimal schedule times over a route, which significantly improves the effectiveness of pricing. Extensive computational results demonstrate the effectiveness of our approach.

Key words: Vehicle routing problem, branch-cut-and-price, labeling algorithm, convex node costs, integrated routing and scheduling

1. Introduction

In this paper, we study an extension of the *vehicle routing problem with time windows* (VRPTW, Desaulniers *et al.*, 2014) that minimizes the sum of routing costs and customers' inconvenience costs. The latter inconvenience cost is defined by a general convex function, one for each customer, that expresses the customer's preference for a specific visit time. We call this problem the *vehicle routing problem with time windows and convex node costs* (VRPTW-CNC).

For a more formal definition of the VRPTW-CNC, we rely on the following definition of the VRPTW: Let $N = \{1, 2, \ldots, n\}$ be the set of customers. Each customer $i \in N$ has a given demand

*Corresponding author.

Email addresses: qhe@umn.edu (Qie He), irnich@uni-mainz.de (Stefan Irnich), ysong3@vcu.edu (Yongjia Song)

q_i and a service time window $[e_i, \ell_i]$. A fleet of K homogeneous vehicles, each with capacity Q, is stationed at the depot, which is modeled by 0 and $n+1$ for representing the start and end of a route, respectively. For simplicity, we assume that $q_0 = q_{n+1} = 0$ and the time windows $[e_0, \ell_0]$ and $[e_{n+1}, \ell_{n+1}]$ at the depot are given. The VRPTW is defined on the directed graph $G = (V, A)$ with vertex set $V = N \cup \{0, n+1\}$ and arc set A. Each arc $(i, j) \in A$ has an associated travel time τ_{ij} and routing cost c_{ij}. A feasible route $P = (r_0, r_1, \ldots, r_m)$ in the VRPTW is an elementary 0-$(n+1)$-path in G of length $m > 1$ that respects the time-window and capacity constraints. The route P is time-window feasible if there exist *schedule times* $t_0, t_1, \ldots, t_m \in \mathbb{R}$ with

$$t_i \in [e_{r_i}, \ell_{r_i}] \qquad \forall i \in [0:m] \tag{1a}$$

$$t_{i-1} + \tau_{r_{i-1}, r_i} \le t_i \qquad \forall i \in [1:m] \tag{1b}$$

where $[i:j]$ denotes the integer interval $\{i, i+1, \ldots, j\}$ for $i, j \in \mathbb{Z}$. The route P respects the vehicle capacity if $\sum_{i=0}^{m} q_{r_i} \le Q$. It has routing costs $c_P = \sum_{i=1}^{m} c_{r_{i-1}, r_i}$. The VRPTW asks for a set of up to K feasible routes that visit each customer exactly once and minimize the sum of the routing costs.

The extension introduced in the VRPTW-CNC is that now each vertex $i \in V$ has a convex inconvenience cost function $f_i(t_i)$ (for the sake of generality, we explicitly include the depot copies 0 and $n+1$). The cost of a route $P = (r_0, r_1, \ldots, r_m)$ becomes the sum of the routing costs c_P and the minimum inconvenience cost resulting from the solution of the following $(m+1)$-dimensional optimization problem:

$$f_P = \min \left\{ \sum_{i=0}^{m} f_{r_i}(t_i) \mid t_0, t_1, \ldots, t_m \text{ satisfy (1)} \right\}. \tag{2}$$

We refer to (2) as the *service scheduling problem*. In summary, the VRPTW-CNC extends the normal routing costs c_P of a route P in the VRPTW to $c_P + f_P$, where the component f_P reflects the cost of customers' inconveniences after optimizing the schedule over route P. The VRPTW-CNC is a three-level optimization problem with interdependent levels for clustering, routing, and schedule optimization.

We propose a *branch-cut-and-price* (BCP) algorithm (see Lübbecke and Desrosiers, 2005; Desaulniers *et al.*, 2005) for the solution of the VRPTW-CNC. The novelty of our algorithm lies in the column-generation mechanism: we show that dynamic-programming based labeling algorithms can be used to simultaneously solve the two lower levels of routing and schedule optimization, while the clustering in the first level is standard. Indeed, clustering relies on an extensive, path-based formulation of the VRPTW-CNC as a set partitioning formulation. Let Ω be the set of all feasible VRPTW routes, and let a_{iP} be an indicator on whether or not customer $i \in N$ is served by route $P \in \Omega$. The set partitioning formulation uses variables λ_P for $P \in \Omega$ to select routes as follows:

$$\min \sum_{P \in \Omega} (c_P + f_P) \lambda_P \tag{3a}$$

$$\text{s.t.} \sum_{P \in \Omega} a_{iP} \lambda_P = 1 \qquad \forall i \in N \tag{3b}$$

$$\sum_{P \in \Omega} \lambda_P \le K \tag{3c}$$

$$\lambda_P \ge 0 \text{ integer} \qquad \forall P \in \Omega. \tag{3d}$$

The objective (3a) minimizes the total routing and customers' inconvenience costs, (3b) ensures that all customers are served in a single visit, and (3c) is the fleet size constraint.

1.1. Literature Review

There are three areas of research related to our paper: (1) routing with costs depending on the schedule time; (2) applications of convex node costs; and (3) BCP algorithms for the VRPTW. We review these areas separately.

Routing with Costs Depending on the Schedule Time. The simplest convex (inconvenience) functions are linear functions. Sexton and Bodin (1985a,b) studied a single-vehicle many-to-many dial-a-ride problem with one-sided time window constraints and a linear penalty cost. They solved the problem in a Benders decomposition framework, and used a heuristic to find good solutions for the master routing problem.

Ioachim *et al.* (1998) introduced the shortest path problem with time windows and linear node costs. This is the subproblem in column-generation approaches that have linking constraints between timing decisions in different routes or schedules (Desaulniers *et al.*, 1998). The survey on synchronization in vehicle routing problems (Drexl, 2012) stresses its importance. The algorithm of Ioachim *et al.* (1998) and its refinements have been used to solve aircraft routing and scheduling problems (Ioachim *et al.*, 1999), simultaneous optimization of flight and pilot schedules in a recovery environment (Stojković and Soumis, 2001), periodic airline fleet assignment with time windows and spacing constraints (Bélanger *et al.*, 2006), vehicle routing problems with pickup-and-delivery requests that require a joint operation of active vehicles (such as trucks) and passive vehicles (loading devices such as containers or swap bodies) (Tilk *et al.*, 2016), and technician routing and scheduling problems (Zamorano and Stolletz, 2017). VRPs with soft time windows that have linear penalty terms result in the same type of subproblem with (piecewise) linear node costs (Liberatore *et al.*, 2011). In summary, there is already highly effective algorithms for handling linear node costs in routing and scheduling applications. Therefore, we concentrate on the case of non-linear and strictly convex inconvenience costs in the following. Concentrating on arbitrary, but strictly convex inconvenience cost functions also has the advantage that each such function has a unique minimum, making algorithmic descriptions simpler, because otherwise tie breaking rules have to be specified.

Dumas *et al.* (1990) studied the service scheduling problem over a fixed route with general convex node cost functions, i.e., problem (2). Our approach relies on several insights that were provided in this article. However, our goal is to provide a solution approach for the VRPTW-CNC column-generation subproblem. This problem is the simultaneous routing and schedule optimization problem, while the algorithm of Dumas *et al.* solves only the schedule optimization part. We therefore summarize in Section 2 the most important findings of Dumas *et al.*

Applications of Convex Node Costs. Jaw *et al.* (1986) studied a multi-vehicle many-to-many dial-a-ride problem with service quality constraints and quadratic inconvenience costs related to the delivery time and ride duration. They developed a sequential insertion heuristic to assign customers to vehicles and to determine a feasible schedule of pickups and deliveries of each vehicle. Fagerholt (2001) studied a pickup and delivery problem with soft time windows in the context of maritime transportation. Various types of inconvenience cost functions, including piecewise constant, piecewise linear, and piecewise quadratic functions etc. were studied in their computational experiments. Hashimoto *et al.* (2006) generalized the standard VRPTW by allowing soft time window and soft traveling time constraints, where both constraints are treated as convex cost functions.

BCP Algorithms for the VRPTW. For many variants of the VRP, the current most successful exact algorithms are based on BCP, in which a set partitioning/covering formulation is solved by branch-and-bound, the linear-programming (LP) relaxation of each node in the branch-and-bound tree is solved by column generation, and cuts are used optionally to strengthen the relaxation bound. The pricing problem of column generation can be formulated as an elementary *shortest path problem with resource constraints* (SPPRC, Irnich and Desaulniers, 2005), which has been proved to be strongly NP-hard (Dror, 1994). The most recent enhancements for solving the VRPTW with BCP are summarized in Pecin *et al.* (2017). We highlight the fundamental BCP components that we tailor to our BCP algorithm for the VRPTW-CNC.

Instead of solving the NP-hard elementary SPPRC, Baldacci *et al.* (2011) introduced the ng-path relaxation, which only allows cycles of a certain type: a customer can only be revisited if another customer not in its pre-defined neighborhood has been visited. The size of the ng-neighborhood controls the tradeoff between the practical difficulty of the pricing subproblem and the quality of the relaxation bound. Baldacci *et al.* (2011, 2012b) and many subsequent studies have shown that using medium-sized ng-neighborhoods often significantly reduces the overall BCP solution time.

The average solution time of the dynamic-programming based labeling algorithms for many SPPRC variants can also be reduced by bidirectional labeling, first suggested for the SPPRC by Righini and Salani (2006). In general, bidirectional labeling becomes a key feature when routes can be long (say more than 20 vertices).

Two rather effective families of valid inequalities are the k-path cuts (Kohl *et al.*, 1999) and the subset-row (SR) inequalities (Jepsen *et al.*, 2008), where the first and second preserve the structure of the SPPRC pricing subproblem while the latter SR inequalities require the inclusion of additional resources in the dynamic-programming based labeling algorithms for the SPPRC.

1.2. Contributions and Structure of the Paper

We summarize the main contributions of this paper as follows.

1. We propose the first exact algorithm for the VRPTW-CNC with general convex inconvenience costs, which combines and extends both the standard VRPTW and classical results on the optimal service scheduling problem over a fixed route (Dumas *et al.*, 1990).
2. We propose an effective bidirectional labeling algorithm for the SPPRC with general convex node costs. This algorithm is crucial for efficiently solving the pricing problem in our proposed BCP algorithm for the VRPTW-CNC, and it could potentially be used in other general contexts.
3. We conduct extensive computational experiments to benchmark the performance of the proposed BCP algorithm for solving the VRPTW-CNC. We study the impact of using inconvenience cost functions with different factors on the difficulty of the column-generation subproblems and the overall problems. We also show the value of integrated optimization of routing and scheduling, by comparing the solutions obtained by the VRPTW-CNC and the solutions obtained by a two-stage heuristic that separates the two optimization problems.

The rest of the paper is organized as follows. In Section 2, we summarize the work of Dumas *et al.* (1990) on the service scheduling problem for a fixed route, but we present their findings using updated notations that allow us to better describe our dynamic-programming based labeling algorithm for the column-generation subproblem. Before presenting the actual labeling algorithm, we study in Section 3 the impact of single-vertex extensions on the service scheduling problem. The BCP algorithm with details on column-generation, branching, and cutting is presented in Section 4.

Computational results are discussed in Section 5 before the paper closes with final conclusions in Section 6.

2. Service Scheduling Problem on a Fixed Path

In this section, we briefly review the work of Dumas *et al.* (1990) who have already shown how an optimal service schedule and its cost can be determined for a fixed path. For the sake of simplicity, we assume that the path is given as $P = (0, 1, 2, \ldots, m)$ (otherwise re-index the vertices). For any $i, j \in [0 : m]$ with $i \leq j$, we define the *(accumulated) travel time* from vertex i to vertex j along path P as $T_{i,j} := \sum_{h=i+1}^{j} \tau_{h-1,h}$.

The algorithm of Dumas *et al.* (1990) first computes the individual minimum $t_i^* \in [e_i, \ell_i]$ of the function $f_i(t)$ for each $i \in [0 : m]$, by relaxing (1b) in the service scheduling problem (2). Two consecutive minima t_{i-1}^* and t_i^* are said to be *in conflict* if $t_{i-1}^* + T_{i-1,i} > t_i^*$. Then, all conflicts are resolved by *aggregation*.

Specifically, the aggregation of two arbitrary consecutive positions $i-1$ and i is done by introducing an equality constraint $t_{i-1} + T_{i-1,i} = t_i$. For this aggregation over $I = [i-1 : i]$, the resulting optimization problem becomes

$$\min f_{i-1}(t_i - T_{i-1,i}) + f_i(t_i) \quad \text{s.t.} \quad t_i \in [e_I, \ell_I]$$

with $[e_I, \ell_I] := [e_{i-1} + T_{i-1,i}, \ell_{i-1} + T_{i-1,i}] \cap [e_i, \ell_i]$, which is again a one-dimensional convex optimization problem over an interval.

To simplify the description, assume that aggregation has already produced $q \geq 1$ disjoint integer intervals $I_1, I_2, \ldots, I_q$ of the form $I_h = [u_h : v_h]$ for $h \in \{1, \ldots, q\}$ such that $\bigcup_{h=1}^{q} I_h = [0 : m]$. In the following, each integer interval I_h is called a *block* (the survey by Vidal *et al.* (2015) uses the same terminology), and the concatenation of blocks $\mathbf{B} = I_1|I_2|\ldots|I_q$ is called a *block structure* for $[0 : m]$.

Each block $I = [u : v]$ reduces the $(v - u + 1)$-dimensional subproblem, $\min \sum_{i=v}^{u} f_i(t_i)$, over $t_u, t_{u+1}, \ldots, t_v$ by introducing equalities $t_u + T_{u,u+1} = t_{u+1}, t_{u+1} + T_{u+1,u+2} = t_{u+2}, \ldots, t_{v-1} + T_{v-1,v} = t_v$ into the following one-dimensional convex optimization problem, expressed in terms of the single variable $t = t_v$:

$$\min f_I(t) := \sum_{i=u}^{v} f_i(t - T_{i,v}) \quad \text{s.t.} \quad t \in [e_I, \ell_I] := \bigcap_{i=u}^{v} [e_i + T_{i,v}, \ell_i + T_{i,v}]$$

We denote its minimum by t_I^*. In each iteration of the algorithm proposed by Dumas *et al.* (1990), one aggregation of consecutive blocks I_{h-1} and I_h is performed if the blocks are in conflict, i.e., their minima $t_{I_{h-1}}^*$ and $t_{I_h}^*$ satisfy $t_{I_{h-1}}^* + T_{v_{h-1},v_h} > t_{I_h}^*$.

In conclusion, the algorithm of Dumas *et al.* (1990) starts with the initial block structure $[0]|[1]|[2]|\ldots|[m]$ (we use the shorthand notation $[i]$ for $[i : i] = \{i\}$) and iteratively resolves conflicts between consecutive blocks I_{h-1} and I_h by aggregating and replacing the two by a single block $(I_{h-1} \cup I_h)$, until no more conflict exists. We call this final block structure an *optimal block structure* for path P. Since the number of aggregation steps is bounded from above by m, this procedure involves solving between $m + 1$ and $2m + 1$ one-dimensional convex optimization problems.

Example 1. *We consider the following path $P = (0, 1, 2, 3, 4, 5, 6)$ with the corresponding time windows and inconvenience cost functions $f_i(t)$'s associated with the seven vertices $i \in [0 : 6]$*

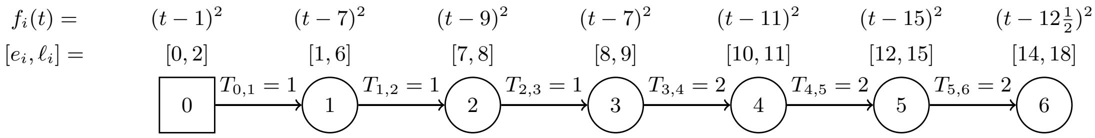


Figure 1: Path $P = (0,1,2,3,4,5,6)$ with time windows, travel times, and inconvenience cost functions.

and travel times $T_{i-1,i}$ (for $i \in [1:6]$) between them as depicted in Figure 1. The algorithm of Dumas et al. *(1990) starts with block structure $[0]||[1]||[2]||[3]||[4]||[5]||[6]$ and optima $t^*_{[0]} = 1,\quad t^*_{[1]} = 6,\quad t^*_{[2]} = 8,\quad t^*_{[3]} = 8,\quad t^*_{[4]} = 11,\quad t^*_{[5]} = 15$, and $t^*_{[6]} = 14$. The only initial conflicts of the form $t^*_{[i-1]} + T_{i-1,i} > t^*_{[i]}$ exist at positions $i = 3$ and $i = 6$. Aggregation of $[2]$ and $[3]$ into $[2:3]$ produces a new inconvenience cost function $f_{[2:3]}(t) = 2(t - 8\frac{1}{2})^2 + 4\frac{1}{2}$ defined over the time window $[7+1, 8+1] \cap [8,9] = [8,9]$. The minimum of $f_{[2:3]}(t)$ on block $[2:3]$ is attained at $t^*_{[2:3]} = 8\frac{1}{2}$.*

Similarly, aggregation of $[5]$ and $[6]$ into $[5:6]$ produces a new inconvenience cost function $f_{[5,6]}(t) = 2(t - 14\frac{3}{4})^2 + 10\frac{1}{8}$ defined over the time window $[12+2, 15+2] \cap [14,18] = [14,17]$, where the minimum value is attained at $t^*_{[5:6]} = 14\frac{3}{4}$. After the two conflicts are resolved via aggregation,

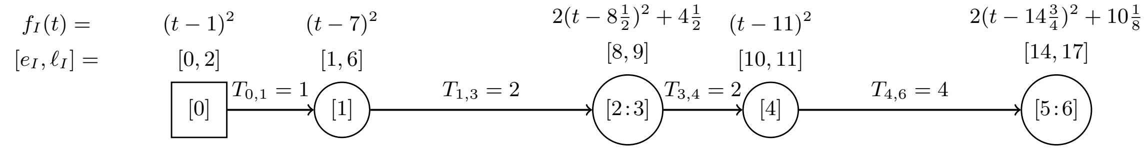


Figure 2: Situation after aggregation of blocks [2] and [3] into [2 : 3] as well as blocks [5] and [6] into [5 : 6] for the path P depicted in Figure 1.

the new situation is depicted in Figure 2. The optimal service times are $t^*_{[0]} = 1,\quad t^*_{[1]} = 6,\quad t^*_{[2:3]} = 8\frac{1}{2},\quad t^*_{[4]} = 11$, and $t^*_{[5:6]} = 14\frac{3}{4}$.

One new conflict between $t^*_{[4]} = 11$ and $t^*_{[5:6]} = 14\frac{3}{4}$ (due to that $t^*_{[4]} + T_{46} = 11 + 4 = 15 > t^*_{[5:6]} = 14\frac{3}{4}$) must again be resolved. The aggregation of $[4]$ and $[5:6]$ into $[4:6]$ leads to an optimal block structure $[0]||[1]||[2:3]||[4:6]$ for path P. The resulting inconvenience cost function of block $[4:6]$ is $f_{[4:6]}(t) = 3(t - 14\frac{5}{6})^2 + 10\frac{1}{6}$ with minimum $t^*_{[4:6]} = 14\frac{5}{6}$. No more conflict exists. Finally, optimal service times for P result from $t^*_0 = 1, t^*_1 = 6, t^*_3 = t^*_{[2:3]} = 8\frac{1}{2}$, and $t^*_6 = t^*_{[4:6]} = 14\frac{5}{6}$ and the three introduced equality constraints $t^*_2 = t^*_3 - 1$, $t^*_4 = t^*_6 - 4$, and $t^*_5 = t^*_6 - 2$. The optimal schedule is given by $(t^*_i) = (1, 6, 7\frac{1}{2}, 8\frac{1}{2}, 10\frac{5}{6}, 12\frac{5}{6}, 14\frac{5}{6})$. □

The following additional results for the service scheduling problem can be shown easily:

- (i) If conflicts are resolved by going through vertices one by one in an increasing order of their indices, there are only $\mathcal{O}(m)$ tests for conflicts to be performed (see also Dumas *et al.*, 1990, p. 149).
- (ii) When the $f_i(t)$'s are strictly convex inconvenience cost functions, the optimal block structure of a path is unique. For the sake of simplicity, strict convexity is assumed in the rest of this paper.

(iii) For quadratic convex inconvenience cost functions, the aggregation and optimization steps can be performed in $\mathcal{O}(1)$ so that the worst-case running time of the algorithm is $\mathcal{O}(m)$. Our computational analysis on the VRPTW-CNC will use quadratic functions.

3. Extensions of a Fixed Path

In the course of a dynamic-programming based labeling algorithm, paths are extended vertex by vertex. Assume that $P = (0, 1, \ldots, m)$ is a path and that one extension (out of several possible extensions) is towards vertex $m+1$. The earliest possible service time at the last vertex m of path P, in the following denoted by α_P, is not affected by any extension; It is straightforward to compute α_P using an as-early-as-possible schedule. However, the latest possible schedule time at vertex m may decrease due to the extension, because it is bounded from above by $\min\{\ell_m, \ell_{m+1} - T_{m,m+1}\} \le \ell_m$. Other and further extensions can decrease the latest possible schedule time at m in different ways.

The consequence of this observation is the following: When it comes to a comparison between different paths (to see if one can be dominated), it is not sufficient to only know its minimum cost f_P when the full time window $[e_m, \ell_m]$ of the schedule time at vertex m can be exploited for path $P = (0, 1, \ldots, m)$. We must also be able to describe the optimal service schedule and its cost for any restricted time window $[\alpha_P, x]$, for any latest possible schedule time at the last vertex $x \in [\alpha_P, \ell_m]$, as a result of a potential extension. We want to determine how the optimal block structure, schedule, and its cost vary depending on the value of x. We denote the latter the *path inconvenience cost function* of P by $f_P(x)$. Specifically, $f_P(x)$ is defined as

$$f_P(x) := \min\left\{\sum_{i=0}^{m} f_i(t_i) \mid t_m \in [\alpha_P, x] \text{ and } t_i + T_{i,i+1} \le t_{i+1}, t_i \in [e_i, \ell_i], \forall i \in [0 : m-1]\right\}.$$

This is sharply different from the dominance check for the standard VRPTW problem, where both cost and schedule time are fixed for a given path, i.e., by serving each customer as early as possible (since the cost does not depend on the schedule time).

3.1. Variation of the Latest Possible Schedule Time at the Last Vertex

This section provides results for a fixed path $P = (0, 1, \ldots, m)$ and variable latest schedule time x at its last vertex m. We first introduce a few notations related to the optimal block structure of a path. Let $\mathbf{B}^P(x)$ denote the optimal block structure over a path $P = (0, 1, \ldots, m)$, assuming that the service time window at the last vertex m is $[\alpha_P, x]$. We assume that $\mathbf{B}^P(\ell_m) = I_1|I_2|\ldots|I_q$ for some $q \in [1 : m]$ with $I_h = [u_h : v_h]$ for $h \in [1 : q]$.

Proposition 1. *Given a path $P = (0, 1, \ldots, m)$,*

(i) for each $x \in [\alpha_P, \ell_m]$, there exists an index $r(x) \in [1 : q]$ such that the block structure is $\mathbf{B}^P(x) = I_1|I_2|\ldots|I_{r(x)-1}|(I_{r(x)} \cup I_{r(x)+1} \cup \cdots \cup I_q)$;

(ii) the function $r : [\alpha_P, \ell_m] \to [1 : q]$ is nondecreasing as x increases;

(iii) all possible optimal block structures of $B^P(x)$ for $x \in [\alpha_P, \ell_m]$ are:

$$\begin{array}{l}
I_1|I_2|\ldots|I_{p-1}|(I_p \cup I_{p+1} \cup \ldots \cup I_{q-1} \cup I_q) \\
I_1|I_2|\ldots|I_{p-1}|\ I_p|\ \ (I_{p+1} \cup \ldots \cup I_{q-1} \cup I_q) \\
\quad\vdots \qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\
I_1|I_2|\ldots|I_{p-1}|\ I_p|I_{p+1}| \quad \ldots \quad |(I_{q-1} \cup I_q) \\
I_1|I_2|\ldots|I_{p-1}|\ I_p|I_{p+1}| \quad \ldots \quad |\ I_{q-1}|\ I_q
\end{array}$$

where $p := r(\alpha_P)$*;*

(iv) if the block structure $I_1|I_2|\ldots|I_{k-1}|(I_k \cup I_{k+1} \cup \ldots \cup I_{q-1} \cup I_q)$ *is optimal for some* $x \in [\alpha_P, \ell_m]$*, then* x *lies in the interval* $\mathbf{S}_k$ *defined as follows:*

(a) if $p < q$ *and* $k = p$*, then* $\mathbf{S}_k := [\alpha_P, t^*_{I_p} + T_{v_p,m}]$*;*

(b) if $p < q$ *and* $p < k < q$*, then* $\mathbf{S}_k := [t^*_{I_{k-1}} + T_{v_{k-1},m},\ t^*_{I_k} + T_{v_k,m}]$*;*

(c) if $p < q$ *and* $k = q$*, then* $\mathbf{S}_k := [t^*_{I_{q-1}} + T_{v_{q-1},m},\ \ell_m]$*;*

(d) if $p = q$*, then* $\mathbf{S}_k := [\alpha_P, \ell_m]$*.*

The intervals $\mathbf{S}_p, \mathbf{S}_{p+1}, \ldots, \mathbf{S}_q$ *form a partition of the time window of vertex* m*, i.e.,* $[\alpha_P, \ell_m] = \mathbf{S}_p \cup \mathbf{S}_{p+1} \cup \ldots \cup \mathbf{S}_{q-1} \cup \mathbf{S}_q$ *and* $\mathbf{S}_k \cap \mathbf{S}_l = \varnothing$ *for* $k \neq l, k, l \in [p : q]$*.*

(v) The path inconvenience cost function has the following form

$$f_P(x) \;=\; \sum_{h=1}^{p-1} f_{I_h}(t^*_{I_h}) + \sum_{h=p}^{r(x)-1} f_{I_h}(t^*_{I_h}) \quad + \quad f_{(I_{r(x)} \cup \ldots \cup I_q)}\left(\min\{x, t^*_{(I_{r(x)} \cup \ldots \cup I_q)}\}\right), \qquad (4)$$

where $x \in \mathbf{S}_{r(x)}$ *with* $r(x) \in [p : q]$*. Note that the first term is a constant, and the second term depends on* x *via* $r(x)$*. Function* $f_P(x)$ *is a monotonically nonincreasing function of* x*.*

Proof: Note first that all statements are trivial if $p = q$, i.e., $\mathbf{B}^P(x) = I_1|I_2|\ldots|I_q$ for any $x \in [\alpha_P, \ell_m]$. We can therefore assume $p < q$ in the following.

All statements are consequences of the algorithm of Dumas *et al.* (1990). Assume that the optimal block structure $\mathbf{B}^P(\ell_m) = I_1|I_2|\ldots|I_q$ has been computed with this algorithm. Then, there exists no conflict between the block minima $t^*_{I_1}, t^*_{I_2}, \ldots, t^*_{I_q}$. Note that for each block $I_h = [u_h : v_h], h \in [1 : q]$, the minimum $t^*_{I_h}$ corresponds to the schedule time at the last vertex v_h. The addition of $T_{v_h,m}$ to these minima produces *shifted minima* $t^*_{I_h} + T_{v_h,m}$ that all correspond to vertex m. By the definition of the optimal block structure, we know that the following inequalities hold

$$t^*_{I_1} + T_{v_1,m} < t^*_{I_2} + T_{v_2,m} < \ldots < t^*_{I_{q-1}} + T_{v_{q-1},m} < t^*_{I_q} + \underbrace{T_{v_q,m}}_{=0} = t^*_{I_q}.$$

Therefore, the stated bounds (in part (iv)) of intervals $\mathbf{S}_k$ for $k \in [1 : q]$ are increasing as k increases. Moreover, the intervals $\mathbf{S}_k$ are non-empty.

Note first that when the upper bound of the time window at the last vertex m decreases starting from $x = \ell_m$, only the block minimum $t^*_{I_q}$ may change. Indeed, with a decreasing x, $t^*_{I_q}$ is non-increasing and it changes continuously. Consequently, the first conflict with x decreasing occurs between the last two blocks I_{q-1} and I_q. Then the algorithm of Dumas *et al.* aggregates these two blocks I_{q-1} and I_q into $(I_{q-1} \cup I_q)$. This happens exactly when x reaches the bound $t^*_{I_{q-1}} + T_{v_{q-1},m}$, which is the lower bound of $\mathbf{S}_q$. The same argument can be applied recursively to $I_1|I_2|\ldots|I_{q-2}|(I_{q-1} \cup I_q)$, which shows the correctness of parts (i)–(iii).

Part (iv) is also a consequence of the above observations. As before, case (d) is trivial and we can therefore assume $p < q$ in the following. Let $\mathbf{B}_k = I_1|I_2|\ldots|I_{k-1}|(I_k \cup I_{k+1} \cup \ldots \cup I_{q-1} \cup I_q)$ be an optimal block structure. Case (c) means that decreasing x must not produce any new conflict, which holds if $x \geq t^*_{I_{q-1}} + T_{v_{q-1},m}$. This leads to the interval $\mathbf{S}_q = [t^*_{I_{q-1}} + T_{v_{q-1},m}, \ell_m]$ as stated in (c).

Now cases (a) and (b) occur when decreasing x creates some conflicts. They can be proven by induction with respect to k, i.e., knowing that the interval $\mathbf{S}_k$ equals $[t^*_{I_{k-1}} + T_{v_{k-1},m}, t^*_{I_k} + T_{v_k,m}]$, we must show that $\mathbf{S}_{k-1}$ has the form of case (a) or (b). Obviously, the upper bound of $\mathbf{S}_{k-1}$ must coincide with the lower bound of $\mathbf{S}_k$. Hence we know that $\mathbf{S}_{k-1} = [l, t^*_{I_{k-1}} + T_{v_{k-1},m}]$ for some value l (this is the upper bound stated in case (a) and (b)). Case (b) results if $t^*_{I_{k-2}} + T_{v_{k-2},m} > \alpha_P$ so that another conflict between I_{k-2} and $(I_{k-1} \cup \dots \cup I_q)$ can be produced by decreasing x, i.e., $l = t^*_{I_{k-2}} + T_{v_{k-2},m}$. Otherwise, case (a) results.

Finally note that equation (4) of part (v) is a direct consequence of (iv), and the monotonicity of function $f_P(x)$ is clear from its definition. □

Example 2. *(continued from Example 1) We now consider a subpath $P = (0, 1, 2, 3, 4, 5)$ of the path analyzed in Example 1. The calculations in Example 1 show that P has the optimal block structure* $[0]||[1]||[2 : 3]||[4]||[5]$ *shown in Figure 3. The optimal service times are* $(t^*_i) = (1, 6, 7\frac{1}{2}, 8\frac{1}{2}, 11, 15)$.

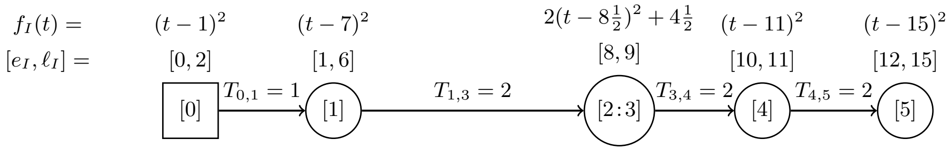


Figure 3: Result of the algorithm of Dumas *et al.* for path $P = (0, 1, 2, 3, 4, 5)$.

We now decrease the upper bound $x = 15$ of the time window of vertex 5. As long as $x \geq 13$, there is no additional conflict created (due to $t^*_{[4]} + T_{45} = 11 + 2 = 13$). Therefore, $f_P(x)$ results from assigning the optimal service times to the first four blocks of $I_1|I_2|I_3|I_4|I_5 = [0]||[1]||[2:3]||[4]||[5]$ and choosing a best value t^*_5 in $[12, x]$ for $f_{[5]}$, which is x. The former evaluates to $(1-1)^2 + (6-7)^2 + (2(8\frac{1}{2} - 8\frac{1}{2})^2 + 4\frac{1}{2}) + (11-11)^2 = 5\frac{1}{2}$, and the latter adds $(x-15)^2$ so that $f_P(x) = (x-15)^2 + 5\frac{1}{2}$ for $x \in \mathbf{S}_5 = [13, 15]$.

We now resolve the first conflict that occurs when x becomes smaller than 13. In this case, blocks I_4 and I_5 must be aggregated into $[4:5]$. The resulting inconvenience cost function on this block is $(t - T_{4,5} - 11)^2 + (t-15)^2 = 2(t-14)^2 + 2$ with an optimum $t^*_{[4:5]} = x$ for x in the interval $\mathbf{S}_4 = [t^*_{[2:3]} + T_{3,5}, 13] = [8\frac{1}{2} + 4, 13] = [12\frac{1}{2}, 13]$. Here, the lower bound of the interval is again computed by considering the next (second) conflict. The resulting function $f_P(x)$ is $\left[(1-1)^2 + (6-7)^2 + (2(8\frac{1}{2} - 8\frac{1}{2})^2 + 4\frac{1}{2})\right] + (2(x-14)^2 + 2) = 2(x-14)^2 + 7\frac{1}{2}$.

Finally, when x becomes smaller than $12\frac{1}{2}$, blocks I_3, I_4, and I_5 must be aggregated into $[3:5]$. A similar calculation shows that on the interval $\mathbf{S}_3 = [12, 12\frac{1}{2}]$ the path inconvenience cost function becomes $f_P(x) = 4(x - 13\frac{1}{4})^2 + 9\frac{3}{4}$. Here, the lower bound 12 for the interval does not result from a conflict but is the earliest service time α_P at vertex 5. Hence, the entire function $f_P(x)$ is known for all possible $x \in [12, 15]$. The function is depicted in Figure 4. □

In summary, each path $P = (0, 1, \dots, m)$ has $q - p + 1$ possible optimal block structures with p and q defined in Proposition 1. Each of these block structures has an associated interval $\mathbf{S}_k$ with $k \in [p : q]$. The interval $[\alpha_P, \ell_m]$ of all possible schedule times at the last vertex m of path P is partitioned into $q - p + 1$ intervals, one for each optimal block structure.

3.2. Optimal Block Structures of an Extended Path

We consider a path $P = (0, 1, \dots, m)$ with the optimal block structure $\mathbf{B}^P(\ell_m) = I_1|I_2|\dots|I_q$. We now extend P towards an arbitrary vertex $m+1$. We assume that $\alpha_P + T_{m,m+1} \leq \ell_{m+1}$ holds

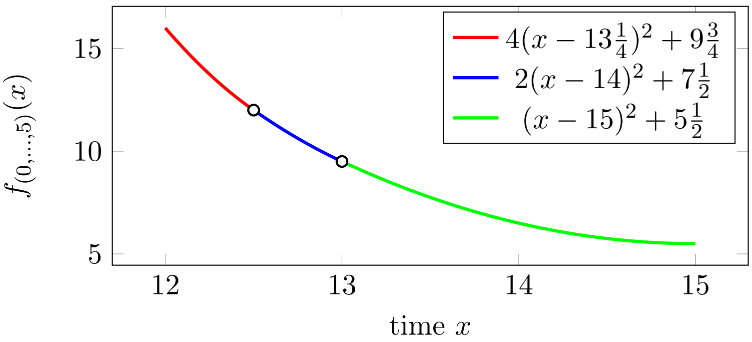


Figure 4: Function $f_P(x)$ for the path $(0, 1, 2, 3, 4, 5)$ from Example 1 defined piecewise on the three intervals $\mathbf{S}_3 = [12, 12\frac{1}{2}]$, $\mathbf{S}_4 = [12\frac{1}{2}, 13]$, and $\mathbf{S}_5 = [13, 15]$.

as otherwise $P' = (P, m+1)$ is infeasible due to a time window violation at vertex $m+1$. Instead of running the algorithm of Dumas *et al.* on P' from scratch, we describe how the new intervals $\mathbf{S}_{p'}, \ldots, \mathbf{S}_{q'}$ (and their corresponding optimal block structures) of P' can be obtained from intervals $\mathbf{S}_p, \ldots, \mathbf{S}_q$ of P. This is the key to the efficient label extension of our proposed BCP algorithm, which will be introduced in Section 4.

First, Proposition 2 given below shows that the first interval $\mathbf{S}_{p'}$ of P' can be obtained by simply looking up which interval among $\mathbf{S}_p, \ldots, \mathbf{S}_q$ of P contains $\alpha_{P'} - T_{m,m+1}$.

Proposition 2. *Given a path $P = (0, 1, \ldots, m)$ with the optimal block structure $\mathbf{B}^P(\ell_m) = I_1|I_2|\ldots|I_q$ and intervals $\mathbf{S}_p, \ldots, \mathbf{S}_q$ of $f_P(x)$, assume that $P' = (P, m+1)$ is feasible regarding the time window constraints. Then, the index $p' = r(\alpha_{P'})$ is given by*

(i) the index k of interval $\mathbf{S}_k$ that contains $\alpha_{P'} - T_{m,m+1}$, if $\alpha_{P'} - T_{m,m+1} < \ell_m$ (in case of a tie, i.e., if $\alpha_{P'} - T_{m,m+1}$ corresponds to a boundary shared by two intervals, the index k is chosen as the larger one);

(ii) the value $q+1$, if $\alpha_{P'} - T_{m,m+1} \geq \ell_m$; (in this case, P' has only one optimal block structure $\mathbf{B}^{P'}(x) = I_1|I_2|\ldots|I_q|[m+1]$ for any $x \in [\alpha_{P'}, \ell_{m+1}]$).

Proof: Note that in case (ii) any feasible service time at vertex $m+1$ cannot make a conflict with block I_q so that $\mathbf{B}^{P'}(x) = I_1|I_2|\ldots|I_q|[m+1]$ for any $x \in [\alpha_{P'}, \ell_{m+1}]$. In case (i), the time $\alpha_{P'} - T_{m,m+1}$ is the latest possible schedule time at vertex m if schedule time at $m+1$ is fixed to $\alpha_{P'}$ (we consider the latest possible schedule time at vertex m, since the path inconvenience cost function is monotonically nonincreasing). Therefore, the interval with $\alpha_{P'} - T_{m,m+1} \in \mathbf{S}_k$ (with the tie break rule specified in case (i)) implies that blocks $I_k|\ldots|I_q|[m+1]$ are aggregated into $(I_k \cup \ldots \cup I_q \cup [m+1])$. We then have $p' = r(\alpha_{P'}) = k$. □

We now describe how to obtain the last interval $\mathbf{S}_{q'}$ and its associated optimal block structure of the extended path $P' = (P, m+1)$, i.e., the one corresponding to its latest possible schedule time $x = \ell_{m+1}$, at the last vertex $m+1$. If $t^*_{I_q} + T_{v_q,m+1} \leq t^*_{m+1}$ then the block structure $I_1|I_2|\ldots|I_q|[m+1]$ is optimal for $(P, m+1)$ and $x = \ell_{m+1}$, because there is no conflict. Otherwise, the conflict $t^*_{I_q} + T_{v_q,m+1} > t^*_{m+1}$ between I_q and $m+1$ has to be resolved by aggregation. If $q = 1$, the optimal block structure for $(P, m+1)$ and $x = \ell_{m+1}$ is simply $I_1 \cup [m+1]$. If $q > 1$, the aggregated block $I_q \cup [m+1]$ with optimum $t^*_{I_q \cup [m+1]}$ may cause another conflict with block

I_{q-1}, since the block minimum is non-increasing after an aggregation (see the proof of Proposition 1), i.e., $t^*_{I_q \cup [m+1]} \le t^*_{I_q}$. In this case, an aggregation into the block $I_{q-1} \cup I_q \cup [m+1]$ is required and the procedure has to be iterated. The procedure terminates with the optimal block structure $I_1|\dots|I_{k-1}|(I_k \cup \dots \cup I_q \cup [m+1])$, where k satisfies the following conditions:

$$t^*_{I_{k-1}} + T_{v_{k-1},m+1} \le t^*_{(I_k \cup \dots \cup I_q \cup [m+1])} \quad \text{and} \quad t^*_{I_k} + T_{v_k,m+1} > t^*_{(I_{k+1} \cup \dots \cup I_q \cup [m+1])}. \tag{5}$$

All of the above observations are summarized in the following proposition.

Proposition 3. *Given a path $P = (0, 1, \dots, m)$ with the optimal block structure $\mathbf{B}^P(\ell_m) = I_1|I_2|\dots|I_q$, assume that the extension of P to $P' = (P, m+1)$ is feasible regarding the time window constraints. Then the optimal block structure $\mathbf{B}^{P'}(\ell_{m+1})$ is given by one of the following two cases:*

(i) $\mathbf{B} = I_1|I_2\dots|I_q|[m+1]$ *("A new block is created.")*
if $t^*_{I_q} + T_{m,m+1} \le t^*_{m+1}$;

(ii) $\mathbf{B} = I_1|\dots|I_{k-1}|(I_k \cup \dots \cup I_q \cup [m+1])$ *("The last few blocks become aggregated.")*
if $t^*_{I_q} + T_{m,m+1} > t^*_{m+1}$ *with k defined in conditions* (5).

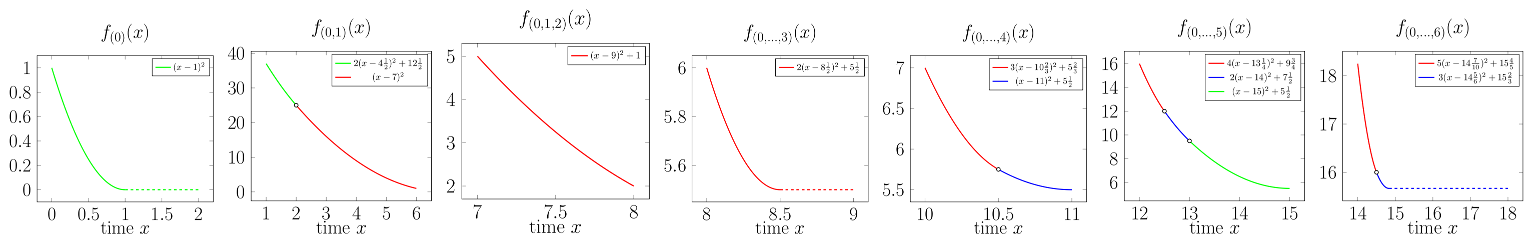
Note that unlike the case for p' shown in Proposition 2, in general we cannot obtain the index k defined in conditions (5) by simply looking up the interval of P that contains $t^*_{m+1} - T_{m,m+1}$ or $\ell_{m+1} - T_{m,m+1}$. Indeed, the index of the interval for the former might be too small, while the index for the latter might be too large, as shown in the following example.

Example 3. *(continued from Examples 1 and 2) For all subpaths P of* $(0, 1, 2, 3, 4, 5, 6)$*, Figure 5 shows the optimal block structure (divided into the fixed part $I_1|\dots|I_{p-1}$ and the part $I_p|\dots|I_q$ that may be aggregated). For each possible aggregated part of $I_p|\dots|I_q$, the figure also shows the resulting aggregated block $(I_k \cup \dots \cup I_q)$ with $p \le k \le q$, the definition of the associated piece of $f_P(x)$, and the interval $\mathbf{S}_k$.*

Consider the extension from $P = (0, 1, 2, 3, 4, 5)$ to $P' = (P, 6)$, where the index k defined in conditions (5) is 4, which is neither identical to $p = 3$ (the index of the interval S_3 that contains $t^*_6 - T_{5,6}$) nor identical to $q = 5$ (the index of the last interval S_5).

4. Branch-Cut-and-Price Algorithm

For solving the linear programming relaxation of the set-partitioning formulation (3), we use a column-generation algorithm (Desaulniers *et al.*, 2005). Starting with a subset $\Omega' \subset \Omega$ of feasible routes, the linear programming relaxation of (3) defined over Ω' is denoted as the *restricted master program* (RMP). The column-generation algorithm alternates between the re-optimization of the RMP and the solution of the column-generation pricing problem. The latter adds negative reduced-cost variables to the RMP, if one exists; otherwise the linear programming relaxation is solved to optimality. For simplicity, we assume that only elementary routes are considered in the pricing problem unless otherwise stated. The kernel of our approach, the dynamic-programming based labeling algorithm to solve the pricing problem for column generation, is presented in Section 4.1. In Section 4.2, we explain how to strengthen the RMP bounds using valid inequalities and our branching rules.



Time window	$[0,2]$	$[1,6]$	$[7,8]$	$[8,9]$	$[10,11]$	$[12,15]$	$[14,18]$
Inconv. fnct.	$f_0(t)=(t-1)^2$	$f_1(t)=(t-7)^2$	$f_2(t)=(t-9)^2$	$f_3(t)=(t-7)^2$	$f_4(t)=(t-11)^2$	$f_5(t)=(t-15)^2$	$f_6(t)=(t-12\frac{1}{2})^2$
Optimum	$t_0^*=1$	$t_1^*=6$	$t_2^*=8$	$t_3^*=8$	$t_4^*=11$	$t_5^*=15$	$t_6^*=14$
Path P	(0)	$(0,1)$	$(0,1,2)$	$(0,\dots,3)$	$(0,\dots,4)$	$(0,\dots,5)$	$(0,\dots,6)$
p	1	1	3	3	3	3	3
q	1	2	3	3	4	5	4
$I_1\vert\dots\vert I_{p_0-1}$			$[0]\Vert[1]$	$[0]\Vert[1]$	$[0]\Vert[1]$	$[0]\Vert[1]$	$[0]\Vert[1]$
$I_{p_0}\vert\dots\vert I_q$	$[0]$	$[0]\Vert[1]$	$[2]$	$[2:3]$	$[2:3]\Vert[4]$	$[2:3]\Vert[4]\Vert[5]$	$[2:3]\Vert[4:6]$
1	$[0]$ $(x-1)^2$ $\mathbf{S}_1=[0,2]$	$[0:1]$ $2(x-4\frac{1}{2})^2+12\frac{1}{2}$ $\mathbf{S}_1=[1,2]$					
2		$[1]$ $(x-7)^2$ $\mathbf{S}_2=[2,6]$					
3			$[2]$ $(x-9)^2+1$ $\mathbf{S}_3=[7,8]$	$[2:3]$ $2(x-8\frac{1}{2})^2+5\frac{1}{2}$ $\mathbf{S}_3=[8,9]$	$[2:4]$ $3(x-10\frac{2}{3})^2+5\frac{2}{3}$ $\mathbf{S}_3=[10,10\frac{1}{2}]$	$[2:5]$ $4(x-13\frac{1}{4})^2+9\frac{3}{4}$ $\mathbf{S}_3=[12,12\frac{1}{2}]$	$[2:6]$ $5(x-14\frac{7}{10})^2+15\frac{4}{5}$ $\mathbf{S}_3=[14,14\frac{1}{2}]$
4					$[4]$ $(x-11)^2+5\frac{1}{2}$ $\mathbf{S}_4=[10\frac{1}{2},11]$	$[4:5]$ $2(x-14\frac{1}{4})^2+7\frac{1}{2}$ $\mathbf{S}_4=[12\frac{1}{2},13]$	$[4:6]$ $3(x-14\frac{5}{6})^2+15\frac{2}{3}$ $\mathbf{S}_4=[14\frac{1}{2},18]$
5						$[5:5]$ $(x-15)^2+5\frac{1}{2}$ $\mathbf{S}_5=[13,15]$	

Figure 5: Path inconvenience cost function $f_P(x)$ for the path $P=(0,1,2,3,4,5,6)$ and all of its subpaths. Each column refers to a subpath. Each entry shows the last block $\mathbf{S}_k$ for $k\in[p:q]$, the (piecewise) definition of the path inconvience function $f_P(x)$, and the interval $\mathbf{S}_k$ on which the respective piece is defined.

4.1. Column Generation.

Let $(\pi_i)_{i\in V}$ be the dual prices of the partitioning constraints (3b) and let μ be the dual price of the fleet size constraint (3c) of the RMP. For the sake of compactness, we define $\pi_0 := \pi_{n+1} := \mu$. The elementary SPPRC pricing problem of the VRPTW-CNC is defined on the digraph (V, A) that defines the instance. We define arc reduced costs $\tilde{c}_{ij}$ as $c_{ij} - (\pi_i + \pi_j)/2$ for all $(i, j) \in A$. Recall that the cost of a route P is $c_P = \sum_{(i,j)\in A(P)} c_{ij} + f_P(\ell_{n+1})$, hence, the reduced cost of P is $\tilde{c}_P = c_P - \mu - \sum_{i\in V(P)} \pi_i = \sum_{(i,j)\in A(P)} \tilde{c}_{ij} + f_P(\ell_{n+1})$, where $V(P)$ and $A(P)$ denote the set of vertices and arcs of path P, respectively.

4.1.1. Definition of Labels.

A label is a representation of a path $P = (0, \ldots, m)$ that has the potential to finally become a feasible 0-$(n+1)$-path with a negative reduced cost. In the VRPTW-CNC, a label has the following attributes:

$R^{cost}(P)$**:** Accumulated reduced routing costs $\sum_{(i,j)\in A(P)} \tilde{c}_{ij}$;

$R^{load}(P)$**:** Sum of demands covered on the path $\sum_{i\in V(P)} q_i$;

$R^{cust,n}(P)$**:** An indicator of whether customer n can be visited in any extension of P, for each customer $n \in N$.

Moreover, the label stores the path inconvenience cost function $f_P(x)$ implicitly, using the following additional attributes:

p**:** Index of the first block in $\mathbf{B}^P(\ell_m)$ that may become aggregated;

q**:** Index of the last block in $\mathbf{B}^P(\ell_m)$;

$f^{const}(P)$**:** The constant part of the path inconvenience cost function $f_P(x)$ that does not depend on x, given by $\sum_{h=1}^{p-1} f_{I_h}(t^*_{I_h})$, cf. the first term in (4);

$s^*_k(P)$**:** Shifted optima defined as $t^*_{I_k} + T_{v_k,m}$ for blocks I_k, $k \in [p : q]$, cf. LHS of (5);

$w_k(P)$**:** Bounds of intervals $\mathbf{S}_k = [w_{k-1}, w_k]$ for $k \in [p : q]$ (recall that interval $\mathbf{S}_k$ corresponds to block structure $I_p|\ldots|I_{k-1}|(I_k \cup \ldots I_q))$; Note that there is no need to separately store α_P because $w_{p-1} = \alpha_P$;

$\Sigma^*_k(P)$**:** Accumulated optimal values defined as $\sum_{h=p}^{k-1} f_{I_h}(t^*_{I_h})$ for the blocks from I_p to I_{k-1} for $k \in [p : q]$, cf. the second term in (4);

$f_{(I_k\cup\ldots\cup I_q)}(x)$**:** A representation of the aggregated inconvenience cost functions for $k \in [p : q]$ for a possible last block $(I_k \cup \ldots \cup I_q)$, cf. the third term of (4). Note that the third term of (4) is not directly $f_{(I_k\cup\ldots\cup I_q)}(x)$ but uses the function $f_{(I_k\cup\ldots\cup I_q)}(\cdot)$ with the argument $\min\{x, t^*_{(I_k\cup\ldots\cup I_q)}\}$ instead of x.

For the sake of convenience, we define the *domain of* f_P as $dom(f_P) = [\alpha_P, \ell_m] = [w_{p-1}, w_q]$.

4.1.2. Resource Extension Functions.

The initial label at the starting vertex 0 has attributes $p = q = 1$, $w_0 = e_0$, $w_1 = \ell_0$, the (shifted) optimum t^*_0 (for $k = 1$), accumulated optimal value 0 (for $k = 1$), and (aggregated) inconvenience cost function f_0 (for $k = 1$). Next, we show how a path $P = (0, \ldots, m)$ is extended to a path $P' = (P, m + 1)$.

With the help of $\alpha_P = w_{p-1}$ and Proposition 2 we can directly compute

$$\alpha_{P'} = \max\{e_{m+1}, \alpha_P + T_{m,m+1}\} = \max\{e_{m+1}, w_{p-1} + T_{m,m+1}\} \tag{6a}$$

$$p' = r(\alpha_{P'}) = \begin{cases} q+1, & \text{if } \alpha_{P'} - T_{m,m+1} \geq \ell_m \\ k, & \text{where } [w_{k-1}, w_k] \text{ contains } \alpha_{P'} - T_{m,m+1}\text{, see Prop. 2.} \end{cases} \tag{6b}$$

A prerequisite of the following steps is the computation of all new shifted optima (we do this for all $k \in [p:q]$ because the new range p' to $q' = r(\ell_{m+1})$ is not yet known), which result from the addition of $T_{m,m+1}$ and are given by

$$s_k^*(P') = s_k^*(P) + T_{m,m+1}. \tag{6c}$$

These are the values on the LHS of conditions (5).

Next, in order to evaluate conditions (5), we compute the aggregated inconvenience cost functions $f_{(I_k \cup \cdots \cup I_q \cup \{m+1\})}$ and their optima (i.e., the values on the RHS of conditions (5)). More precisely, this process is started with $k = q+1$, in each iteration the respective conditions (5) of Proposition 3 is tested and k is decreased afterwards. At the beginning, for $k = q+1$, the resulting function is the inconvenience cost function f_{m+1} with optimum t_{m+1}^* (which can be computed a priori and stored with the instance). Then, the condition for a new block (Proposition 3 part (i)) is tested. If true, we know that $I_{q'} = [m+1]$ forms a separate, new block with $q' = q+1$. Otherwise, k is decremented (giving $k = q$), we compute the aggregated inconvenience cost function $f_{(I_q \cup \{m+1\})}$ and its optimum, and check conditions (5), i.e., Proposition 3 part (ii). If true, the last block is $(I_q \cup \{m+1\})$, otherwise k is decremented again and the process is repeated. Concerning the computational complexity, recall that the label of P already stores the functions $f_{(I_k \cup \cdots \cup I_q)}$ so that the new aggregated functions $f_{(I_k \cup \cdots \cup I_q \cup \{m+1\})}$ result from a single aggregation step with two one-dimensional convex functions. Summarizing, we have

$$q' = \begin{cases} q+1, & \text{if } t_{I_q}^* + T_{m,m+1} \leq t_{m+1}^* \\ k \text{ (computed as in Condition 5)}, & \text{otherwise} \end{cases} \tag{6d}$$

and have computed the new aggregated inconvenience cost functions $f_{(I_k \cup \cdots \cup I_q \cup \{m+1\})}$ for all $k \in [p':q']$.

The missing parts are now simple to compute. First, for p' the associated bound is $w_{p'-1}(P') = \alpha_{P'}$ given by (6a). Second, for $k \in [p':q']$, we have

$$w_k(P') = \begin{cases} \min\{w_k(P) + T_{m,m+1}, \ell_{m+1}\}, & \text{if } k = q' \\ w_k(P) + T_{m,m+1}, & \text{otherwise} \end{cases} \tag{6e}$$

$$f^{const}(P') = f^{const}(P) + \Sigma_{p'}^*(P) \tag{6f}$$

$$\Sigma_k^*(P') = \begin{cases} \Sigma_k^*(P) - \Sigma_{p'}^*(P), & \text{if } k \leq q \\ \Sigma_q^*(P) + f_{I_q}(t_{I_q}^*) - \Sigma_{p'}^*(P) & \text{if } k = q+1 \end{cases} \tag{6g}$$

where in the last term the value $f_{I_q}(t_{I_q}^*)$ is computed with the help of the function f_{I_q} stored within the label of P (optimization of a one-dimensional convex function). Third, the standard VRPTW attributes are updated via

$$R^{cost}(P') = R^{cost}(P) + \tilde{c}_{m,m+1} \tag{6h}$$

$$R^{load}(P') = R^{load}(P) + q_{m+1} \tag{6i}$$

$$R^{cust,n}(P') = \begin{cases} R^{cust,n}(P) + 1, & \text{if } n = m+1 \\ R^{cust,n}, & \text{otherwise} \end{cases} \quad \forall n \in N. \tag{6j}$$

The resulting label is feasible if and only if $\alpha_{P'} = w_{p'-1}(P') \leq \ell_{m+1}$, $R^{load}(P') \leq Q$, and $R^{cust,n}(P') \leq 1$ for all $n \in N$.

4.1.3. Dominance.

Let P_1 and P_2 be two different paths ending at the same vertex m with associated labels $(R_1^{cost}, R_1^{load}, (R_1^{cust,n})_{n\in N}, f_{P_1})$ and $(R_2^{cost}, R_2^{load}, (R_2^{cust,n})_{n\in N}, f_{P_2})$, respectively. Here we write, for example, R_1^{cost} instead of $R^{cost}(P_1)$ to simplify the notation of each attribute. In the VRPTW, P_1 dominates P_2 if $R_1^{cost} \leq R_2^{cost}$ and all three following conditions

$$\alpha_{P_1} \leq \alpha_{P_2}, \quad R_1^{load} \leq R_2^{load}, \quad \text{and } R_1^{cust,n} \leq R_2^{cust,n} \quad \forall n \in N \tag{7}$$

hold. By contrast, in the VRPTW-CNC, the tradeoff between the latest possible schedule time x at the last vertex m and the cost must be considered. The consequence is that there exist three possibilities to define valid dominance rules:

1. *Complete dominance between paths P_1 and P_2:* We say that P_1 completely dominates P_2 if (7) holds and
$$R_1^{cost} + f_{P_1}(x) \leq R_2^{cost} + f_{P_2}(x) \qquad \forall x \in dom(P_2) = [\alpha_{P_2}, \ell_m]. \tag{8}$$
In this case, P_2 can be discarded.
2. *Dominance on any real interval:* We say that P_1 properly dominates P_2 on a real interval $[a, b] \subseteq dom(P_2) = [\alpha(P_2), \ell_m]$ if (7) holds and
$$R_1^{cost} + f_{P_1}(x) < R_2^{cost} + f_{P_2}(x) \qquad \forall x \in [a, b]. \tag{9}$$
In this case, one cannot immediately discard the label of P_2. Instead, several paths may be needed to dominate P_2. Such an approach was first suggested in (Ioachim *et al.*, 1998) for a piecewise linear tradeoff, and suggested in (Irnich and Villeneuve, 2006) for k-cycle elimination. Indeed, for a collection of paths $\{P_1^c\}_{c\in C}$, if each P_1^c properly dominates P_2 on a real interval $[a^c, b^c]$ and the union of these intervals is $dom(P_2)$, then P_2 can be discarded.
3. *Dominance on one or several intervals $\mathbf{S}_k$ of P_2:* We say that P_1 dominates P_2 on interval $\mathbf{S}_k$ of P_2 if (7) holds and
$$R_1^{cost} + f_{P_1}(x) < R_2^{cost} + f_{P_2}(x) \qquad \forall x \in \mathbf{S}_k. \tag{10}$$
When (10) holds, the part of the piecewise convex function $f_{P_2}(x)$ defined over interval $\mathbf{S}_k$ can be discarded. For a given path, an interval $\mathbf{S}_k$ corresponds to an optimal block structure and a one-dimensional convex function $f_{P_2}(x)$ defined over $\mathbf{S}_k$, so this dominance idea can be implemented in a new labeling algorithm where each label represents a path with a fixed block structure. The advantage of this implementation is that each label only needs to store a single one-dimensional convex function instead of a piecewise convex function, so that label extension and dominance rules (10) can be implemented in a compact manner and be tested very efficiently. Please refer the implementation details to (He and Song, 2016). However, additional opportunities for label elimination can be offered by dominance on any real interval, which is not limited to intervals $\mathbf{S}_k$ only.

We implement the second dominance rule by storing with each label, i.e., each path $P = (1, \ldots, m)$, the union of intervals on which it is properly dominated. Specifically, the interval of dominance is initialized as $\varnothing$. Then, for two paths P_1 and P_2, in order to determine the inclusion-wise maximal intervals of dominance in (9), we compute the roots of $R_1^{cost} - R_2^{cost} + f_{P_1}(x) - f_{P_2}(x)$. Let the roots be $x_1, x_2, \ldots, x_t$ (a difference of convex functions may have many roots). Then either $(-\infty, x_1)$, (x_2, x_3), (x_4, x_5), ... or (x_1, x_2), (x_3, x_4), ... are the inclusion-wise maximal intervals. Via a sign test, we can determine which case applies. Note that these intervals are open intervals, and we represent them as slightly smaller closed intervals using ε-arithmetic. As soon as this union becomes identical with the domain of f_P, $dom(f_P)$, the label can be discarded. Note that care must be taken to avoid a cyclic dominance between labels, e.g., at time x when equality holds in (8). This is exactly the reason why the "<" sign is used in the definition of dominance on intervals (cf. (9)).

4.1.4. Refinements.

Several powerful refinements have been invented over the years to significantly accelerate the column-generation process. We use the ng-path relaxation and bidirectional labeling. We next describe in details how we adapt the bidirectional labeling procedure into the proposed BCP algorithm for the VRPTW-CNC, and briefly describe the ng-path relaxation in Appendix A.

Bidirectional labeling for SPPRCs was coined by Righini and Salani (2006) in order to mitigate the explosion of labels typically observed when paths grow longer. In bidirectional labeling, both forward paths and backward paths are created, but processed only up to a so-called half-way point, e.g., defined by the midpoint of the planning horizon in VRPTW variants. When labeling terminates, suitable forward and backward labels must be merged to obtain complete feasible 0-$(n+1)$-paths. Several subsequent works have shown that bounded bidirectional labeling algorithms are usually superior to their monodirectional counterparts.

A prerequisite for bidirectional labeling is the capability of reversing the labeling process. For the VRPTW-CNC, the use of the transposed VRPTW-CNC instance allows us to fully reuse the non-trivial forward resource extension functions (cf. Section 4.1.2) also for backward labeling. In the *transposed VRPTW-CNC instance*, arcs are reversed (each original arc (i,j) becomes the arc (j,i)), all time windows and inconvenience cost functions are inverted ($[e_i,\ell_i]$ becomes $[-\ell_i,-e_i]$; $f_i(t)$ becomes $f_i(-t)$), the depot vertices 0 and $n+1$ swap their roles, and all other inputs remain unchanged (fleet size, capacity, demands, routing costs, and travel times). Every feasible $(n+1)$-0-path in the transposed VRPTW-CNC instance is, after reversion, a feasible 0-$(n+1)$-path in the original instance with identical routing and inconvenience costs, and vice versa.

The merging procedure is the more intricate part of bidirectional labeling. For the sake of convenience, from now on we describe backward paths as paths $P'^{bw}=(m,\ldots,n+1)$ that start at a vertex m and end at the destination $n+1$, even though they are generated in reverse direction in the transposed VRPTW-CNC instance. Similar to forward paths, for the backward path P'^{bw} we define $\beta_{P'^{bw}}$ as the latest point in time when the service at m can feasibly start, i.e., $\beta_{P'^{bw}}=-w_{p'}$ where $w_{p'}$ is the lower interval bound of the first interval $\mathbf{S}_{p'}$ when labeling in the transposed VRPTW-CNC instance. Finally, we also describe their inconvenience cost function $f_{P'^{bw}}(x)$ as a non-decreasing function in x over the (original) time window $[\beta_{P'^{bw}},\ell_m]\subseteq[e_m,\ell_m]$. The merge is performed as follows:

First, we choose α_P as the (strictly) monotone attribute that is compared against the half-way point H, i.e., a time point chosen in "the middle" of the time horizon (see also the next paragraph).

Second, the merge condition that a feasible path $P=(P^{fw},P'^{bw})$ results from the concatenation of a forward path P^{fw} and a backward path P'^{bw}, ending/starting at m respectively, is simpler to describe if instead of P'^{bw} we consider its (backward) predecessor path $P^{bw}=pred(P'^{bw})$ starting at $m+1$, i.e., $P'^{bw}=(m,P^{bw})$. The following three conditions must hold for a feasible concatenation $P=(P^{fw},(m,m+1),P^{bw})$:

$$\alpha_{P^{fw}}+T_{m,m+1} \leq \beta_{P^{bw}} \tag{11a}$$

$$R^{load}_{fw}+R^{load}_{bw} \leq Q \tag{11b}$$

$$R^{cust,n}_{fw}+R^{cust,n}_{bw} \leq 1 \qquad \forall n\in N \tag{11c}$$

where $R_{fw}, f_{P^{fw}}$ and $R_{bw}, f_{P^{fw}}$ refer to the attributes of paths P^{fw} and P^{bw}, respectively.

Third, the computation of the reduced cost of $P = (P^{fw}, (m, m+1), P^{bw})$ is the most difficult part. We generalize the ideas of Tilk *et al.* (2016) for this step (they presented bidirectional labeling for piecewise linear costs). Note first that the reduced cost can be written as

$$\tilde{c}_P = R^{cost}_{fw} + \tilde{c}_{m,m+1} + R^{cost}_{bw} + \min_{\substack{x_{fw} \in dom(P^{fw}),\\ x_{bw} \in dom(P^{bw}):\\ x_{fw} + T_{m,m+1} \le x_{bw}}} \Big(f_{P^{fw}}(x_{fw}) + f_{P^{bw}}(x_{bw}) \Big). \tag{12}$$

The determination of the minimum in (12) is simple and given by the optima of $f_{P^{fw}}$ and $f_{P^{bw}}$ if they are not conflicting, i.e., if $t^*_{fw} + T_{m,m+1} \le t^*_{bw}$ (recall that for breaking ties the optimum is chosen as the smallest/largest possible value, if $f_{P^{fw}}/f_{P^{bw}}$ contains a constant piece, respectively).

Hence, we can assume that there is a conflict between the minima, i.e., $t^*_{fw} + T_{m,m+1} > t^*_{bw}$. Then, we know that $x_{fw} + T_{m,m+1} = x_{bw}$ must hold in (12). One possibility to compute an optimal x_{fw} (or x_{bw}) is to iteratively aggregate the last few blocks of P^{fw} and P^{bw}. There is however a more elegant way to accomplish this task when the derivative of $f_{P^{fw}}$ and $f_{P^{bw}}$ can be computed (which is simple in the case of quadratic inconvenience cost functions): At the minimum, x_{fw} fulfills that $\frac{df_{P^{fw}}}{dx}(x_{fw}) + \frac{df_{P^{bw}}}{dx}(x_{fw} + T_{m,m+1}) = 0$. Note that the above function $d(x) := \frac{df_{P^{fw}}}{dx}(x) + \frac{df_{P^{bw}}}{dx}(x + T_{m,m+1})$ is continuous, strictly monotone, and piecewise defined. The pieces result from the intersection of intervals $\mathbf{S}^{fw}_k$ of the forward label of path P^{fw} with shifted intervals $\mathbf{S}^{bw}_{k'} - T_{m,m+1}$ of the backward label of path P^{bw}. (Intervals $\mathbf{S}^{fw}_k$'s and $\mathbf{S}^{bw}_{k'}$'s are defined for the forward label and backward label, respectively, cf. Proposition 1.) We start with $k = r(\alpha_{P^{fw}})$ and k' such that $\alpha_{P^{fw}} + T_{m,m+1} \in \mathbf{S}^{bw}_{k'}$. Hence, the first piece is $[a, b] = [w^{fw}_{k-1}, w^{fw}_k] \cap [w^{bw}_{k'-1} - T_{m,m+1}, w^{bw}_{k'} - T_{m,m+1}]$. If $d(a) \ge 0$, then $x_{fw} = a$ and $x_{bw} = a + T_{m,m+1}$ and the minimum in (12) is known. If $d(a) < 0$ and $d(b) \ge 0$, then the unique root x^* of $d(x)$, i.e., an $x^* \in [a, b]$ with $d(x^*) = 0$, gives the minima $x_{fw} = x^*$ and $x_{bw} = x^* + T_{m,m+1}$ for (12). Otherwise $d(a) < 0$ and $d(b) < 0$. Then, we either increase k or decrease k' (or both) to determine the next piece $[a, b] = \mathbf{S}^{fw}_k \cap (\mathbf{S}^{bw}_{k'} - T_{m,m+1})$ and repeat the above two tests. If passing the last piece, the optimum is attained at the latest possible time $x_{fw} = \min\{\ell_m, \beta_{P^{bw}} - T_{m,m+1}\}$ and $x_{bw} = \min\{\ell_m + T_{m,m+1}, \beta_{P^{bw}}\}$.

Finally, the generation of duplicate paths in the merging step is avoided by using the following check: forward paths P^{fw} must either end at the destination vertex $n+1$ (then the merge is with the trivial backward path $P'^{bw} = (n+1)$) or $\alpha_{P^{fw}} \ge H$ must hold. For details refer to (Righini and Salani, 2006).

Example 4. *(continued from Example 1 to 3) We consider again the forward path $P^{fw} = (0, 1, 2, 3, 4, 5)$ analyzed in Example 2 with $f_{P^{fw}}$ depicted in Figure 4). To demonstrate the merge procedure, we consider the backward path $P'^{bw} = (5, 6, n+1)$. Data for vertex 6 and arc (5, 6) was already provided in Example 1. In addition, we define the time window $[e_{n+1}, \ell_{n+1}] = [20, 26]$ and the inconvenience cost function $f_{n+1}(t) = (t - 24)^2$ of the destination $n+1$. The travel time between 6 and $n+1$ is $T_{6,n+1} = 8$.*

With the planning horizon $[0, 26]$, we define the half-way point as $H = 13$ so that $m = 5$ is really the merge vertex. The predecessor path of P'^{bw} is $P^{bw} = (6, n+1)$. The backward paths and the inconvenience cost function $f_{P^{bw}}(x)$ are shown in Figure 6. As $f^*_{fw} = 15$, $f^*_{bw} = 14$, and $T_{5,6} = 2$, the optima of the forward and the backward path are in conflict. Figure 7 shows how merging is performed on this example in detail: The merge procedure stops already in the second iteration with

$k = 4$ and $k' = 2$. The intersection is $[a, b] = \mathbf{S}_4^{fw} \cap \mathbf{S}_2^{bw} = [12\frac{1}{2}, 13] \cap [14-2, 16-2] = [12\frac{1}{2}, 13]$. *By computing the root of $d(x)$ we get $x_{fw} = 12\frac{5}{6}$ and $x_{bw} = 14\frac{5}{6}$.* □

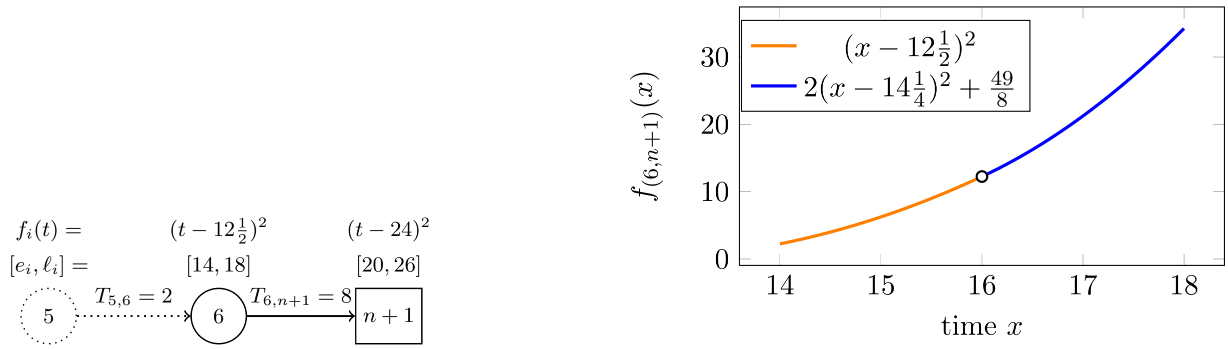


Figure 6: Backward path $P'^{bw} = (5, 6, n+1)$ and its predecessor $P^{bw} = (6, n+1)$ with time windows, travel times, and inconvenience cost functions (left side); Function $f_{P^{bw}}(x)$ for the backward path $P^{bw} = (6, n+1)$ of Example 4 defined piecewise on the two intervals $\mathbf{S}_2 = [14, 16]$ and $\mathbf{S}_1 = [16, 18]$ (right side).

Operations on Inconvenience Cost Functions. We briefly summarize the operations on inconvenience cost functions that are needed in the course of the dynamic-programming based labeling algorithm as described above. The operations are:

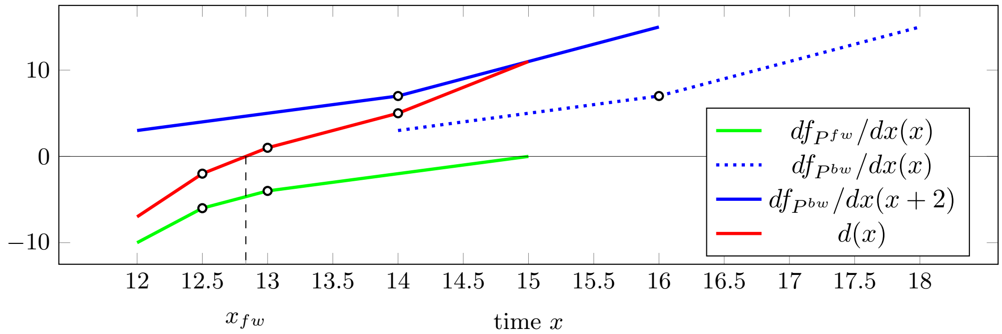
(i) the shift of a function f by $T \in \mathbb{R}$, i.e., the function $x \mapsto f(x - T)$;
(ii) the addition/subtraction of two functions f and g;
(iii) the addition of a constant to a function f;
(iv) the determination of the sign (positive/negative/0) of f at a point $x \in \mathbb{R}$;
(v) the determination of the roots of a function f;
(vi) the reflection of a function f, i.e., the function $x \mapsto f(-x)$; and
(vii) the determination of the derivative $\frac{df}{dx}(x)$, as well as operations on $\frac{df}{dx}(x)$ such as shift, addition, roots, and sign test defined above for $f(x)$.

In the simplest case of a pure forward labeling approach, operations (i) and (ii) are needed when extending labels (aggregation is based on shift and addition). Operations (iii)–(v) are needed in the dominance rules because intervals of dominance can be determined by computing the roots of $f_{P_1}(x) - f_{P_2}(x) + R_1^{cost} - R_2^{cost}$, see (8) and (9). Pure backward labeling requires in addition the reflection operation (vi), while our merge procedure needs operations (vii) as it is based on comparing the derivatives of the forward and backward labels.

It is clear that in the case of convex quadratic inconvenience cost functions, all these operations can be performed in $\mathcal{O}(1)$. In the cases of convex polynomials of degree no more than 4, the determination of roots can be tricky but explicit formulas are available. In the cases of more general convex inconvenience cost functions, numerical approaches such as the Newton's method could be applied to compute the roots.

4.2. Cutting Planes, Branching, and Node Selection

We use two classes of valid inequalities to strengthen the linear programming relaxation of (3), 2-path cuts and SR inequalities. We briefly sketch the inequalities, the separation algorithms we use,



Iteration		1	2	3	4	
Forward	$p =$	3	4	5	5	
	$\mathbf{S}_p^{fw} =$	$[12, 12\frac{1}{2}]$	$[12\frac{1}{2}, 13]$	$[13, 15]$	$[13, 15]$	
	$f_{P^{fw}}(x) =$	$4(x-13\frac{1}{4})^2+9\frac{1}{4}$	$2(x-14)^2+7\frac{1}{2}$	$(x-15)^2+5\frac{1}{2}$	$(x-15)^2+5\frac{1}{2}$	
	$df_{P^{fw}}/dx(x) =$	$8x-106$	$4x-56$	$2x-30$	$2x-30$	
Backward	$q =$	2	2	2	1	
	$\mathbf{S}_q^{bw} =$	$[14, 16]$	$[14, 16]$	$[14, 16]$	$[16, 18]$	
	$f_{P^{bw}}(x) =$	$(x-12\frac{1}{2})^2$	$(x-12\frac{1}{2})^2$	$(x-12\frac{1}{2})^2$	$(x-14\frac{1}{4})^2+\frac{49}{8}$	
	$df_{P^{bw}}/dx(x) =$	$2x-25$	$2x-25$	$2x-25$	$4x-57$	
Intersection	$[a, b] =$	$[12, 12\frac{1}{2}]$	$[12\frac{1}{2}, 13]$	$[13, 14]$	$[14, 15]$	
	$df_{P^{fw}}/dx(a) =$	-10	-6	-4	-2	± 0 (for $b = 15$)
	$df_{P^{bw}}/dx(a+2) =$	$+3$	$+4$	$+5$	$+7$	$+11$ (for $b = 15$)
	$d(a), d(b) =$	$-7, -2$	$-2, +1$	$+1, +5$	$+5, +11$	

Figure 7: Merging of forward path $P^{fw} = (0, 1, 2, 3, 4, 5)$ and backward path $P'^{bw} = (5, 6, n+1)$ of Example 4.

and their impact on the SPPRC pricing subproblem in Appendix A. We use a two-level branching scheme: first we branch on the number of vehicles whenever $\sum_{P\in\Omega}\lambda_P$ is fractional. The second level is the standard branching on arcs $(i,j)\in A$: let the coefficient b_{ijP} denote the number of times that arc (i,j) occurs in path P, we choose the arc with value $\sum_{P\in\Omega} b_{ijP}\lambda_P$ closest to 0.5 among all arcs $(i,j)\in A$. We use the best first search strategy to select the next node to explore in the branch-and-bound tree.

5. Computational Results

The BCP algorithm was implemented in C++ and compiled into 64-bit single-thread code with MS Visual Studio 2015. The experiments were conducted on a standard PC with an Intel(R) Core(TM) i7-5930k clocked at 3.5 GHz and 64 GB of RAM, by allowing a single thread for each run. CPLEX 12.7.0 was used to solve the RMPs in the BCP algorithm.

We use the well-known VRPTW benchmark instances from Solomon (1987) with the standard convention of rounding to one decimal place (following Kohl *et al.* (1999) and later articles) when computing the coefficients c_{ij} and τ_{ij} for all arcs $(i,j)\in A$. We define the inconvenience cost function at each customer $i\in N$ as a convex quadratic function $f_i(t)=\rho\cdot\left(t-\frac{e_i+\ell_i}{2}\right)^2$, where $\rho\geq 0$ is an *inconvenience cost factor* that we vary in the following experiments. The value of every inconvenience cost function is zero in the middle of the corresponding time window. For the depot node 0 and $n+1$, we assume that there is no inconvenience cost.

The BCP algorithm uses an ng-neighborhood of size 12 and applies heuristic pricing using smaller networks with 2, 5, and 10 arcs per vertex. Moreover, we stop adding SR inequalities when a total limit of 100 added cuts is reached, and no more than 10 SR inequalities per customer are allowed. There is no limitation for adding 2-path cuts.

5.1. Variation of the Inconvenience Cost Factor ρ

In the first experiment, we analyze how the costs and fleet sizes vary as we vary the inconvenience cost factor ρ. The experiment is performed in order to define reasonable values for the inconvenience factor ρ, which is used to scale the quadratic inconvenience cost functions $f_i(t)$ defined above.

Figure 8 shows how the inconvenience cost, routing cost, and the overall cost for a 50-customer instance `R102.50` depend on the factor ρ. Note that the solution of this instance as a pure VRPTW, i.e., for $\rho=0$, has a routing cost of 909.0 and 11 vehicles/routes are used. Starting from $\rho=10^{-3}$, we increase ρ by a factor of 1.05 in each iteration so that we reach $\rho=100$ in 239 iterations. Each instance, for a given ρ value, is solved twice as explained in the following.

In the left part of this figure, the fleet size is not limited ($K=\infty$ as in the original Solomon instances). With an increasing ρ, the inconvenience cost varies non-monotonically between 0 and approximately 102. In contrast, the routing and overall costs increase monotonically. The routing cost function is a staircase function while the overall cost changes continuously.

We also display on the left-hand-side of Figure 8 the change of the fleet size: Starting with 11 vehicles for $\rho=10^{-3}$, the final solution for $\rho=100$ requires 23 vehicles and the average route length is nearly two customers per route. This means that a high ρ value drives the optimal solution to only visit customers at their preferred service time, i.e., the middle of the corresponding service time window. A high ρ value does not increase the inconvenience cost much because high inconvenience costs can be circumvented by using more vehicles. Indeed, starting from $\rho\approx 36$, the solution does not change any more because the 23 routes can ensure that every customer $i \in N$ is served at the preferred time $(e_i + \ell_i)/2$ so that routing and overall costs become identical (=1389.9).

In the right part of Figure 8, we fix the number K of vehicles to 11. Now the inconvenience and overall costs increase very fast. After $\rho \approx 0.39$, the inconvenience cost exceeds the routing cost, which indicates that a larger ρ hardly makes sense.

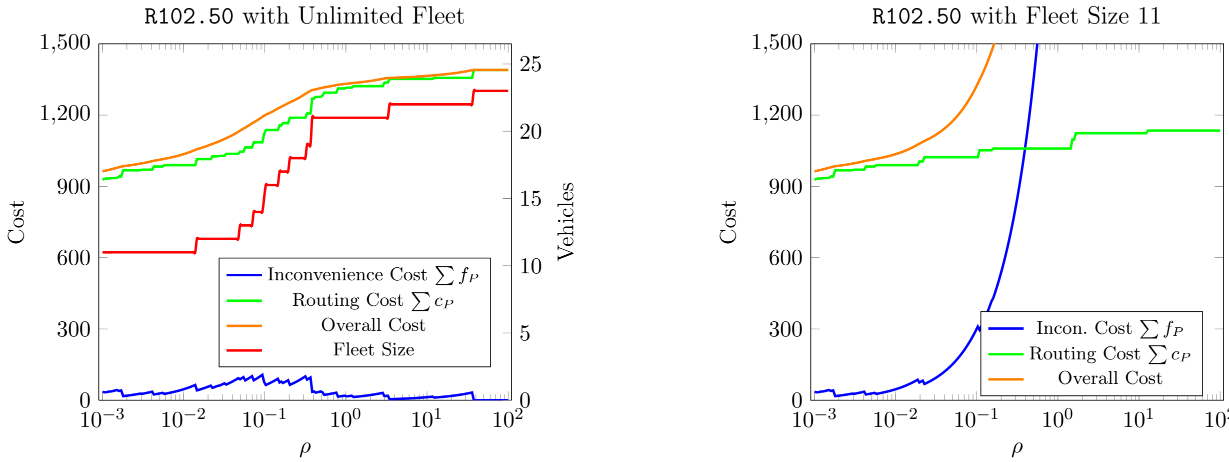


Figure 8: Costs and fleet sizes depending on ρ for instance R102.50.

5.2. Computational Performance

Based on the insights of the preceding section, we use $\rho = 0.001, 0.01, 0.05$, and 0.1 in the following experiments, in which we also distinguish between unlimited and limited fleet. In the latter case, K is always chosen as the fleet size K^* of the respective optimal VRPTW solution. As optimal routing costs c^* and fleet sizes K^* of the Solomon VRPTW instance are scattered over the literature, we provide these values in Section B of the e-Companion. Overall, the testbed for the VRPTW-CNC comprises $56 \cdot 3 \cdot 2 \cdot 4 = 1344$ instances (56 Solomon instances; $n = 25$, 50, and 100; unlimited and limited fleet sizes; four ρ values).

Table 1 summarizes the outcome of the performance evaluation. The table entries have the following meaning:

- *#opt*: the number of instances solved to proven optimality;
- *Time*: the average computational time in seconds, where the computational time for any unsolved instance is counted as the overall time limit of 3600 seconds;
- *B&B*: the average number of branch-and-bound nodes solved by the BCP algorithm, where entries smaller than one can result if root nodes were not solved;
- The columns in section $\pm\%$ report the increase of costs and fleet sizes relative to the respective optimal VRPTW solutions, i.e., the respective c^* and K^*:
  - $\sum c_P$: the average percentage of increase in terms of the routing cost ($=100 \times (\sum c_P - c^*)/c^*$);
  - $\sum f_P$: the average percentage of the inconvenience cost relative to c^* ($=100 \times \sum f_P / c^*$);
  - *Fleet*: the average percentage of increase in terms of the fleet size ($=100 \times (K - K^*)/K^*$).

Note that averages in this section are taken only over those VRPTW-CNC instances that were solved to optimality. Further aggregated results, grouped by the Solomon classes R1, C1, RC1, R2, C2, and RC2 can be found in Section C of the e-Companion.

Instances		Unlimited Fleet						Limited Fleet				
					±%						±%	
ρ	n	#opt	Time	B&B	$\sum c_P$	$\sum f_P$	Fleet	#opt	Time	B&B	$\sum c_P$	$\sum f_P$
0.001	25	56/56	63.5	1.4	11.9	9.0	30.7	53/56	357.1	1.0	6.6	19.4
	50	40/56	1495.1	2.4	7.8	7.3	13.2	33/56	1659.7	1.7	2.8	7.5
	100	10/56	3060.7	0.8	0.2	3.7	0.7	10/56	3025.3	0.6	0.2	3.7
	Total	106/168	1539.8	1.5	9.3	7.9	21.3	96/168	1680.7	1.1	4.6	13.7
0.01	25	56/56	1.8	1.1	36.8	14.9	89.0	52/56	398.9	0.9	16.7	155.2
	50	51/56	480.8	4.9	44.3	15.7	125.4	26/56	2119.4	5.3	10.0	26.6
	100	19/56	2541.0	34.8	44.5	18.3	128.8	9/56	3075.4	0.5	2.5	29.9
	Total	126/168	1007.9	13.6	41.0	15.8	109.7	87/168	1864.6	2.2	13.2	103.8
0.05	25	56/56	0.5	1.0	59.9	10.6	141.5	50/56	450.1	1.0	19.7	555.7
	50	55/56	125.8	1.4	68.1	16.0	181.7	24/56	2291.5	3.1	13.0	133.1
	100	37/56	1604.9	2.1	98.5	16.6	262.8	9/56	3172.1	0.5	5.1	144.2
	Total	148/168	577.1	1.5	72.6	14.1	186.8	83/168	1971.2	1.5	16.2	388.9
0.1	25	56/56	0.5	1.0	65.0	13.2	157.4	51/56	471.1	1.1	23.3	1097.0
	50	55/56	162.2	3.4	79.6	17.2	210.8	24/56	2359.3	9.1	14.8	256.5
	100	43/56	1186.0	4.7	104.9	18.3	297.6	8/56	3233.9	0.7	5.5	233.2
	Total	154/168	449.5	3.0	81.4	16.1	215.6	83/168	2021.4	3.6	19.1	770.7
Total		534/672	893.6	4.9	55.1	13.8	144.1	349/672	1884.5	2.1	13.0	305.4

Table 1: Aggregated results over 1344 VRPTW-CNC instances.

Comparing the number of solved instances and the average computational time, the VRPTW-CNC with limited fleet size is much harder to solve than the one with unlimited fleet size (349 vs. 534 instances solved out of a total of 672 instances, 1884.5 seconds vs. 893.6 seconds).

The impact of the inconvenience cost factor ρ on the performance of the BCP algorithm strongly depends on the presence of a fleet size limit. On one hand, a larger ρ makes instances with unlimited fleet easier to solve. A larger ρ value somewhat guides the optimal solution to where the service start time at each customer $i \in N$ is close to $(e_i + \ell_i)/2$, i.e., where the inconvenience costs diminish. Hence, a large ρ mimics the pure VRPTW with narrow time windows. This is in line with the observation that the fleet size in use drastically increases with a larger ρ: meeting the preferred service start time $(e_i + \ell_i)/2$ of each customer makes the schedule optimization less flexible (all these preferred times concentrate around the midpoint of the planning horizon), and a cost-efficient solution only results when more vehicles can compensate the loss in the schedule flexibility. Note that for unlimited fleet, the routing costs can increase up to +652% while the inconvenience costs never exceed 114.2% of the routing cost in the optimal VRPTW solution, see detailed results in Section D of the e-Companion.

On the other hand, a larger ρ makes the limited fleet instances harder to solve. We interpret this outcome in the following way: With a limited fleet size $K = K^*$, even finding feasible solutions becomes a (practically) more difficult optimization problem. From BCP algorithms for the VRPTW it is known that, when branching on the number of vehicles, the $\leq$-branch often consumes a rather long computational time. Moreover, in the VRPTW-CNC with limited fleet, a larger ρ makes the inconvenience costs the main part of the overall cost. Indeed, the detailed results in Section D of the e-Companion show that inconvenience costs can reach 27294% (i.e. a factor of 273 in the overall

	Geometric mean $z^*/z(2SH)$											
	$n = 25$						$n = 50$					
ρ	R1	C1	RC1	R2	C2	RC2	R1	C1	RC1	R2	C2	RC2
0.001	0.974	0.645	0.979	0.634	0.347	0.594	0.993	0.746	0.992	0.735	0.468	0.636
0.01	0.827	0.400	0.795	0.163	0.146	0.153	0.738	0.426	0.924	0.122	0.109	0.153
0.05	0.504	0.150	0.458	0.050	0.048	0.040	0.425	0.166	0.454	0.036	0.029	0.030
0.1	0.380	0.088	0.305	0.027	0.027	0.022	0.310	0.094	0.325	0.020	0.015	0.021

Table 2: Comparison between VRPTW-CNC solutions and solutions obtained with the two-stage heuristic.

cost). Here, inconvenience costs completely superimpose the routing costs, which never increase by more than 104.3%. This obviously complicates finding good routing decisions for the SPPRC subproblem.

5.3. Value of Integrated Optimization

Finally, we compare the costs of optimal VRPTW-CNC solutions with the costs of solutions obtained by the following two-stage heuristic (*2SH*): First solve the respective VRPTW to obtain the routes and routing costs $\sum c_P$, and then solve the service scheduling problem (2) over each route to obtain $\sum f_P$. While the VRPTW-CNC optimizes $\sum(c_P + f_P)$ simultaneously, 2SH optimizes $\sum c_P$ first, and then optimizes f_P for each obtained route P.

Table 2 shows the value of integrated optimization. More precisely, let z^* denote the cost of an optimal solution to the VRPTW-CNC with unlimited fleet and let $z(2SH)$ denote the cost of the solution obtained by the 2SH. To make it a fair comparison, when solving the VRPTW in the 2SH, we set the number of vehicles to be the same as the one used in the optimal solution of the VRPTW-CNC. The geometric mean of the ratios $z^*/z(2SH)$ (note that ratios are always less than or equal to 1) is shown for instances with different sizes ($n = 25$ and $n = 50$) and for different Solomon classes. We omit instances with $n = 100$ due to the relatively small number of instances solved to optimality for the VRPTW-CNC in this case.

We see from Table 2 that the value of integrated routing and schedule optimization is significant in terms of the total cost. Sequentially solving the two problems separately leads to a much higher cost, mostly as a result of a high inconvenience cost. First, comparing the results across different classes of instances, we see that the benefit of integrated routing and scheduling decisions is higher in the C1 and C2 instances, where customers are located roughly in several clusters, whereas it is lower in the R1 and R2 instances, where customers are randomly spread out (results on RC1 and RC2 instances lie between these two extremes). Optimal VRPTW solutions to C1 and C2 instances fully exploit the cluster structure to minimize the routing cost. Their routes are locally concentrated on the clusters, leading to a high inconvenience cost in 2SH. In addition, we see that the value of integrated decision is much higher for instances with wider time windows (Series 2 with classes R2, C2, and RC2) than for instances with narrower time windows (Series 1). The explanation is simple: All optimal VRPTW solutions exploit the entire time windows (for the purpose of minimizing travel cost). For instances with wider time windows, some service start times then lie farther away from the preferred service start times of the VRPTW-CNC, leading to higher inconvenience costs. Finally, comparing the results for different ρ values, we see that the value of integration increases monotonically as the value of ρ increases. Indeed, the differences between optimal routes obtained by the VRPTW-CNC and optimal routes obtained by the VRPTW become more significant when more weights are put on the inconvenience cost in the objective function, as ρ increases.

6. Conclusions

Customer satisfaction and transportation cost are among the most frequently used metrics by logistics companies to measure operational performance. In this paper, we proposed the VRPTW-CNC as a model that captures both objectives, because it minimizes a linear combination of routing costs and customers' inconvenience costs, by determining vehicle routes for a given fleet and scheduled service times for customers simultaneously. With general convex inconvenience cost functions, the VRPTW-CNC falls into the category of mixed-integer convex programs, which are very challenging to solve in general. We proposed a BCP algorithm to solve the VRPTW-CNC to optimality. Its key component is an effective labeling algorithm for the pricing problem, which heavily exploits the structure of optimal schedule times over a fixed route. Extensive computational results demonstrated that: (1) Taking customers' inconvenience costs into consideration has a significant impact on the decisions of fleet size and vehicle routes; (2) With a fixed fleet size, placing more emphasis on customers' inconvenience costs increases the difficulty of the problem; (3) VRPTW-CNC instances with up to 100 customers can be solved exactly with our BCP algorithm. It can serve as a useful planning tool to evaluate the trade-off between operating cost and service quality for fleet managers.

References

Baldacci, R., Mingozzi, A., and Roberti, R. (2011). New route relxation and pricing strategies for the vehicle routing problem. *Operations Research*, **59**, 1269–1283.

Baldacci, R., Mingozzi, A., and Roberti, R. (2012a). New state-space relaxations for solving the traveling salesman problem with time windows. *INFORMS Journal on Computing*, **24**(3), 356–371.

Baldacci, R., Mingozzi, A., and Roberti, R. (2012b). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, **218**(1), 1–6.

Bélanger, N., Desaulniers, G., Soumis, F., and Desrosiers, J. (2006). Periodic airline fleet assignment with time windows, spacing constraints, and time dependent revenues. *European Journal of Operational Research*, **175**(3), 1754–1766.

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

Desaulniers, G., Desrosiers, J., and Solomon, M., editors (2005). *Column Generation*. Springer, New York, NY.

Desaulniers, G., Madsen, O. B., and Ropke, S. (2014). The vehicle routing problem with time windows. In D. Vigo and P. Toth, editors, *Vehicle Routing*, chapter 5, pages 119–159. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Drexl, M. (2012). Synchronization in vehicle routing-a survey of VRPs with multiple synchronization constraints. *Transportation Science*, **46**(3), 297–316.

Dror, M. (1994). Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research*, **42**(5), 977–978.

Dumas, Y., Soumis, F., and Desrosiers, J. (1990). Technical note: optimizing the schedule for a fixed vehicle path with convex inconvenience costs. *Transportation Science*, **24**(2), 145–152.

Fagerholt, K. (2001). Ship scheduling with soft time windows: An optimisation based approach. *European Journal of Operational Research*, **131**(3), 559–571.

Hashimoto, H., Ibaraki, T., Imahori, S., and Yagiura, M. (2006). The vehicle routing problem with flexible time windows and traveling times. *Discrete Applied Mathematics*, **154**(16), 2271–2290.

He, Q. and Song, Y. (2016). Vehicle routing problems with time windows and convex node costs. Technical report, Optimization Online. `http://www.optimization-online.org/DB_FILE/2016/08/5608.pdf`.

Ioachim, I., Gelinas, S., Soumis, F., and Desrosiers, J. (1998). A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. *Networks*, **31**(3), 193–204.

Ioachim, I., Desrosiers, J., Soumis, F., and Bélanger, N. (1999). Fleet assignment and routing with schedule synchronization constraints. *European Journal of Operational Research*, **119**(1), 75–90.

Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 33–65. Springer.

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

Jaw, J.-J., Odoni, A. R., Psaraftis, H. N., and Wilson, N. H. (1986). A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research Part B: Methodological*, **20**(3), 243–257.

Jepsen, M., Petersen, B., Spoorendonk, S., and Pisinger, D. (2008). Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, **56**(2), 497–511.

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

Liberatore, F., Righini, G., and Salani, M. (2011). A column generation algorithm for the vehicle routing problem with soft time windows. *4OR*, **9**(1), 49–82.

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

Pecin, D., Contardo, C., Desaulniers, G., and Uchoa, E. (2017). New enhancements for the exact solution of the vehicle routing problem with time windows. *INFORMS Journal on Computing*, **29**(3), 489–502.

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

Sexton, T. R. and Bodin, L. D. (1985a). Optimizing single vehicle many-to-many operations with desired delivery times: I. Scheduling. *Transportation Science*, **19**(4), 378–410.

Sexton, T. R. and Bodin, L. D. (1985b). Optimizing single vehicle many-to-many operations with desired delivery times: II. Routing. *Transportation Science*, **19**(4), 411–435.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, **35**(2), 254–265.

Stojković, M. and Soumis, F. (2001). An optimization model for the simultaneous operational flight and pilot scheduling problem. *Management Science*, **47**(9), 1290–1305.

Tilk, C., Bianchessi, N., Drexl, M., Irnich, S., and Meisel, F. (2016). Branch-and-price-and-cut for the active-passive vehicle-routing problem. *Transportation Science*.

Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2015). Timing problems and algorithms: Time decisions for sequences of activities. *Networks*, **65**(2), 102–128.

Zamorano, E. and Stolletz, R. (2017). Branch-and-price approaches for the multiperiod technician routing and scheduling problem. *European Journal of Operational Research*, **257**(1), 55–68.

Appendix

A. Additional Refinement of the BCP Algorithm using *ng*-Routes and Valid Inequalities

ng-Path Relaxation. As mentioned in Section 1.1, allowing non-elementary *ng*-paths in the RMP helps to control the tradeoff between the strength of the corresponding linear programming relaxation bound and the practical difficulty of solving SPPRC subproblems.

The *ng*-path relaxation is a family of relaxations defined by the so-called *neighborhoods* $(N_i)_{i \in N}$, where each N_i fulfills $i \in N_i \subset N$. For a customer $i \in N$, the neighborhood N_i comprises a set of customers $j \in N_i$ with a high potential to occur on a cycle $C = (i, \dots, j, \dots, i)$. The *ng*-path relaxation allows exactly those non-elementary paths $P = (r_0, r_1, \dots, r_m)$, where on each cycle $i = r_j = r_k$, $j < k$, there exists an index h with $j < h < k$ and $i \notin N_h$, see Baldacci *et al.* (2011) for details.

2-*Path Cuts.* The first class of valid inequalities (cuts) is the *k-path cuts* as introduced by Kohl *et al.* (1999) for the VRPTW with $k = 2$. For any subset $W \subset N$ that cannot be visited by a single vehicle due to time window constraints, the flow into/out of W must be greater than or equal to 2. Let $\delta^+(W)$ be the positive cut set consisting of all arcs $(i, j) \in A$ with $i \in W$ and $j \notin W$, then the corresponding 2-path inequality is given by $\sum_{P \in \Omega} \sum_{(i,j) \in \delta^+(W)} b_{ijP} \lambda_P \geq 2$, where the coefficient b_{ijP} describes the number of times that route P traverses arc (i, j). Testing that W qualifies for a 2-path cut can be done by showing that there is no feasible solution to the *traveling salesman problem with time windows* (TSPTW) defined on $W \cup \{0, n+1\}$ (with origin 0, destination $n+1$, and travel times and time windows as in the VRPTW-CNC). 2-path cuts are robust cuts, i.e., they leave the structure of the pricing problem unaltered. Only the dual value of each 2-path cut present in the RMP needs to be subtracted from the reduced cost of the arcs $(i, j) \in \delta^-(W)$. We use the heuristic proposed by Kohl *et al.* (1999) to generate candidate sets W. For each candidate set, we solve the corresponding TSPTW with a dynamic programming algorithm (as the one of Baldacci *et al.*, 2012a), and if infeasible the corresponding inequality is added.

Subset-Row Inequalities. The second class of valid inequalities that we use is the *subset-row inequalities* (SR inequalities, Jepsen *et al.*, 2008) defined on subsets $U \subset N$ of cardinality three. Defining h_{UP} as the number of times that route P visits a customer in U, the associated SR inequality is $\sum_{P \in \Omega} \lfloor \frac{h_{UP}}{2} \rfloor \lambda_P \leq 1$. Violated SR inequalities can be separated by straightforward enumeration. The addition of SR inequalities to the RMP, however, requires some adjustments in the labeling algorithm. The value of the dual price $\sigma_U \leq 0$ of a subset-row inequality defined on subset U has to be subtracted from the reduced cost of a label for every second visit to vertices in U (one can interpret $-\sigma_U$ as a *penalty* for visiting vertices in U). Therefore, an additional binary resource $R^{SR,U}$ is needed in each label for each (active) inequality counting the parity of the number of times that a vertex in U is visited (for details see Jepsen *et al.*, 2008).

For comparing two paths P_1 and P_2 w.r.t. dominance, let

$$\sigma(P_1, P_2) = \sum_{U: R_1^{SR,U}=1, R_2^{SR,U}=0} \sigma_U$$

be the sum of the dual prices of SR inequalities on which P_1 is closer to paying the penalty $-\sigma_U$ than P_2. Then, the non-negative term $-\sigma(P_1, P_2)$ has to be added to the LHS of (8) or (9). Note that the resources $T^{SR,U}$ are not compared directly, i.e., the only dominance conditions are given by (7) and the modified inequalities (8) and (9), respectively.

B. Optimal Routing Costs and Fleet Sizes of Solomon's VRPTW Instances

Table 3 gives the minimum routing cost c^* and the number K^* of vehicles used in this solution for all Solomon instances. These results were either computed with our own BCP algorithm for the VRPTW or taken from the literature[1].

Series 1 — Short Routes							Series 2 — Long Routes						
	$n = 25$		$n = 50$		$n = 100$			$n = 25$		$n = 50$		$n = 100$	
Instance	c^*	K^*	c^*	K^*	c^*	K^*	Instance	c^*	K^*	c^*	K^*	c^*	K^*
R101	617.1	8	1044.0	12	1637.7	20	R201	463.3	4	791.9	6	1143.2	8
R102	547.1	7	909.0	11	1466.6	18	R202	410.5	4	698.5	5	1029.6	8
R103	454.6	5	772.9	9	1208.7	14	R203	391.4	3	605.3	5	870.8	6
R104	416.9	4	625.4	6	971.5	11	R204	355.0	2	506.4	2	731.3	5
R105	530.5	6	899.3	9	1355.3	15	R205	393.0	3	690.1	4	949.8	5
R106	465.4	5	793.0	8	1234.6	13	R206	374.4	3	632.4	4	875.9	5
R107	424.3	4	711.1	7	1064.6	11	R207	361.6	3	575.5	3	794.0	4
R108	397.3	4	617.7	6	932.1	10	R208	328.2	1	487.7	2	701.0	4
R109	441.3	5	786.8	8	1146.9	13	R209	370.7	2	600.6	4	854.8	5
R110	444.1	5	697.0	7	1068.0	12	R210	404.6	3	645.6	4	900.5	6
R111	428.8	4	707.2	7	1048.7	12	R211	350.9	2	535.5	3	746.7	4
R112	393.0	4	630.2	6	948.6	10							
C101	191.3	3	362.4	5	827.3	10	C201	214.7	2	360.2	3	589.1	3
C102	190.3	3	361.4	5	827.3	10	C202	214.7	2	360.2	3	589.1	3
C103	190.3	3	361.4	5	826.3	10	C203	214.7	2	359.8	3	588.7	3
C104	186.9	3	358.0	5	822.9	10	C204	213.1	1	588.7	2	588.1	3
C105	191.3	3	362.4	5	827.3	10	C205	214.7	2	359.8	3	586.4	3
C106	191.3	3	362.4	5	827.3	10	C206	214.7	2	359.8	3	586.4	3
C107	191.3	3	362.4	5	827.3	10	C207	214.5	2	359.6	3	585.8	3
C108	191.3	3	362.4	5	827.3	10	C208	214.5	2	350.5	2	585.8	3
C109	191.3	3	362.4	5	827.3	10							
RC101	461.1	4	944.0	8	1619.8	15	RC201	360.2	3	684.8	5	1261.8	9
RC102	351.8	3	822.5	7	1457.4	14	RC202	338.0	3	613.6	5	1092.3	8
RC103	332.8	3	710.9	6	1258.0	11	RC203	326.9	3	555.3	4	923.7	5
RC104	306.6	3	545.8	5	1132.3	10	RC204	299.7	3	444.2	3	783.5	4
RC105	411.3	4	855.3	8	1513.7	15	RC205	338.0	3	630.2	5	1154.0	7
RC106	345.5	3	723.2	6	1372.7	12	RC206	324.0	3	610.0	5	1051.1	7
RC107	298.3	3	642.7	6	1207.8	12	RC207	298.3	3	558.6	4	962.9	6
RC108	294.5	3	598.1	6	1114.2	11	RC208	269.1	2	476.7	3	776.1	4

Table 3: Optimal routing costs c^* and fleet size K^* for Solomon's VRPTW instances

[1] We would like to thank Prof. Guy Desaulniers for providing some values that we could not compute by ourselves or retrieve from public sources.

C. Aggregated Computational Results

Tables 4 and 5 provide more details on the performance of the BCP algorithm for the VRPTW-CNC. Results are presented in aggregated manner grouped by Solomon class (R1, C1, RC1, R2, C2, RC2). The meaning of the column entries are identical to those of Table 1.

			Unlimited Fleet						Limited Fleet				
Instances						±%						±%	
ρ	n	Class	#opt	Time	B&B	$\sum c_P$	$\sum f_P$	Fleet	#opt	Time	B&B	$\sum c_P$	$\sum f_P$
0.001	25	R1	12/12	1.3	1.0	0.6	4.6	5.8	12/12	0.2	1.0	0.4	5.0
		C1	9/9	19.1	2.3	21.7	19.4	33.3	8/9	473.6	1.0	7.5	55.8
		RC1	8/8	6.8	1.0	0.8	4.7	0.0	8/8	2.9	1.0	0.8	4.7
		R2	11/11	263.0	1.9	16.1	7.3	50.0	10/11	707.8	0.9	12.9	11.4
		C2	8/8	30.4	1.0	24.6	15.1	93.8	7/8	736.1	0.9	9.3	45.5
		RC2	8/8	22.1	1.0	10.7	4.6	6.3	8/8	254.6	1.0	10.8	6.5
		Total	56/56	63.5	1.4	11.9	9.0	30.7	53/56	357.1	1.0	6.6	19.4
	50	R1	9/12	1075.2	3.3	0.8	4.8	1.4	9/12	924.4	2.7	0.7	5.0
		C1	7/9	1175.5	3.7	18.3	15.0	31.4	6/9	1375.7	0.8	0.2	15.2
		RC1	8/8	1398.7	4.8	0.6	6.1	0.0	8/8	459.2	4.8	0.6	6.1
		R2	4/11	2518.7	0.6	10.2	5.6	11.3	5/11	2376.7	1.1	10.8	7.5
		C2	6/8	1531.1	1.5	13.0	5.5	33.3	1/8	3151.6	0.1	0.0	1.7
		RC2	6/8	1137.4	0.8	8.8	6.3	8.3	4/8	1805.1	0.5	6.3	5.6
		Total	40/56	1495.1	2.4	7.8	7.3	13.2	33/56	1659.7	1.7	2.8	7.5
	100	R1	3/12	2814.5	2.1	0.4	1.1	0.0	3/12	2716.5	1.5	0.4	1.1
		C1	4/9	2190.6	1.2	0.2	6.4	0.0	4/9	2060.1	0.6	0.2	6.4
		RC1	2/8	2905.2	0.5	0.1	1.8	3.3	2/8	2956.7	1.0	0.1	1.8
		R2	0/11	3600.0	n.a.	n.a.	n.a.	n.a.	0/11	3600.0	n.a.	n.a.	n.a.
		C2	1/8	3283.6	0.4	0.0	4.2	0.0	1/8	3277.6	0.1	0.0	4.2
		RC2	0/11	3600.0	n.a.	n.a.	n.a.	n.a.	0/11	3600.0	n.a.	n.a.	n.a.
		Total	10/56	3060.7	0.8	0.2	3.7	0.7	10/56	3025.3	0.6	0.2	3.7
	Total		106/168	1539.8	1.5	9.3	7.9	21.3	96/168	1680.7	1.1	4.6	13.7
0.01	25	R1	12/12	0.2	1.0	12.5	11.5	26.6	12/12	4.6	1.0	6.9	29.4
		C1	9/9	0.2	1.2	84.2	22.2	144.4	8/9	451.6	1.0	15.6	526.3
		RC1	8/8	9.0	1.3	11.0	23.4	8.3	8/8	8.9	1.0	6.1	29.5
		R2	11/11	0.8	1.0	33.0	7.3	86.4	9/11	842.2	0.8	33.3	19.7
		C2	8/8	0.8	1.5	68.0	18.9	287.5	7/8	873.2	0.9	18.5	422.0
		RC2	8/8	1.5	1.0	19.2	9.7	6.3	8/8	237.1	1.0	22.9	17.2
		Total	56/56	1.8	1.1	36.8	14.9	89.0	52/56	398.9	0.9	16.7	155.2
	50	R1	12/12	98.2	4.3	17.6	14.3	36.2	8/12	1309.9	1.8	8.1	20.1
		C1	9/9	2.6	6.1	100.1	26.2	177.8	5/9	1662.3	0.7	4.8	82.7
		RC1	4/8	1865.6	16.5	7.4	8.4	10.3	4/8	2251.0	32.4	5.3	19.1
		R2	11/11	268.4	1.0	39.0	11.6	145.8	3/11	2871.5	0.3	22.9	4.6
		C2	8/8	43.1	1.5	65.5	21.0	325.0	2/8	2974.4	0.3	6.9	3.8
		RC2	7/8	937.4	1.3	23.5	9.7	16.7	4/8	1827.2	0.5	16.5	4.7
		Total	51/56	480.8	4.9	44.3	15.7	125.4	26/56	2119.4	5.3	10.0	26.6
	100	R1	4/12	2536.7	3.3	4.2	6.3	6.6	3/12	2756.8	1.4	2.5	5.0
		C1	7/9	1083.5	164.2	67.4	26.1	118.6	4/9	2085.7	0.4	2.5	54.2
		RC1	2/8	2795.3	5.9	5.1	6.1	10.0	1/8	3198.2	0.5	0.1	5.9
		R2	1/11	3337.0	0.3	13.1	5.2	37.5	0/11	3600.0	n.a.	n.a.	n.a.
		C2	4/8	2177.1	47.3	80.2	29.6	375.0	1/8	3297.9	0.1	5.4	31.7
		RC2	1/8	3202.6	0.1	12.2	3.3	33.3	0/8	3600.0	n.a.	n.a.	n.a.
		Total	19/56	2541.0	34.8	44.5	18.3	128.8	9/56	3075.4	0.5	2.5	29.9
	Total		126/168	1007.9	13.6	41.0	15.8	109.7	87/168	1864.6	2.2	13.2	103.8

Table 4: Aggregated results over 672 VRPTW-CNC instances.

Instances			Unlimited Fleet						Limited Fleet				
						±%						±%	
ρ	n	Class	#opt	Time	B&B	$\sum c_P$	$\sum f_P$	Fleet	#opt	Time	B&B	$\sum c_P$	$\sum f_P$
0.05	25	R1	12/12	0.1	1.0	30.3	14.0	67.4	12/12	14.2	1.2	10.9	134.8
		C1	9/9	0.1	1.0	119.6	7.2	188.9	8/9	430.8	1.0	23.4	2616.3
		RC1	8/8	1.2	1.0	49.3	20.1	71.9	8/8	7.8	1.0	8.9	140.3
		R2	11/11	0.3	1.2	42.6	10.4	158.3	9/11	890.8	0.8	39.2	79.7
		C2	8/8	0.5	1.0	96.4	1.4	356.3	6/8	934.3	0.8	18.9	510.6
		RC2	8/8	1.4	1.0	34.7	9.6	31.3	7/8	477.9	1.0	18.3	47.6
		Total	56/56	0.5	1.0	59.9	10.6	141.5	50/56	450.1	1.0	19.7	555.7
	50	R1	12/12	3.8	1.3	38.2	19.2	84.1	9/12	1551.2	3.5	13.0	95.9
		C1	9/9	0.5	1.0	140.0	8.3	233.3	5/9	1674.6	0.6	8.7	403.5
		RC1	8/8	201.2	3.3	42.4	25.2	60.9	4/8	2046.9	14.8	8.3	58.2
		R2	11/11	20.3	1.0	55.8	17.7	241.2	2/11	2998.1	0.2	20.5	12.4
		C2	8/8	153.4	1.0	96.8	0.8	420.8	0/8	3600.0	n.a.	n.a.	n.a.
		RC2	7/8	492.0	1.0	42.8	24.5	54.0	4/8	2060.5	0.5	19.5	14.1
		Total	55/56	125.8	1.4	68.1	16.0	181.7	24/56	2291.5	3.1	13.0	133.1
	100	R1	10/12	1141.5	6.3	42.5	23.8	84.9	3/12	2861.2	1.5	6.4	16.9
		C1	9/9	12.4	1.0	136.4	9.5	218.9	4/9	2516.8	0.4	4.1	267.6
		RC1	7/8	1446.6	2.3	37.6	18.5	65.6	1/8	3233.5	0.4	5.2	17.8
		R2	4/11	2667.4	0.6	85.1	35.0	315.6	0/11	3600.0	n.a.	n.a.	n.a.
		C2	4/8	1837.1	0.8	325.6	0.3	1250.0	1/8	3298.0	0.1	5.4	158.4
		RC2	3/8	2557.0	0.6	28.6	7.2	60.5	0/8	3600.0	n.a.	n.a.	n.a.
		Total	37/56	1604.9	2.1	98.5	16.6	262.8	9/56	3172.1	0.5	5.1	144.2
	Total		148/168	577.1	1.5	72.6	14.1	186.8	83/168	1971.2	1.5	16.2	388.9
0.1	25	R1	12/12	0.0	1.0	37.1	17.8	91.0	12/12	11.6	2.0	13.5	266.0
		C1	9/9	0.1	1.0	128.0	0.9	200.0	8/9	432.9	1.0	27.8	5225.8
		RC1	8/8	0.5	1.0	57.9	27.3	84.4	8/8	10.5	1.0	9.5	279.7
		R2	11/11	0.3	1.0	47.4	12.7	190.9	9/11	1013.4	0.8	40.1	158.3
		C2	8/8	1.0	1.0	97.6	1.0	362.5	6/8	922.4	0.8	18.9	1021.2
		RC2	8/8	1.3	1.0	34.7	19.2	31.3	8/8	466.7	1.0	31.7	144.8
		Total	56/56	0.5	1.0	65.0	13.2	157.4	51/56	471.1	1.1	23.3	1097.0
	50	R1	12/12	1.6	1.0	50.8	20.5	112.8	9/12	1828.2	8.2	14.9	189.2
		C1	9/9	0.2	1.0	147.9	2.9	244.4	5/9	1664.1	0.6	10.3	804.0
		RC1	8/8	158.4	16.3	55.8	31.2	80.1	3/8	2357.0	50.1	10.0	93.2
		R2	11/11	13.9	1.0	67.5	18.4	304.5	2/11	3032.8	0.2	20.5	24.9
		C2	7/8	555.2	0.9	110.6	0.9	481.0	1/8	3245.9	0.1	8.3	10.6
		RC2	8/8	399.8	2.5	59.4	26.9	85.2	4/8	2127.6	0.5	22.3	23.2
		Total	55/56	162.2	3.4	79.6	17.2	210.8	24/56	2359.3	9.1	14.8	256.5
	100	R1	11/12	483.1	6.3	58.6	24.3	119.5	3/12	2954.2	2.6	7.6	31.9
		C1	9/9	7.2	1.0	144.3	7.5	230.0	3/9	2629.6	0.3	2.6	474.0
		RC1	4/8	1831.2	19.3	26.8	14.8	54.8	1/8	3214.6	0.4	8.2	31.1
		R2	8/11	1526.1	0.9	91.1	33.2	340.8	0/11	3600.0	n.a.	n.a.	n.a.
		C2	6/8	1338.8	0.8	234.1	0.7	916.7	1/8	3483.2	0.1	5.4	316.9
		RC2	5/8	2300.5	1.1	65.7	24.7	193.5	0/8	3600.0	n.a.	n.a.	n.a.
		Total	43/56	1186.0	4.7	104.9	18.3	297.6	8/56	3233.9	0.7	5.5	233.2
	Total		154/168	449.5	3.0	81.4	16.1	215.6	83/168	2021.4	3.6	19.1	770.7

Table 5: Aggregated results over 672 VRPTW-CNC instances.

D. Detailed Computational Results

The following 24 Tables 6–29 provide results per instance. Columns with headers identical to those mentioned in the manuscript have identical meaning. The additional columns have the following meaning:
Opt: Optimal solution value (=routing cost plus inconvenience cost);
UB: Upper bound resulting from a possibly not optimal integer solution;
LB_{root}: Lower bound resulting from the linear programming relaxation after adding cuts;
LB_{tree}: Lower bound resulting from branch-and-bound;
%*Gap*: Remaining gap in percent, i.e., $100 \cdot (UB - LB_{tree})/LB_{tree}$;
2-path: Number of separated 2-path cuts
SR: Number of separated subset-row inequalities

	VRPTW-CNC with Unlimited Fleet												VRPTW	
	Bounds							Cuts		Costs and Fleet				
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	Fleet	c^*	K^*
R101	617.21		617.21		0.000	0.1	1	0	0	617.1	0.11	8	617.1	8
R102	570.20		570.20		0.000	0.2	1	5	0	550.9	19.30	7	547.1	7
R103	490.88		490.88		0.000	1.0	1	6	10	463.5	27.38	6	454.6	5
R104	455.12		455.12		0.000	0.9	1	0	0	416.9	38.22	4	416.9	4
R105	532.06		532.06		0.000	0.2	1	0	0	530.7	1.36	6	530.5	6
R106	489.29		489.29		0.000	0.4	1	6	0	473.3	15.99	5	465.4	5
R107	461.95		461.95		0.000	1.1	1	9	0	430.8	31.15	5	424.3	4
R108	434.51		434.51		0.000	3.0	1	0	0	399.2	35.31	4	397.3	4
R109	448.38		448.38		0.000	0.4	1	0	0	441.3	7.08	5	441.3	5
R110	454.59		454.59		0.000	3.3	1	0	20	444.1	10.48	5	444.1	5
R111	458.37		458.37		0.000	3.1	1	7	19	430.1	28.27	5	428.8	4
R112	415.46		415.46		0.000	1.8	1	0	0	393.0	22.46	4	393.0	4
C101	193.51		193.51		0.000	0.4	1	0	0	191.3	2.21	3	191.3	3
C102	300.46		299.02		0.000	7.0	3	0	9	262.8	37.65	5	190.3	3
C103	385.49		384.06		0.000	77.0	3	0	25	327.6	57.89	6	190.3	3
C104	432.25		432.25		0.000	22.4	1	0	3	345.9	86.35	7	186.9	3
C105	200.09		200.09		0.000	0.6	1	0	0	191.3	8.79	3	191.3	3
C106	193.52		193.52		0.000	0.4	1	0	0	191.3	2.22	3	191.3	3
C107	211.21		211.21		0.000	0.9	1	0	0	191.3	19.91	3	191.3	3
C108	226.63		226.11		0.000	4.5	3	0	0	191.3	35.33	3	191.3	3
C109	272.83		271.41		0.000	58.7	7	0	29	191.3	81.53	3	191.3	3
RC101	463.52		463.52		0.000	0.7	1	14	0	461.1	2.42	4	461.1	4
RC102	364.32		364.32		0.000	0.5	1	0	0	352.0	12.32	3	351.8	3
RC103	356.87		356.87		0.000	1.8	1	0	0	336.0	20.87	3	332.8	3
RC104	347.48		347.48		0.000	11.2	1	0	0	319.0	28.48	3	306.6	3
RC105	424.48		424.48		0.000	0.4	1	0	0	412.5	11.98	4	411.3	4
RC106	352.58		352.58		0.000	0.7	1	0	0	345.5	7.08	3	345.5	3
RC107	315.68		315.68		0.000	5.1	1	0	0	298.8	16.88	3	298.3	3
RC108	317.20		317.20		0.000	33.7	1	0	0	296.5	20.70	3	294.5	3
R201	483.87		483.87		0.000	1.1	1	0	6	470.2	13.67	4	463.3	4
R202	487.80		487.80		0.000	1.3	1	0	0	462.9	24.90	4	410.5	4
R203	479.30		476.14		0.000	52.0	7	0	20	461.0	18.30	4	391.4	3
R204	460.39		460.39		0.000	1540.8	1	0	69	418.5	41.89	3	355.0	2
R205	461.44		461.44		0.000	2.3	1	0	6	444.7	16.74	4	393.0	3
R206	474.20		474.20		0.000	3.0	1	0	0	431.7	42.50	3	374.4	3
R207	474.18		471.89		0.000	190.6	3	0	26	449.5	24.68	4	361.6	3
R208	455.40		455.40		0.000	1053.6	1	0	40	410.1	45.30	3	328.2	1
R209	454.81		454.79		0.000	16.9	3	0	20	427.4	27.41	4	370.7	2
R210	480.57		480.57		0.000	22.0	1	0	20	454.5	26.07	3	404.6	3
R211	444.54		444.54		0.000	9.2	1	0	0	427.4	17.14	4	350.9	2
C201	219.03		219.03		0.000	1.8	1	0	0	214.7	4.33	2	214.7	2
C202	335.49		335.49		0.000	87.8	1	0	60	312.6	22.89	4	214.7	2
C203	413.68		413.68		0.000	16.9	1	0	3	349.1	64.58	5	214.7	2
C204	474.57		474.57		0.000	5.3	1	0	0	371.2	103.37	6	213.1	1
C205	227.64		227.64		0.000	4.1	1	0	0	214.7	12.94	2	214.7	2
C206	231.49		231.49		0.000	8.2	1	0	0	223.8	7.69	2	214.7	2
C207	244.03		244.03		0.000	32.5	1	0	0	235.5	8.53	2	214.5	2
C208	249.45		249.45		0.000	86.9	1	0	30	214.7	34.75	2	214.5	2
RC201	378.76		378.76		0.000	0.6	1	0	0	363.1	15.66	3	360.2	3
RC202	367.29		367.29		0.000	3.8	1	0	0	352.7	14.59	3	338.0	3
RC203	357.86		357.86		0.000	19.7	1	0	0	343.7	14.16	3	326.9	3
RC204	342.73		342.73		0.000	33.1	1	0	0	326.5	16.23	3	299.7	3
RC205	371.64		371.64		0.000	1.0	1	0	0	359.9	11.74	3	338.0	3
RC206	376.45		376.45		0.000	1.0	1	0	0	363.1	13.35	3	324.0	3
RC207	367.81		367.81		0.000	15.0	1	0	0	354.3	13.51	3	298.3	3
RC208	363.96		363.96		0.000	102.4	1	0	0	347.3	16.66	3	269.1	2

Table 6: Detailed results for the VRPTW-CNC with $\rho = 0.001$ for the $n = 25$-customer instances.

	VRPTW-CNC with Unlimited Fleet												VRPTW	
	Bounds							Cuts		Costs and Fleet				
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	%*Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	Fleet	c^*	K^*
R101	1044.19		1044.19		0.000	0.6	1	3	0	1044.0	0.19	12	1044.0	12
R102	963.32		963.32		0.000	5.3	1	5	10	928.6	34.72	11	909.0	11
R103	839.55		838.96		0.000	518.5	3	3	61	789.8	49.75	9	772.9	9
R104			740.48	740.48		3600.0	1	8	30				625.4	6
R105	903.19		897.80		0.000	16.6	15	6	3	899.3	3.89	9	899.3	9
R106	851.78		850.05		0.000	76.3	3	2	34	804.7	47.08	9	793.0	8
R107	792.01		792.01		0.000	188.9	1	0	20	717.4	74.61	7	711.1	7
R108			726.26	726.26		3600.0	1	4	10				617.7	6
R109	803.26		800.45		0.000	194.4	9	5	55	786.9	16.36	8	786.8	8
R110	737.99		737.99		0.000	331.3	1	9	30	698.0	39.99	7	697.0	7
R111	768.85		768.76		0.000	770.1	3	2	34	711.4	57.45	7	707.2	7
R112			693.34	693.34		3600.0	1	2	30				630.2	6
C101	367.50		367.50		0.000	4.6	1	0	0	362.4	5.10	5	362.4	5
C102	609.67		608.24		0.000	484.9	7	0	67	516.8	92.87	9	361.4	5
C103	789.15		787.71		0.000	2195.7	9	0	50	665.0	124.15	12	361.4	5
C104			1013.99	1015.29		3600.0	9	0	18				358.0	5
C105	382.73		382.73		0.000	7.5	1	0	0	362.4	20.33	5	362.4	5
C106	379.66		379.66		0.000	5.8	1	0	0	365.0	14.66	5	362.4	5
C107	402.56		402.56		0.000	17.9	1	0	0	362.4	40.16	5	362.4	5
C108	447.79		446.42		0.000	662.8	3	0	20	365.0	82.79	5	362.4	5
C109			515.41	515.41		3600.0	1	0	90				362.4	5
RC101	947.85		947.85		0.000	3.3	1	65	0	944.0	3.85	8	944.0	8
RC102	857.08		857.08		0.000	408.5	1	24	76	826.7	30.38	7	822.5	7
RC103	780.38		780.38		0.000	1657.5	1	8	40	716.3	64.07	6	710.9	6
RC104	635.71		635.71		0.000	2062.2	1	0	0	559.4	76.31	5	545.8	5
RC105	879.33		879.33		0.000	43.7	1	31	20	857.4	21.93	8	855.3	8
RC106	741.74		741.74		0.000	31.1	1	7	10	723.4	18.34	6	723.2	6
RC107	689.71		684.41		0.000	3550.6	31	5	10	645.8	43.91	6	642.7	6
RC108	658.95		658.95		0.000	3433.1	1	3	20	599.4	59.55	6	598.1	6
R201	825.19		825.19		0.000	5.0	1	0	0	793.7	31.48	6	791.9	6
R202	819.34		819.34		0.000	897.3	1	0	40	793.5	25.84	6	698.5	5
R203			798.41	798.41		3600.0	1	0	10				605.3	5
R204						3600.0	0	0	0				506.4	2
R205	804.06		804.06		0.000	594.5	1	0	60	762.1	41.96	5	690.1	4
R206			794.52	794.52		3600.0	1	0	60				632.4	4
R207						3600.0	0	0	0				575.5	3
R208						3600.0	0	0	0				487.7	2
R209			781.51	781.51		3600.0	1	0	57				600.6	4
R210	808.12		808.12		0.000	1009.2	1	0	0	752.3	55.82	4	645.6	4
R211						3600.0	0	0	0				535.5	3
C201	366.44		366.44		0.000	32.3	1	0	0	360.2	6.24	3	360.2	3
C202	602.71		602.71		0.000	1008.3	1	0	0	535.4	67.31	6	360.2	3
C203			812.59	812.59		3600.0	1	0	100				359.8	3
C204			1106.73	1107.48		3600.0	5	0	19				588.7	2
C205	378.06		378.06		0.000	380.5	1	0	30	373.4	4.66	3	359.8	3
C206	384.27		384.27		0.000	527.6	1	0	0	377.6	6.67	3	359.8	3
C207	404.27		404.27		0.000	1492.4	1	0	0	395.1	9.17	3	359.6	3
C208	412.88		412.88		0.000	1607.5	1	0	10	389.1	23.78	4	350.5	2
RC201	720.25		720.25		0.000	7.8	1	0	0	695.0	25.25	5	684.8	5
RC202	708.65		708.65		0.000	201.3	1	0	0	682.2	26.45	5	613.6	5
RC203	670.43		670.43		0.000	1200.8	1	0	0	634.9	35.53	5	555.3	4
RC204						3600.0	0	0	0				444.2	3
RC205	707.44		707.44		0.000	66.4	1	0	0	665.7	41.74	5	630.2	5
RC206	700.57		700.57		0.000	49.7	1	0	0	652.6	47.97	5	610.0	5
RC207	680.38		680.38		0.000	373.5	1	0	0	630.6	49.78	5	558.6	4
RC208						3600.0	0	0	0				476.7	3

Table 7: Detailed results for the VRPTW-CNC with $\rho = 0.001$ for the $n = 50$-customer instances.

	VRPTW-CNC with Unlimited Fleet													
	Bounds							Cuts		Costs and Fleet			VRPTW	
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	Fleet	c^*	K^*
R101	1638.32		1634.59		0.000	52.9	13	5	7	1637.7	0.62	20	1637.7	20
R102	1522.14		1522.14		0.000	205.3	1	0	0	1484.2	37.94	18	1466.6	18
R103						3600.0	0	0	0				1208.7	14
R104						3600.0	0	0	0				971.5	11
R105	1363.92		1363.35		0.000	1115.5	9	10	63	1355.3	8.62	15	1355.3	15
R106			1316.82	1316.82		3600.0	1	4	10				1234.6	13
R107						3600.0	0	0	0				1064.6	11
R108						3600.0	0	0	0				932.1	10
R109			1174.33	1174.33		3600.0	1	5	20				1146.9	13
R110						3600.0	0	0	0				1068.0	12
R111						3600.0	0	0	0				1048.7	12
R112						3600.0	0	0	0				948.6	10
C101	837.74		837.74		0.000	36.1	1	0	0	827.3	10.44	10	827.3	10
C102			1322.94	1322.94		3600.0	1	0	80				827.3	10
C103			1734.12	1734.12		3600.0	1	0	20				826.3	10
C104			2161.16	2161.16		3600.0	1	0	10				822.9	10
C105	868.90		868.90		0.000	111.4	1	0	0	827.3	41.60	10	827.3	10
C106	913.00		912.72		0.000	1436.8	3	0	30	833.1	79.90	10	827.3	10
C107	908.63		908.63		0.000	131.6	1	0	0	827.3	81.33	10	827.3	10
C108			988.10	988.10		3600.0	1	0	50				827.3	10
C109			1119.10	1119.10		3600.0	1	0	10				827.3	10
RC101	1629.64		1629.64		0.000	230.5	1	78	39	1619.8	9.84	15	1619.8	15
RC102			1525.14	1525.14		3600.0	1	35	20				1457.4	14
RC103						3600.0	0	0	0				1258.0	11
RC104						3600.0	0	0	0				1132.3	10
RC105	1561.72		1561.72		0.000	1411.4	1	28	40	1516.0	45.72	16	1513.7	15
RC106			1382.45	1382.45		3600.0	1	16	20				1372.7	12
RC107						3600.0	0	0	0				1207.8	12
RC108						3600.0	0	0	0				1114.2	11
R201						3600.0	0	0	0				1143.2	8
R202						3600.0	0	0	0				1029.6	8
R203						3600.0	0	0	0				870.8	6
R204						3600.0	0	0	0				731.3	5
R205						3600.0	0	0	0				949.8	5
R206						3600.0	0	0	0				875.9	5
R207						3600.0	0	0	0				794.0	4
R208						3600.0	0	0	0				701.0	4
R209						3600.0	0	0	0				854.8	5
R210						3600.0	0	0	0				900.5	6
R211						3600.0	0	0	0				746.7	4
C201	613.99		613.99		0.000	1068.5	1	0	0	589.1	24.89	3	589.1	3
C202						3600.0	0	0	0				589.1	3
C203						3600.0	0	0	0				588.7	3
C204						3600.0	0	0	0				588.1	3
C205			657.64	657.64		3600.0	1	0	10				586.4	3
C206						3600.0	0	0	0				586.4	3
C207			695.19	695.19		3600.0	1	0	10				585.8	3
C208						3600.0	0	0	0				585.8	3
RC201						3600.0	0	0	0				1261.8	9
RC202						3600.0	0	0	0				1092.3	8
RC203						3600.0	0	0	0				923.7	5
RC204						3600.0	0	0	0				783.5	4
RC205						3600.0	0	0	0				1154.0	7
RC206						3600.0	0	0	0				1051.1	7
RC207						3600.0	0	0	0				962.9	6
RC208						3600.0	0	0	0				776.1	4

Table 8: Detailed results for the VRPTW-CNC with $\rho = 0.001$ for the $n = 100$-customer instances.

	VRPTW-CNC with Unlimited Fleet												VRPTW	
	Bounds							Cuts		Costs and Fleet				
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	%*Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	Fleet	c^*	K^*
R101	618.20		618.20		0.000	0.0	1	0	0	617.1	1.10	8	617.1	8
R102	606.99		606.99		0.000	0.1	1	0	0	591.8	15.19	8	547.1	7
R103	589.54		589.54		0.000	0.0	1	0	0	538.2	51.34	7	454.6	5
R104	585.12		585.12		0.000	0.2	1	0	0	498.3	86.82	6	416.9	4
R105	544.30		544.30		0.000	0.0	1	0	0	530.7	13.60	6	530.5	6
R106	570.17		570.17		0.000	0.1	1	2	8	557.3	12.87	7	465.4	5
R107	579.07		579.07		0.000	0.1	1	0	7	513.4	65.67	6	424.3	4
R108	578.59		578.59		0.000	0.3	1	0	0	498.3	80.28	6	397.3	4
R109	505.31		505.31		0.000	0.1	1	0	0	449.3	56.01	5	441.3	5
R110	514.41		514.41		0.000	0.1	1	0	0	466.4	48.01	5	444.1	5
R111	563.34		563.34		0.000	0.2	1	0	10	516.5	46.84	6	428.8	4
R112	542.42		542.42		0.000	0.8	1	0	10	436.3	106.12	5	393.0	4
C101	213.38		213.38		0.000	0.1	1	0	0	191.3	22.08	3	191.3	3
C102	438.53		438.32		0.000	0.2	3	0	10	431.7	6.83	9	190.3	3
C103	616.81		616.81		0.000	0.3	1	0	7	531.7	85.11	12	190.3	3
C104	758.46		758.46		0.000	0.1	1	0	5	626.8	131.66	15	186.9	3
C105	271.93		271.93		0.000	0.2	1	0	20	211.3	60.63	4	191.3	3
C106	213.51		213.51		0.000	0.1	1	0	0	191.3	22.21	3	191.3	3
C107	298.81		298.81		0.000	0.2	1	0	6	298.8	0.01	6	191.3	3
C108	315.63		315.63		0.000	0.2	1	0	9	306.9	8.73	6	191.3	3
C109	403.71		403.71		0.000	0.1	1	0	0	362.3	41.41	8	191.3	3
RC101	485.34		485.34		0.000	0.1	1	14	0	461.1	24.24	4	461.1	4
RC102	448.45		448.06		0.000	1.6	3	0	0	360.1	88.35	3	351.8	3
RC103	479.61		479.61		0.000	2.2	1	0	20	358.6	121.01	3	332.8	3
RC104	498.07		498.07		0.000	14.8	1	0	20	396.3	101.77	4	306.6	3
RC105	508.85		508.85		0.000	0.8	1	0	30	448.0	60.85	4	411.3	4
RC106	398.88		398.88		0.000	0.3	1	0	0	359.6	39.28	3	345.5	3
RC107	414.24		414.24		0.000	5.6	1	0	20	327.6	86.64	3	298.3	3
RC108	466.69		466.69		0.000	46.8	1	0	34	371.3	95.38	4	294.5	3
R201	503.97		503.97		0.000	0.1	1	0	0	501.5	2.47	4	463.3	4
R202	534.74		534.74		0.000	0.1	1	0	0	526.9	7.84	4	410.5	4
R203	555.61		555.61		0.000	0.8	1	0	14	509.3	46.31	5	391.4	3
R204	556.07		556.07		0.000	1.1	1	0	8	487.0	69.07	5	355.0	2
R205	504.61		504.61		0.000	0.1	1	0	0	501.5	3.11	4	393.0	3
R206	534.74		534.74		0.000	0.2	1	0	0	526.9	7.84	4	374.4	3
R207	555.80		555.80		0.000	1.1	1	0	12	509.3	46.50	5	361.6	3
R208	556.30		556.30		0.000	0.8	1	0	10	487.0	69.30	5	328.2	1
R209	506.42		506.42		0.000	0.2	1	0	0	502.5	3.92	4	370.7	2
R210	527.03		527.03		0.000	1.1	1	0	10	519.0	8.03	3	404.6	3
R211	508.69		508.69		0.000	3.5	1	0	8	484.6	24.09	4	350.9	2
C201	234.01		234.01		0.000	0.3	1	0	0	234.0	0.01	3	214.7	2
C202	435.31		435.31		0.000	0.6	1	0	0	389.3	46.01	6	214.7	2
C203	665.71		665.71		0.000	1.0	1	0	28	532.2	133.51	10	214.7	2
C204	809.58		806.78		0.000	1.3	5	0	10	680.2	129.38	14	213.1	1
C205	255.81		255.81		0.000	0.3	1	0	0	255.8	0.01	4	214.7	2
C206	244.02		244.02		0.000	0.4	1	0	0	240.2	3.82	3	214.7	2
C207	262.91		262.91		0.000	2.0	1	0	0	262.9	0.01	3	214.5	2
C208	297.04		297.04		0.000	0.5	1	0	0	285.7	11.34	5	214.5	2
RC201	395.10		395.10		0.000	0.1	1	0	0	395.1	0.00	3	360.2	3
RC202	431.12		431.12		0.000	1.0	1	0	0	379.7	51.42	3	338.0	3
RC203	446.72		446.72		0.000	2.1	1	0	0	368.3	78.42	3	326.9	3
RC204	447.33		447.33		0.000	4.3	1	0	0	351.1	96.22	3	299.7	3
RC205	385.91		385.91		0.000	0.1	1	0	0	385.4	0.51	3	338.0	3
RC206	395.10		395.10		0.000	0.2	1	0	0	395.1	0.00	3	324.0	3
RC207	386.81		386.81		0.000	0.5	1	0	0	378.6	8.20	3	298.3	3
RC208	384.08		384.08		0.000	3.8	1	0	0	374.2	9.88	3	269.1	2

Table 9: Detailed results for the VRPTW-CNC with $\rho = 0.01$ for the $n = 25$-customer instances.

	VRPTW-CNC with Unlimited Fleet												VRPTW	
	Bounds							Cuts		Costs and Fleet				
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	Fleet	c^*	K^*
R101	1045.93		1045.93		0.000	0.1	1	3	0	1044.0	1.93	12	1044.0	12
R102	1035.52		1035.52		0.000	0.3	1	0	0	989.3	46.22	11	909.0	11
R103	1025.38		1025.38		0.000	6.9	1	0	10	909.4	115.98	11	772.9	9
R104	1049.19		1049.19		0.000	35.3	1	0	4	831.4	217.79	11	625.4	6
R105	938.17		934.93		0.000	1.3	5	3	2	899.3	38.86	9	899.3	9
R106	977.57		977.57		0.000	4.5	1	0	20	927.5	50.07	10	793.0	8
R107	997.19		993.31		0.000	270.5	31	0	60	884.4	112.79	10	711.1	7
R108	1036.75		1036.75		0.000	43.9	1	0	0	831.4	205.35	11	617.7	6
R109	908.40		908.40		0.000	6.5	1	4	30	822.8	85.60	10	786.8	8
R110	903.41		903.41		0.000	31.7	1	0	30	821.0	82.41	10	697.0	7
R111	977.13		976.73		0.000	153.6	7	0	60	872.2	104.93	10	707.2	7
R112	938.03		938.03		0.000	623.2	1	0	40	819.5	118.53	10	630.2	6
C101	413.36		413.36		0.000	0.4	1	0	0	362.4	50.96	5	362.4	5
C102	893.92		893.72		0.000	0.7	3	0	10	831.3	62.62	16	361.4	5
C103	1282.17		1282.17		0.000	0.8	1	0	5	1090.5	191.67	23	361.4	5
C104	1845.51		1840.11		0.000	9.8	45	0	6	1536.3	309.21	34	358.0	5
C105	522.96		522.96		0.000	4.6	1	0	50	418.9	104.06	7	362.4	5
C106	479.91		479.91		0.000	3.3	1	0	50	457.7	22.21	7	362.4	5
C107	563.42		563.42		0.000	1.7	1	0	17	563.4	0.02	10	362.4	5
C108	609.41		609.41		0.000	1.2	1	0	10	582.8	26.61	10	362.4	5
C109	741.72		741.72		0.000	0.7	1	0	0	659.8	81.92	13	362.4	5
RC101	979.86		979.86		0.000	0.7	1	62	0	945.1	34.76	8	944.0	8
RC102	997.84		991.02		0.000	360.7	27	19	100	946.2	51.64	9	822.5	7
RC103		1022.20	1012.37	1021.57	0.062	3600.0	41	6	100				710.9	6
RC104			1017.11	1017.11		3600.0	1	0	40				545.8	5
RC105	1002.69		1002.69		0.000	32.6	1	27	80	955.8	46.89	9	855.3	8
RC106	874.23		866.50		0.000	131.1	19	7	30	742.5	131.73	6	723.2	6
RC107			925.22	934.22		3600.0	41	3	100				642.7	6
RC108			981.12	981.12		3600.0	1	1	60				598.1	6
R201	880.39		880.39		0.000	0.7	1	0	0	849.8	30.59	7	791.9	6
R202	884.25		884.25		0.000	14.4	1	0	10	863.1	21.15	7	698.5	5
R203	915.81		915.81		0.000	116.5	1	0	20	859.2	56.61	9	605.3	5
R204	981.52		981.52		0.000	889.4	1	0	49	785.4	196.12	10	506.4	2
R205	875.53		875.53		0.000	5.1	1	0	0	849.1	26.43	7	690.1	4
R206	888.55		888.55		0.000	128.7	1	0	20	863.1	25.45	7	632.4	4
R207	916.00		916.00		0.000	203.2	1	0	20	859.2	56.80	9	575.5	3
R208	981.86		981.86		0.000	1059.1	1	0	49	785.4	196.46	10	487.7	2
R209	876.82		876.82		0.000	70.6	1	0	10	863.8	13.02	8	600.6	4
R210	882.43		882.43		0.000	77.0	1	0	0	852.2	30.23	6	645.6	4
R211	871.59		871.59		0.000	387.9	1	0	0	831.4	40.19	8	535.5	3
C201	373.42		373.42		0.000	14.5	1	0	0	373.4	0.02	3	360.2	3
C202	806.98		806.98		0.000	107.8	1	0	70	717.3	89.68	11	360.2	3
C203	1297.10		1297.10		0.000	8.3	1	0	29	1027.9	269.20	19	359.8	3
C204	1956.17		1956.15		0.000	5.5	5	0	9	1608.5	347.67	32	588.7	2
C205	395.22		395.22		0.000	9.1	1	0	0	395.2	0.02	4	359.8	3
C206	389.53		389.53		0.000	22.2	1	0	0	385.7	3.83	4	359.8	3
C207	418.74		418.74		0.000	92.5	1	0	0	414.2	4.54	4	359.6	3
C208	480.06		480.06		0.000	85.0	1	0	0	457.1	22.96	6	350.5	2
RC201	753.13		753.13		0.000	0.7	1	0	0	742.2	10.93	5	684.8	5
RC202	793.10		793.10		0.000	13.3	1	0	0	723.2	69.90	5	613.6	5
RC203	852.34		850.66		0.000	3080.4	3	0	20	671.7	180.64	5	555.3	4
RC204			946.28	946.28		3600.0	1	0	20				444.2	3
RC205	767.42		767.42		0.000	1.3	1	0	0	746.4	21.02	5	630.2	5
RC206	755.41		755.41		0.000	2.1	1	0	0	740.7	14.71	5	610.0	5
RC207	758.26		758.26		0.000	37.4	1	0	0	723.7	34.56	5	558.6	4
RC208	752.17		752.17		0.000	764.1	1	0	0	703.6	48.57	5	476.7	3

Table 10: Detailed results for the VRPTW-CNC with $\rho = 0.01$ for the $n = 50$-customer instances.

	VRPTW-CNC with Unlimited Fleet													
	Bounds							Cuts		Costs and Fleet			VRPTW	
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	Fleet	c^*	K^*
R101	1045.93		1045.93		0.000	0.1	1	3	0	1044.0	1.93	12	1044.0	12
R102	1035.52		1035.52		0.000	0.3	1	0	0	989.3	46.22	11	909.0	11
R103	1025.38		1025.38		0.000	6.9	1	0	10	909.4	115.98	11	772.9	9
R104	1049.19		1049.19		0.000	35.3	1	0	4	831.4	217.79	11	625.4	6
R105	938.17		934.93		0.000	1.3	5	3	2	899.3	38.86	9	899.3	9
R106	977.57		977.57		0.000	4.5	1	0	20	927.5	50.07	10	793.0	8
R107	997.19		993.31		0.000	270.5	31	0	60	884.4	112.79	10	711.1	7
R108	1036.75		1036.75		0.000	43.9	1	0	0	831.4	205.35	11	617.7	6
R109	908.40		908.40		0.000	6.5	1	4	30	822.8	85.60	10	786.8	8
R110	903.41		903.41		0.000	31.7	1	0	30	821.0	82.41	10	697.0	7
R111	977.13		976.73		0.000	153.6	7	0	60	872.2	104.93	10	707.2	7
R112	938.03		938.03		0.000	623.2	1	0	40	819.5	118.53	10	630.2	6
C101	413.36		413.36		0.000	0.4	1	0	0	362.4	50.96	5	362.4	5
C102	893.92		893.72		0.000	0.7	3	0	10	831.3	62.62	16	361.4	5
C103	1282.17		1282.17		0.000	0.8	1	0	5	1090.5	191.67	23	361.4	5
C104	1845.51		1840.11		0.000	9.8	45	0	6	1536.3	309.21	34	358.0	5
C105	522.96		522.96		0.000	4.6	1	0	50	418.9	104.06	7	362.4	5
C106	479.91		479.91		0.000	3.3	1	0	50	457.7	22.21	7	362.4	5
C107	563.42		563.42		0.000	1.7	1	0	17	563.4	0.02	10	362.4	5
C108	609.41		609.41		0.000	1.2	1	0	10	582.8	26.61	10	362.4	5
C109	741.72		741.72		0.000	0.7	1	0	0	659.8	81.92	13	362.4	5
RC101	979.86		979.86		0.000	0.7	1	62	0	945.1	34.76	8	944.0	8
RC102	997.84		991.02		0.000	360.7	27	19	100	946.2	51.64	9	822.5	7
RC103		1022.20	1012.37	1021.57	0.062	3600.0	41	6	100				710.9	6
RC104			1017.11	1017.11		3600.0	1	0	40				545.8	5
RC105	1002.69		1002.69		0.000	32.6	1	27	80	955.8	46.89	9	855.3	8
RC106	874.23		866.50		0.000	131.1	19	7	30	742.5	131.73	6	723.2	6
RC107			925.22	934.22		3600.0	41	3	100				642.7	6
RC108			981.12	981.12		3600.0	1	1	60				598.1	6
R201	880.39		880.39		0.000	0.7	1	0	0	849.8	30.59	7	791.9	6
R202	884.25		884.25		0.000	14.4	1	0	10	863.1	21.15	7	698.5	5
R203	915.81		915.81		0.000	116.5	1	0	20	859.2	56.61	9	605.3	5
R204	981.52		981.52		0.000	889.4	1	0	49	785.4	196.12	10	506.4	2
R205	875.53		875.53		0.000	5.1	1	0	0	849.1	26.43	7	690.1	4
R206	888.55		888.55		0.000	128.7	1	0	20	863.1	25.45	7	632.4	4
R207	916.00		916.00		0.000	203.2	1	0	20	859.2	56.80	9	575.5	3
R208	981.86		981.86		0.000	1059.1	1	0	49	785.4	196.46	10	487.7	2
R209	876.82		876.82		0.000	70.6	1	0	10	863.8	13.02	8	600.6	4
R210	882.43		882.43		0.000	77.0	1	0	0	852.2	30.23	6	645.6	4
R211	871.59		871.59		0.000	387.9	1	0	0	831.4	40.19	8	535.5	3
C201	373.42		373.42		0.000	14.5	1	0	0	373.4	0.02	3	360.2	3
C202	806.98		806.98		0.000	107.8	1	0	70	717.3	89.68	11	360.2	3
C203	1297.10		1297.10		0.000	8.3	1	0	29	1027.9	269.20	19	359.8	3
C204	1956.17		1956.15		0.000	5.5	5	0	9	1608.5	347.67	32	588.7	2
C205	395.22		395.22		0.000	9.1	1	0	0	395.2	0.02	4	359.8	3
C206	389.53		389.53		0.000	22.2	1	0	0	385.7	3.83	4	359.8	3
C207	418.74		418.74		0.000	92.5	1	0	0	414.2	4.54	4	359.6	3
C208	480.06		480.06		0.000	85.0	1	0	0	457.1	22.96	6	350.5	2
RC201	753.13		753.13		0.000	0.7	1	0	0	742.2	10.93	5	684.8	5
RC202	793.10		793.10		0.000	13.3	1	0	0	723.2	69.90	5	613.6	5
RC203	852.34		850.66		0.000	3080.4	3	0	20	671.7	180.64	5	555.3	4
RC204			946.28	946.28		3600.0	1	0	20				444.2	3
RC205	767.42		767.42		0.000	1.3	1	0	0	746.4	21.02	5	630.2	5
RC206	755.41		755.41		0.000	2.1	1	0	0	740.7	14.71	5	610.0	5
RC207	758.26		758.26		0.000	37.4	1	0	0	723.7	34.56	5	558.6	4
RC208	752.17		752.17		0.000	764.1	1	0	0	703.6	48.57	5	476.7	3

Table 11: Detailed results for the VRPTW-CNC with $\rho = 0.01$ for the $n = 100$-customer instances.

	VRPTW-CNC with Unlimited Fleet													
	Bounds							Cuts		Costs and Fleet			VRPTW	
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	Fleet	c^*	K^*
R101	622.59		622.59		0.000	0.0	1	0	0	617.1	5.49	8	617.1	8
R102	654.72		654.72		0.000	0.0	1	0	0	619.3	35.42	9	547.1	7
R103	693.00		693.00		0.000	0.0	1	0	5	624.3	68.70	9	454.6	5
R104	733.07		733.07		0.000	0.1	1	0	7	623.2	109.87	9	416.9	4
R105	593.84		593.84		0.000	0.0	1	0	0	538.0	55.84	6	530.5	6
R106	621.43		621.43		0.000	0.0	1	0	0	565.5	55.93	7	465.4	5
R107	687.87		687.87		0.000	0.1	1	0	5	619.8	68.07	9	424.3	4
R108	731.26		731.26		0.000	0.1	1	0	7	623.2	108.06	9	397.3	4
R109	612.51		612.51		0.000	0.1	1	0	0	537.7	74.80	6	441.3	5
R110	593.58		593.58		0.000	0.1	1	0	0	550.9	42.68	7	444.1	5
R111	656.66		656.66		0.000	0.1	1	0	2	621.0	35.66	9	428.8	4
R112	647.82		647.82		0.000	0.6	1	0	8	577.1	70.72	8	393.0	4
C101	284.95		284.95		0.000	0.3	1	0	29	241.5	43.45	5	191.3	3
C102	439.20		439.20		0.000	0.0	1	0	0	439.2	0.00	9	190.3	3
C103	678.60		678.60		0.000	0.0	1	0	0	678.6	0.00	14	190.3	3
C104	838.50		838.50		0.000	0.0	1	0	0	838.5	0.00	18	186.9	3
C105	298.84		298.84		0.000	0.1	1	0	0	298.8	0.04	6	191.3	3
C106	283.81		283.81		0.000	0.5	1	0	40	211.3	72.51	4	191.3	3
C107	298.84		298.84		0.000	0.1	1	0	0	298.8	0.04	6	191.3	3
C108	335.53		335.53		0.000	0.1	1	0	0	328.2	7.33	7	191.3	3
C109	420.43		420.43		0.000	0.1	1	0	0	420.4	0.03	9	191.3	3
RC101	533.56		533.56		0.000	0.1	1	14	0	503.6	29.96	5	461.1	4
RC102	590.37		590.37		0.000	1.0	1	0	60	543.6	46.76	6	351.8	3
RC103	622.84		622.84		0.000	0.8	1	0	20	540.3	82.54	6	332.8	3
RC104	643.71		643.71		0.000	0.3	1	0	0	526.5	117.21	6	306.6	3
RC105	572.54		572.54		0.000	0.3	1	0	30	546.8	25.74	6	411.3	4
RC106	526.75		526.75		0.000	1.9	1	0	50	427.9	98.85	4	345.5	3
RC107	538.00		538.00		0.000	3.1	1	0	40	469.3	68.70	5	298.3	3
RC108	597.42		597.42		0.000	1.7	1	0	10	537.4	60.02	6	294.5	3
R201	513.34		513.34		0.000	0.1	1	0	0	502.5	10.84	4	463.3	4
R202	554.49		554.49		0.000	0.1	1	0	0	534.9	19.59	5	410.5	4
R203	627.70		627.43		0.000	0.3	3	0	3	574.6	53.10	8	391.4	3
R204	678.22		678.22		0.000	0.2	1	0	9	580.4	97.82	8	355.0	2
R205	515.56		515.56		0.000	0.1	1	0	0	509.7	5.86	4	393.0	3
R206	554.50		554.50		0.000	0.2	1	0	0	534.9	19.60	5	374.4	3
R207	627.71		627.71		0.000	0.2	1	0	3	574.6	53.11	8	361.6	3
R208	678.38		678.38		0.000	0.2	1	0	9	580.4	97.98	8	328.2	1
R209	521.56		521.56		0.000	0.1	1	0	0	503.5	18.06	4	370.7	2
R210	531.13		531.13		0.000	0.4	1	0	0	526.9	4.23	4	404.6	3
R211	543.32		543.32		0.000	1.4	1	0	0	510.9	32.42	5	350.9	2
C201	234.04		234.04		0.000	0.3	1	0	0	234.0	0.04	3	214.7	2
C202	456.79		456.79		0.000	0.5	1	0	0	441.1	15.69	7	214.7	2
C203	727.10		727.10		0.000	0.1	1	0	0	725.5	1.60	13	214.7	2
C204	877.90		877.90		0.000	0.0	1	0	0	876.3	1.60	17	213.1	1
C205	255.84		255.84		0.000	0.3	1	0	0	255.8	0.04	4	214.7	2
C206	249.82		249.82		0.000	0.8	1	0	0	244.9	4.92	3	214.7	2
C207	262.93		262.93		0.000	1.4	1	0	0	262.9	0.03	3	214.5	2
C208	325.43		325.43		0.000	0.4	1	0	0	325.4	0.03	6	214.5	2
RC201	395.10		395.10		0.000	0.1	1	0	0	395.1	0.00	3	360.2	3
RC202	479.31		479.31		0.000	0.3	1	0	0	455.2	24.11	4	338.0	3
RC203	558.08		558.08		0.000	0.7	1	0	0	489.5	68.57	5	326.9	3
RC204	618.85		618.85		0.000	2.9	1	0	10	531.8	87.05	6	299.7	3
RC205	387.95		387.95		0.000	0.2	1	0	0	385.4	2.55	3	338.0	3
RC206	395.10		395.10		0.000	0.1	1	0	0	395.1	0.00	3	324.0	3
RC207	408.96		408.96		0.000	0.3	1	0	0	385.0	23.96	3	298.3	3
RC208	409.16		409.16		0.000	6.5	1	0	0	380.1	29.06	3	269.1	2

Table 12: Detailed results for the VRPTW-CNC with $\rho = 0.05$ for the $n = 25$-customer instances.

	VRPTW-CNC with Unlimited Fleet													
	Bounds							Cuts		Costs and Fleet			VRPTW	
Instance	Opt	UB	LB_{root}	LB_{tree}	$\%Gap$	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	Fleet	c^*	K^*
R101	1053.65		1053.65		0.000	0.1	1	0	0	1044.0	9.65	12	1044.0	12
R102	1135.00		1135.00		0.000	0.5	1	0	12	1046.0	89.00	13	909.0	11
R103	1223.83		1223.83		0.000	0.5	1	0	0	1032.9	190.93	15	772.9	9
R104	1414.91		1414.91		0.000	0.6	1	0	0	1119.6	295.31	18	625.4	6
R105	1025.57		1025.57		0.000	0.6	1	0	16	983.1	42.47	11	899.3	9
R106	1106.45		1106.45		0.000	1.7	1	0	19	1025.7	80.75	12	793.0	8
R107	1191.62		1191.62		0.000	1.0	1	0	0	1011.4	180.22	15	711.1	7
R108	1403.01		1403.01		0.000	10.0	1	0	15	1119.6	283.41	18	617.7	6
R109	1050.15		1050.15		0.000	1.6	1	0	10	963.6	86.55	12	786.8	8
R110	1067.18		1067.18		0.000	4.0	1	0	16	961.0	106.18	13	697.0	7
R111	1129.80		1128.09		0.000	10.3	5	0	13	1061.6	68.20	13	707.2	7
R112	1158.95		1158.95		0.000	14.4	1	0	0	993.1	165.85	13	630.2	6
C101	546.73		546.73		0.000	1.6	1	0	39	449.1	97.63	8	362.4	5
C102	933.80		933.80		0.000	0.1	1	0	0	933.8	0.00	18	361.4	5
C103	1404.26		1404.26		0.000	0.1	1	0	0	1403.5	0.76	28	361.4	5
C104	2031.31		2031.31		0.000	0.1	1	0	0	1992.3	39.01	41	358.0	5
C105	563.48		563.48		0.000	0.2	1	0	0	563.4	0.08	10	362.4	5
C106	550.26		550.26		0.000	1.0	1	0	30	477.7	72.56	8	362.4	5
C107	563.48		563.48		0.000	0.2	1	0	0	563.4	0.08	10	362.4	5
C108	693.32		693.32		0.000	0.5	1	0	0	631.9	61.42	12	362.4	5
C109	781.88		781.88		0.000	0.6	1	0	0	781.8	0.08	15	362.4	5
RC101	1046.67		1046.67		0.000	0.6	1	52	0	977.8	68.87	9	944.0	8
RC102	1127.70		1127.70		0.000	3.1	1	8	30	986.1	141.60	10	822.5	7
RC103	1248.62		1248.62		0.000	27.2	1	0	20	983.8	264.82	10	710.9	6
RC104	1477.77		1473.51		0.000	1424.6	19	0	55	1097.5	380.27	12	545.8	5
RC105	1062.80		1062.80		0.000	1.4	1	20	4	1013.8	49.00	10	855.3	8
RC106	1066.62		1066.62		0.000	34.5	1	0	90	1002.8	63.82	10	723.2	6
RC107	1068.78		1068.78		0.000	17.7	1	0	30	997.2	71.58	10	642.7	6
RC108	1245.41		1245.41		0.000	100.8	1	0	10	980.9	264.52	10	598.1	6
R201	908.43		908.43		0.000	0.7	1	0	0	896.5	11.93	8	791.9	6
R202	949.19		949.19		0.000	3.4	1	0	0	898.1	51.09	9	698.5	5
R203	1079.26		1079.26		0.000	4.0	1	0	0	935.8	143.46	12	605.3	5
R204	1320.89		1320.89		0.000	3.4	1	0	0	1052.2	268.69	17	506.4	2
R205	911.52		911.52		0.000	2.4	1	0	0	896.1	15.42	8	690.1	4
R206	963.56		963.56		0.000	13.3	1	0	0	913.2	50.36	9	632.4	4
R207	1080.03		1080.03		0.000	5.3	1	0	0	935.8	144.23	12	575.5	3
R208	1321.42		1321.42		0.000	2.1	1	0	0	1052.2	269.22	17	487.7	2
R209	896.27		896.27		0.000	15.9	1	0	0	891.3	4.97	8	600.6	4
R210	932.25		932.25		0.000	53.3	1	0	0	901.4	30.85	7	645.6	4
R211	972.02		972.02		0.000	119.2	1	0	0	904.2	67.82	9	535.5	3
C201	373.50		373.50		0.000	617.6	1	0	0	373.4	0.10	3	360.2	3
C202	834.10		834.10		0.000	1.7	1	0	0	832.4	1.70	13	360.2	3
C203	1406.73		1406.73		0.000	0.2	1	0	0	1405.6	1.13	25	359.8	3
C204	2141.42		2141.42		0.000	0.2	1	0	0	2140.3	1.12	40	588.7	2
C205	395.30		395.30		0.000	17.8	1	0	0	395.2	0.10	4	359.8	3
C206	400.18		400.18		0.000	15.2	1	0	0	395.2	4.98	4	359.8	3
C207	433.14		433.14		0.000	541.7	1	0	30	418.4	14.74	4	359.6	3
C208	524.19		524.19		0.000	33.3	1	0	0	524.1	0.09	8	350.5	2
RC201	769.85		769.85		0.000	0.9	1	0	0	754.9	14.95	5	684.8	5
RC202	889.32		889.32		0.000	7.6	1	0	0	797.6	91.72	6	613.6	5
RC203	1106.40		1106.40		0.000	223.0	1	0	40	890.0	216.40	8	555.3	4
RC204	1391.89		1391.89		0.000	79.5	1	0	4	962.8	429.09	10	444.2	3
RC205	796.43		796.43		0.000	4.3	1	0	0	773.8	22.63	5	630.2	5
RC206	787.27		787.27		0.000	1.2	1	0	0	753.4	33.87	5	610.0	5
RC207	812.16		812.16		0.000	19.4	1	0	0	759.3	52.86	5	558.6	4
RC208			852.12	852.12		3600.0	1	0	20				476.7	3

Table 13: Detailed results for the VRPTW-CNC with $\rho = 0.05$ for the $n = 50$-customer instances.

	VRPTW-CNC with Unlimited Fleet													
	Bounds							Cuts		Costs and Fleet			VRPTW	
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	Fleet	c^*	K^*
R101	1666.29		1662.78		0.000	5.3	9	5	10	1641.0	25.29	21	1637.7	20
R102	1860.80		1860.80		0.000	27.0	1	0	6	1734.5	126.30	25	1466.6	18
R103	2074.25		2074.05		0.000	521.8	5	0	26	1761.0	313.25	27	1208.7	14
R104	2395.65		2395.29		0.000	916.2	5	0	25	1835.3	560.35	30	971.5	11
R105	1597.43		1596.68		0.000	194.0	15	0	54	1476.5	120.93	17	1355.3	15
R106	1800.03		1800.03		0.000	231.4	1	4	50	1600.3	199.73	22	1234.6	13
R107	2018.79		2018.34		0.000	1719.0	11	0	58	1661.6	357.19	25	1064.6	11
R108	2367.97		2367.97		0.000	587.4	1	0	30	1823.8	544.17	30	932.1	10
R109	1662.47		1662.47		0.000	362.3	1	0	40	1493.7	168.77	19	1146.9	13
R110	1750.23		1750.11		0.000	1933.2	5	0	60	1614.2	136.03	22	1068.0	12
R111		1756.12	1754.45	1755.99	0.007	3600.0	20	0	59				1048.7	12
R112			1877.49	1877.49		3600.0	1	0	32				948.6	10
C101	1208.17		1208.17		0.000	65.6	1	0	80	979.0	229.18	15	827.3	10
C102	2086.92		2086.92		0.000	1.1	1	0	0	2070.9	16.02	34	827.3	10
C103	3213.20		3213.20		0.000	0.9	1	0	0	3087.2	126.00	53	826.3	10
C104	4459.76		4459.76		0.000	0.7	1	0	0	4333.1	126.67	76	822.9	10
C105	1307.82		1307.82		0.000	2.8	1	0	0	1307.7	0.12	20	827.3	10
C106	1306.15		1306.15		0.000	17.0	1	0	18	1237.9	68.25	17	827.3	10
C107	1307.82		1307.82		0.000	4.6	1	0	0	1307.7	0.12	20	827.3	10
C108	1543.80		1543.80		0.000	6.8	1	0	0	1406.2	137.60	22	827.3	10
C109	1846.18		1846.18		0.000	12.2	1	0	0	1846.1	0.08	30	827.3	10
RC101	1886.69		1886.69		0.000	11.6	1	37	6	1761.5	125.19	19	1619.8	15
RC102	2073.01		2073.01		0.000	180.8	1	1	46	1844.7	228.31	21	1457.4	14
RC103	2325.49		2324.83		0.000	1526.1	5	0	49	1829.8	495.69	22	1258.0	11
RC104			2694.12	2694.12		3600.0	1	0	65				1132.3	10
RC105	1980.09		1980.09		0.000	65.8	1	21	10	1870.2	109.89	21	1513.7	15
RC106	1937.75		1935.23		0.000	536.7	5	0	46	1772.4	165.35	19	1372.7	12
RC107	2051.50		2050.69		0.000	2439.0	3	0	64	1851.9	199.61	21	1207.8	12
RC108	2313.83		2313.83		0.000	3213.1	1	0	24	1969.3	344.53	23	1114.2	11
R201	1404.47		1404.47		0.000	142.0	1	0	0	1376.3	28.17	13	1143.2	8
R202			1478.26	1478.26		3600.0	1	0	40				1029.6	8
R203			1762.03	1762.03		3600.0	1	0	52				870.8	6
R204	2183.10		2183.10		0.000	933.3	1	0	10	1705.5	477.60	28	731.3	5
R205	1401.57		1401.57		0.000	2226.7	1	0	0	1364.7	36.87	12	949.8	5
R206						3600.0	0	0	0				875.9	5
R207			1761.76	1761.76		3600.0	1	0	44				794.0	4
R208	2183.86		2183.86		0.000	839.7	1	0	4	1705.5	478.37	28	701.0	4
R209						3600.0	0	0	0				854.8	5
R210						3600.0	0	0	0				900.5	6
R211						3600.0	0	0	0				746.7	4
C201						3600.0	0	0	0				589.1	3
C202	1740.95		1740.95		0.000	65.9	1	0	0	1738.1	2.85	27	589.1	3
C203	3105.18		3105.18		0.000	2.1	1	0	0	3102.9	2.28	51	588.7	3
C204	4425.98		4425.98		0.000	1.5	1	0	0	4423.7	2.28	75	588.1	3
C205	752.24		752.24		0.000	227.8	1	0	0	752.0	0.24	9	586.4	3
C206						3600.0	0	0	0				586.4	3
C207			764.95	764.95		3600.0	1	0	10				585.8	3
C208			1007.82	1007.82		3600.0	1	0	10				585.8	3
RC201	1504.80		1504.80		0.000	175.5	1	0	0	1481.3	23.50	12	1261.8	9
RC202	1671.49		1671.49		0.000	674.1	1	0	4	1495.0	176.49	13	1092.3	8
RC203			2045.98	2045.98		3600.0	1	0	20				923.7	5
RC204			2494.56	2494.56		3600.0	1	0	14				783.5	4
RC205	1560.56		1560.56		0.000	1606.2	1	0	0	1519.6	40.96	13	1154	7
RC206						3600.0	0	0	0				1051.1	7
RC207						3600.0	0	0	0				962.9	6
RC208						3600.0	0	0	0				776.1	4

Table 14: Detailed results for the VRPTW-CNC with $\rho = 0.05$ for the $n = 100$-customer instances.

	VRPTW-CNC with Unlimited Fleet													
	Bounds							Cuts		Costs and Fleet			VRPTW	
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	%*Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	Fleet	c^*	K^*
R101	628.09		628.09		0.000	0.0	1	0	0	617.1	10.99	8	617.1	8
R102	688.14		688.14		0.000	0.0	1	0	0	632.4	55.74	9	547.1	7
R103	760.36		760.36		0.000	0.0	1	0	1	639.1	121.26	10	454.6	5
R104	805.51		805.51		0.000	0.1	1	0	1	671.1	134.41	11	416.9	4
R105	629.85		629.85		0.000	0.0	1	0	2	590.0	39.85	8	530.5	6
R106	661.65		661.65		0.000	0.0	1	0	0	581.8	79.85	8	465.4	5
R107	754.59		754.59		0.000	0.1	1	0	1	634.6	119.99	10	424.3	4
R108	801.91		801.91		0.000	0.1	1	0	1	671.1	130.81	11	397.3	4
R109	657.06		657.06		0.000	0.0	1	0	0	604.3	52.76	8	441.3	5
R110	627.48		627.48		0.000	0.0	1	0	0	574.3	53.18	8	444.1	5
R111	684.64		684.64		0.000	0.0	1	0	0	636.3	48.34	9	428.8	4
R112	705.45		705.45		0.000	0.2	1	0	0	621.5	83.95	9	393.0	4
C101	298.87		298.87		0.000	0.1	1	0	6	298.8	0.07	6	191.3	3
C102	439.21		439.21		0.000	0.0	1	0	0	439.2	0.01	9	190.3	3
C103	678.61		678.61		0.000	0.0	1	0	0	678.6	0.01	14	190.3	3
C104	838.50		838.50		0.000	0.0	1	0	0	838.5	0.00	18	186.9	3
C105	298.87		298.87		0.000	0.1	1	0	0	298.8	0.07	6	191.3	3
C106	298.87		298.87		0.000	0.1	1	0	3	298.8	0.07	6	191.3	3
C107	298.87		298.87		0.000	0.1	1	0	0	298.8	0.07	6	191.3	3
C108	342.87		342.87		0.000	0.1	1	0	0	328.2	14.67	7	191.3	3
C109	420.46		420.46		0.000	0.1	1	0	0	420.4	0.06	9	191.3	3
RC101	563.52		563.52		0.000	0.2	1	14	20	503.6	59.92	5	461.1	4
RC102	631.78		631.78		0.000	0.3	1	0	30	553.1	78.68	6	351.8	3
RC103	703.64		703.64		0.000	0.2	1	0	7	549.6	154.04	6	332.8	3
RC104	760.93		760.93		0.000	0.3	1	0	7	526.5	234.43	6	306.6	3
RC105	593.28		593.28		0.000	0.1	1	0	10	552.7	40.58	6	411.3	4
RC106	570.80		570.80		0.000	0.8	1	0	40	557.5	13.30	6	345.5	3
RC107	559.27		559.27		0.000	1.7	1	0	36	540.8	18.47	6	298.3	3
RC108	657.44		657.44		0.000	0.3	1	0	0	537.4	120.04	6	294.5	3
R201	521.42		521.42		0.000	0.0	1	0	0	509.7	11.72	4	463.3	4
R202	564.88		564.88		0.000	0.1	1	0	0	544.3	20.58	6	410.5	4
R203	669.33		669.33		0.000	0.1	1	0	1	586.6	82.73	8	391.4	3
R204	746.37		746.37		0.000	0.1	1	0	3	627.4	118.97	10	355.0	2
R205	521.42		521.42		0.000	0.1	1	0	0	509.7	11.72	4	393.0	3
R206	565.14		565.14		0.000	0.1	1	0	0	544.3	20.84	6	374.4	3
R207	669.34		669.34		0.000	0.1	1	0	1	586.6	82.74	8	361.6	3
R208	746.37		746.37		0.000	0.1	1	0	3	627.4	118.97	10	328.2	1
R209	526.47		526.47		0.000	0.2	1	0	0	517.2	9.27	4	370.7	2
R210	535.36		535.36		0.000	0.9	1	0	0	526.9	8.46	4	404.6	3
R211	559.20		559.20		0.000	1.6	1	0	0	541.9	17.30	5	350.9	2
C201	234.09		234.09		0.000	0.3	1	0	0	234.0	0.09	3	214.7	2
C202	462.50		462.50		0.000	0.3	1	0	0	462.5	0.00	8	214.7	2
C203	728.70		728.70		0.000	0.1	1	0	0	725.5	3.20	13	214.7	2
C204	879.50		879.50		0.000	0.0	1	0	0	876.3	3.20	17	213.1	1
C205	255.89		255.89		0.000	0.3	1	0	0	255.8	0.09	4	214.7	2
C206	254.75		254.75		0.000	0.4	1	0	0	244.9	9.85	3	214.7	2
C207	262.97		262.97		0.000	6.0	1	0	0	262.9	0.07	3	214.5	2
C208	325.47		325.47		0.000	0.4	1	0	0	325.4	0.07	6	214.5	2
RC201	395.10		395.10		0.000	0.1	1	0	0	395.1	0.00	3	360.2	3
RC202	503.41		503.41		0.000	0.2	1	0	0	455.2	48.21	4	338.0	3
RC203	626.65		626.65		0.000	0.4	1	0	10	489.5	137.15	5	326.9	3
RC204	705.90		705.90		0.000	0.4	1	0	0	531.8	174.10	6	299.7	3
RC205	390.50		390.50		0.000	0.2	1	0	0	385.4	5.10	3	338.0	3
RC206	395.10		395.10		0.000	0.1	1	0	0	395.1	0.00	3	324.0	3
RC207	432.93		432.93		0.000	0.5	1	0	0	385.0	47.92	3	298.3	3
RC208	438.23		438.23		0.000	8.9	1	0	10	380.1	58.12	3	269.1	2

Table 15: Detailed results for the VRPTW-CNC with $\rho = 0.1$ for the $n = 25$-customer instances.

	VRPTW-CNC with Unlimited Fleet													
	Bounds							Cuts		Costs and Fleet			VRPTW	
Instance	Opt	UB	LB_{root}	LB_{tree}	$\%Gap$	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	Fleet	c^*	K^*
R101	1062.50		1062.50		0.000	0.0	1	0	0	1046.1	16.40	12	1044.0	12
R102	1199.13		1199.13		0.000	0.3	1	0	2	1120.1	79.03	15	909.0	11
R103	1359.46		1359.46		0.000	0.4	1	0	0	1156.2	203.26	18	772.9	9
R104	1617.88		1617.88		0.000	1.0	1	0	4	1351.2	266.68	23	625.4	6
R105	1064.07		1064.07		0.000	0.1	1	0	0	990.2	73.87	11	899.3	9
R106	1169.66		1169.66		0.000	0.2	1	0	0	1068.4	101.26	14	793.0	8
R107	1318.12		1318.12		0.000	0.4	1	0	0	1112.4	205.72	17	711.1	7
R108	1609.55		1609.55		0.000	0.8	1	0	2	1290.1	319.45	22	617.7	6
R109	1108.71		1108.71		0.000	0.4	1	0	0	1025.5	83.21	13	786.8	8
R110	1145.83		1145.83		0.000	1.6	1	0	5	1049.1	96.73	14	697.0	7
R111	1182.13		1182.13		0.000	2.4	1	0	7	1101.9	80.23	15	707.2	7
R112	1281.90		1281.90		0.000	11.3	1	0	8	1095.4	186.50	15	630.2	6
C101	563.56		563.56		0.000	0.1	1	0	0	563.4	0.16	10	362.4	5
C102	933.81		933.81		0.000	0.1	1	0	0	933.8	0.01	18	361.4	5
C103	1405.02		1405.02		0.000	0.1	1	0	0	1403.5	1.52	28	361.4	5
C104	2070.31		2070.31		0.000	0.1	1	0	0	1992.3	78.01	41	358.0	5
C105	563.56		563.56		0.000	0.2	1	0	0	563.4	0.16	10	362.4	5
C106	565.36		565.36		0.000	0.3	1	0	3	565.2	0.16	10	362.4	5
C107	563.56		563.56		0.000	0.3	1	0	0	563.4	0.16	10	362.4	5
C108	702.66		702.66		0.000	0.3	1	0	0	687.9	14.76	13	362.4	5
C109	781.95		781.95		0.000	0.7	1	0	0	781.8	0.15	15	362.4	5
RC101	1091.43		1091.43		0.000	0.3	1	12	0	1004.9	86.53	10	944.0	8
RC102	1253.22		1251.03		0.000	7.1	5	0	34	1065.3	187.92	11	822.5	7
RC103	1450.19		1443.39		0.000	175.2	43	0	36	1149.6	300.60	12	710.9	6
RC104	1756.17		1748.87		0.000	192.8	13	0	32	1323.4	432.77	15	545.8	5
RC105	1111.80		1111.80		0.000	0.5	1	19	0	1013.8	98.00	10	855.3	8
RC106	1128.44		1128.44		0.000	0.6	1	0	0	1010.4	118.04	10	723.2	6
RC107	1137.03		1137.03		0.000	1.7	1	0	0	1005.1	131.93	10	642.7	6
RC108	1428.93		1419.00		0.000	889.1	65	0	46	1144.5	284.43	12	598.1	6
R201	920.37		920.37		0.000	0.7	1	0	0	896.5	23.87	8	791.9	6
R202	989.57		989.57		0.000	1.5	1	0	0	923.9	65.67	10	698.5	5
R203	1188.99		1188.99		0.000	1.7	1	0	10	1030.6	158.39	15	605.3	5
R204	1507.39		1507.39		0.000	2.4	1	0	9	1233.0	274.39	21	506.4	2
R205	926.95		926.95		0.000	2.6	1	0	0	896.1	30.85	8	690.1	4
R206	998.07		998.07		0.000	7.5	1	0	0	945.2	52.87	10	632.4	4
R207	1189.13		1189.13		0.000	7.2	1	0	10	1030.6	158.53	15	575.5	3
R208	1507.59		1507.59		0.000	2.4	1	0	9	1233.0	274.59	21	487.7	2
R209	901.23		901.23		0.000	17.2	1	0	0	891.3	9.93	8	600.6	4
R210	958.19		958.19		0.000	45.2	1	0	0	913.6	44.59	8	645.6	4
R211	994.66		994.66		0.000	64.9	1	0	0	973.3	21.36	11	535.5	3
C201						3600.0	0	0	0				360.2	3
C202	835.79		835.79		0.000	2.2	1	0	0	832.4	3.39	13	360.2	3
C203	1407.85		1407.85		0.000	0.2	1	0	0	1405.6	2.25	25	359.8	3
C204	2142.54		2142.54		0.000	0.1	1	0	0	2140.3	2.24	40	588.7	2
C205	395.41		395.41		0.000	14.2	1	0	0	395.2	0.21	4	359.8	3
C206	405.17		405.17		0.000	680.6	1	0	0	395.2	9.97	4	359.8	3
C207	435.56		435.56		0.000	98.8	1	0	0	431.4	4.16	4	359.6	3
C208	524.28		524.28		0.000	45.6	1	0	0	524.1	0.18	8	350.5	2
RC201	784.79		784.79		0.000	0.9	1	0	0	754.9	29.89	5	684.8	5
RC202	959.05		959.05		0.000	3.9	1	0	0	850.5	108.55	7	613.6	5
RC203	1290.36		1285.20		0.000	401.8	13	0	53	1013.9	276.46	10	555.3	4
RC204	1681.55		1681.55		0.000	82.1	1	0	24	1265.8	415.75	14	444.2	3
RC205	815.37		815.37		0.000	4.0	1	0	0	779.6	35.77	5	630.2	5
RC206	821.14		821.14		0.000	2.4	1	0	0	753.4	67.74	5	610.0	5
RC207	865.02		865.02		0.000	18.2	1	0	0	759.3	105.72	5	558.6	4
RC208	904.60		904.60		0.000	2684.8	1	0	28	838.1	66.50	6	476.7	3

Table 16: Detailed results for the VRPTW-CNC with $\rho = 0.1$ for the $n = 50$-customer instances.

	VRPTW-CNC with Unlimited Fleet												VRPTW	
	Bounds							Cuts		Costs and Fleet				
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	Fleet	c^*	K^*
R101	1691.57		1689.95		0.000	10.8	19	1	12	1641.0	50.57	21	1637.7	20
R102	1954.64		1954.63		0.000	13.3	3	0	2	1797.2	157.44	27	1466.6	18
R103	2288.70		2288.70		0.000	48.1	1	0	5	1963.3	325.40	33	1208.7	14
R104	2770.25		2770.25		0.000	72.2	1	0	13	2203.6	566.65	39	971.5	11
R105	1681.90		1681.90		0.000	6.1	1	0	0	1558.8	123.10	19	1355.3	15
R106	1935.67		1933.61		0.000	132.2	7	0	21	1725.9	209.77	25	1234.6	13
R107	2269.23		2269.23		0.000	113.8	1	0	19	1910.4	358.83	32	1064.6	11
R108	2759.32		2759.32		0.000	76.8	1	0	18	2196.3	563.02	39	932.1	10
R109	1803.87		1802.71		0.000	649.6	19	0	52	1591.4	212.47	21	1146.9	13
R110	1864.10		1864.10		0.000	268.9	1	0	16	1687.5	176.60	24	1068.0	12
R111	1875.01		1872.89		0.000	805.9	9	0	26	1735.1	139.91	25	1048.7	12
R112			2061.21	2062.47		3600.0	12	0	10				948.6	10
C101	1306.66		1306.66		0.000	12.1	1	0	30	1243.7	62.96	19	827.3	10
C102	2102.93		2102.93		0.000	2.4	1	0	0	2070.9	32.03	34	827.3	10
C103	3314.29		3314.29		0.000	1.3	1	0	0	3165.8	148.49	54	826.3	10
C104	4575.63		4575.63		0.000	0.7	1	0	0	4419.1	156.53	77	822.9	10
C105	1307.94		1307.94		0.000	4.1	1	0	0	1307.7	0.24	20	827.3	10
C106	1336.27		1336.27		0.000	9.1	1	0	0	1295.6	40.67	19	827.3	10
C107	1307.94		1307.94		0.000	5.3	1	0	0	1307.7	0.24	20	827.3	10
C108	1624.08		1624.08		0.000	8.0	1	0	7	1509.3	114.78	24	827.3	10
C109	1846.26		1846.26		0.000	21.7	1	0	0	1846.1	0.16	30	827.3	10
RC101	1990.66		1990.66		0.000	8.4	1	2	19	1849.4	141.26	21	1619.8	15
RC102	2280.18		2276.29		0.000	141.8	11	0	24	1889.7	390.48	23	1457.4	14
RC103		2665.44	2657.01	2662.37	0.115	3600.0	47	0	78				1258.0	11
RC104		3170.56	3162.33	3165.80	0.150	3600.0	44	0	38				1132.3	10
RC105	2085.14		2085.14		0.000	49.8	1	3	14	1905.1	180.04	21	1513.7	15
RC106	2050.06		2050.06		0.000	49.4	1	0	5	1885.9	164.16	21	1372.7	12
RC107		2189.58	2185.43	2188.49	0.050	3600.0	36	0	30				1207.8	12
RC108			2568.67	2571.78		3600.0	13	0	34				1114.2	11
R201	1427.78		1427.78		0.000	117.0	1	0	0	1384.9	42.88	13	1143.2	8
R202	1556.19		1556.19		0.000	705.9	1	0	10	1431.9	124.29	15	1029.6	8
R203	1961.00		1961.00		0.000	191.3	1	0	10	1623.1	337.90	25	870.8	6
R204	2563.71		2563.70		0.000	261.8	3	0	16	2030.8	532.91	35	731.3	5
R205	1436.46		1436.46		0.000	2187.0	1	0	0	1373.7	62.76	12	949.8	5
R206	1563.84		1563.84		0.000	1914.9	1	0	8	1451.6	112.24	16	875.9	5
R207	1962.00		1962.00		0.000	352.4	1	0	10	1623.1	338.90	25	794.0	4
R208	2564.51		2564.51		0.000	257.1	1	0	16	2030.8	533.71	35	701.0	4
R209						3600.0	0	0	0				854.8	5
R210						3600.0	0	0	0				900.5	6
R211						3600.0	0	0	0				746.7	4
C201						3600.0	0	0	0				589.1	3
C202	1743.81		1743.81		0.000	26.9	1	0	0	1738.1	5.71	27	589.1	3
C203	3107.46		3107.46		0.000	1.6	1	0	0	3102.9	4.56	51	588.7	3
C204	4428.26		4428.26		0.000	1.7	1	0	0	4423.7	4.56	75	588.1	3
C205	752.48		752.48		0.000	1096.0	1	0	0	752.0	0.48	9	586.4	3
C206						3600.0	0	0	0				586.4	3
C207	772.76		772.76		0.000	1829.8	1	0	0	763.4	9.36	6	585.8	3
C208	1008.04		1008.04		0.000	554.6	1	0	0	1007.6	0.44	15	585.8	3
RC201	1522.49		1522.49		0.000	132.8	1	0	10	1488.3	34.19	12	1261.8	9
RC202	1797.82		1797.82		0.000	1634.4	1	0	52	1583.1	214.72	15	1092.3	8
RC203			2334.46	2335.39		3600.0	4	0	50				923.7	5
RC204	2953.52		2953.52		0.000	1492.9	1	0	40	2252.4	701.12	31	783.5	4
RC205	1594.78		1594.78		0.000	2245.8	1	0	30	1548.0	46.78	13	1154.0	7
RC206	1595.76		1595.76		0.000	2098.4	1	0	20	1515.5	80.26	13	1051.1	7
RC207						3600.0	0	0	0				962.9	6
RC208						3600.0	0	0	0				776.1	4

Table 17: Detailed results for the VRPTW-CNC with $\rho = 0.1$ for the $n = 100$-customer instances.

	VRPTW-CNC with Limited Fleet												
	Bounds							Cuts		Costs		VRPTW	
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	c^*	K^*
R101	617.21		617.21		0.000	0.0	1	0	0	617.1	0.11	617.1	8
R102	570.20		570.20		0.000	0.0	1	5	0	550.9	19.30	547.1	7
R103	493.21		493.21		0.000	0.1	1	0	0	454.6	38.61	454.6	5
R104	455.12		455.12		0.000	0.2	1	0	0	416.9	38.22	416.9	4
R105	532.06		532.06		0.000	0.0	1	0	0	530.7	1.36	530.5	6
R106	489.29		489.29		0.000	0.1	1	6	0	473.3	15.99	465.4	5
R107	467.27		467.27		0.000	0.3	1	0	0	429.8	37.47	424.3	4
R108	434.51		434.51		0.000	0.4	1	0	0	399.2	35.31	397.3	4
R109	448.38		448.38		0.000	0.1	1	0	0	441.3	7.08	441.3	5
R110	454.59		454.59		0.000	0.2	1	0	10	444.1	10.48	444.1	5
R111	460.98		460.98		0.000	0.2	1	0	0	428.8	32.18	428.8	4
R112	415.46		415.46		0.000	0.3	1	0	0	393.0	22.46	393.0	4
C101	193.51		193.51		0.000	0.1	1	0	0	191.3	2.21	191.3	3
C102	414.13		414.13		0.000	50.2	1	0	20	235.6	178.53	190.3	3
C103	781.75		781.75		0.000	603.4	1	0	30	259.3	522.45	190.3	3
C104			1036.67	1036.67		3600.0	1	0	20			186.9	3
C105	200.09		200.09		0.000	0.1	1	0	0	191.3	8.79	191.3	3
C106	193.52		193.52		0.000	0.1	1	0	0	191.3	2.22	191.3	3
C107	211.21		211.21		0.000	0.2	1	0	0	191.3	19.91	191.3	3
C108	226.63		226.63		0.000	0.9	1	0	0	191.3	35.33	191.3	3
C109	272.83		272.83		0.000	7.5	1	0	0	191.3	81.53	191.3	3
RC101	463.52		463.52		0.000	0.1	1	14	0	461.1	2.42	461.1	4
RC102	364.32		364.32		0.000	0.6	1	0	0	352.0	12.32	351.8	3
RC103	356.87		356.87		0.000	5.1	1	0	0	336.0	20.87	332.8	3
RC104	347.48		347.48		0.000	8.4	1	0	0	319.0	28.48	306.6	3
RC105	424.48		424.48		0.000	0.1	1	0	0	412.5	11.98	411.3	4
RC106	352.58		352.58		0.000	0.8	1	0	0	345.5	7.08	345.5	3
RC107	315.68		315.68		0.000	3.0	1	0	0	298.8	16.88	298.3	3
RC108	317.20		317.20		0.000	5.1	1	0	0	296.5	20.70	294.5	3
R201	483.87		483.87		0.000	0.2	1	0	6	470.2	13.67	463.3	4
R202	487.80		487.80		0.000	0.4	1	0	0	462.9	24.90	410.5	4
R203	479.35		479.35		0.000	3.5	1	0	0	454.7	24.65	391.4	3
R204	488.01		488.01		0.000	3523.3	1	0	10	410.4	77.61	355.0	2
R205	465.55		465.55		0.000	0.3	1	0	0	420.7	44.85	393.0	3
R206	474.20		474.20		0.000	1.3	1	0	0	431.7	42.50	374.4	3
R207	474.56		474.56		0.000	55.9	1	0	20	443.2	31.36	361.6	3
R208						3600.0	0	0	0			328.2	1
R209	480.09		480.09		0.000	24.8	1	0	10	407.2	72.89	370.7	2
R210	480.57		480.57		0.000	2.1	1	0	20	454.5	26.07	404.6	3
R211	475.16		475.16		0.000	574.3	1	0	10	407.2	67.96	350.9	2
C201	219.03		219.03		0.000	0.8	1	0	0	214.7	4.33	214.7	2
C202	365.73		365.73		0.000	118.4	1	0	0	251.4	114.33	214.7	2
C203	788.77		788.77		0.000	2115.4	1	0	0	287.0	501.77	214.7	2
C204						3600.0	0	0	0			213.1	1
C205	227.64		227.64		0.000	22.0	1	0	0	214.7	12.94	214.7	2
C206	231.49		231.49		0.000	4.3	1	0	0	223.8	7.69	214.7	2
C207	244.03		244.03		0.000	17.6	1	0	0	235.5	8.53	214.5	2
C208	249.45		249.45		0.000	10.5	1	0	10	214.7	34.75	214.5	2
RC201	378.76		378.76		0.000	0.2	1	0	0	363.1	15.66	360.2	3
RC202	367.29		367.29		0.000	1.3	1	0	0	352.7	14.59	338.0	3
RC203	357.86		357.86		0.000	1.9	1	0	0	343.7	14.16	326.9	3
RC204	342.73		342.73		0.000	39.0	1	0	0	326.5	16.23	299.7	3
RC205	371.64		371.64		0.000	0.8	1	0	0	359.9	11.74	338.0	3
RC206	376.45		376.45		0.000	1.2	1	0	0	363.1	13.35	324.0	3
RC207	367.81		367.81		0.000	5.6	1	0	0	354.3	13.51	298.3	3
RC208	407.60		407.60		0.000	1987.2	1	0	0	349.8	57.80	269.1	2

Table 18: Detailed results for the VRPTW-CNC with $\rho = 0.001$ for the $n = 25$-customer instances.

	VRPTW-CNC with Limited Fleet												
	Bounds							Cuts		Costs		VRPTW	
Instance	Opt	UB	LB_{root}	LB_{tree}	$\%Gap$	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	c^*	K^*
R101	1044.19		1044.19		0.000	0.1	1	3	0	1044.0	0.19	1044.0	12
R102	963.32		963.32		0.000	0.8	1	7	10	928.6	34.72	909.0	11
R103	839.55		839.55		0.000	22.4	1	0	0	789.8	49.75	772.9	9
R104			748.52	748.52		3600.0	1	10	30			625.4	6
R105	903.19		903.19		0.000	1.5	1	5	30	899.3	3.89	899.3	9
R106	854.06		854.06		0.000	9.6	1	8	30	794.4	59.66	793.0	8
R107	792.01		792.01		0.000	19.2	1	0	0	717.4	74.61	711.1	7
R108			728.59	728.59		3600.0	1	10	40			617.7	6
R109	803.26		799.99		0.000	58.8	21	5	61	786.9	16.36	786.8	8
R110	737.99		737.99		0.000	34.1	1	4	0	698.0	39.99	697.0	7
R111	768.85		768.85		0.000	146.6	1	10	18	711.4	57.45	707.2	7
R112			697.87	697.87		3600.0	1	3	70			630.2	6
C101	367.50		367.50		0.000	0.5	1	0	0	362.4	5.10	362.4	5
C102			1194.16	1194.16		3600.0	1	0	10			361.4	5
C103						3600.0	0	0	0			361.4	5
C104						3600.0	0	0	0			358.0	5
C105	382.73		382.73		0.000	1.4	1	0	0	362.4	20.33	362.4	5
C106	379.66		379.66		0.000	0.6	1	0	0	365.0	14.66	362.4	5
C107	402.56		402.56		0.000	8.6	1	0	0	362.4	40.16	362.4	5
C108	447.79		447.79		0.000	141.7	1	0	0	365.0	82.79	362.4	5
C109	530.77		530.77		0.000	1428.8	1	0	0	362.4	168.37	362.4	5
RC101	947.85		947.85		0.000	0.8	1	65	0	944.0	3.85	944.0	8
RC102	857.08		857.08		0.000	50.8	1	25	70	826.7	30.38	822.5	7
RC103	780.38		780.38		0.000	394.8	1	8	30	716.3	64.07	710.9	6
RC104	635.71		635.71		0.000	962.7	1	0	0	559.4	76.31	545.8	5
RC105	879.33		879.33		0.000	4.0	1	31	10	857.4	21.93	855.3	8
RC106	741.74		741.74		0.000	8.3	1	7	20	723.4	18.34	723.2	6
RC107	689.71		684.41		0.000	1354.4	31	5	10	645.8	43.91	642.7	6
RC108	658.95		658.95		0.000	897.5	1	3	20	599.4	59.55	598.1	6
R201	825.19		825.19		0.000	1.5	1	0	0	793.7	31.48	791.9	6
R202	822.11		821.31		0.000	714.2	5	0	76	778.4	43.71	698.5	5
R203						3600.0	0	0	0			605.3	5
R204						3600.0	0	0	0			506.4	2
R205	820.72		820.38		0.000	1379.6	3	0	100	751.7	69.02	690.1	4
R206	795.14		795.14		0.000	1832.6	1	0	70	739.1	56.04	632.4	4
R207						3600.0	0	0	0			575.5	3
R208						3600.0	0	0	0			487.7	2
R209			788.98	788.98		3600.0	1	0	40			600.6	4
R210	808.12		808.12		0.000	615.6	1	0	0	752.3	55.82	645.6	4
R211						3600.0	0	0	0			535.5	3
C201	366.44		366.44		0.000	13.0	1	0	0	360.2	6.24	360.2	3
C202						3600.0	0	0	0			360.2	3
C203						3600.0	0	0	0			359.8	3
C204						3600.0	0	0	0			588.7	2
C205						3600.0	0	0	0			359.8	3
C206						3600.0	0	0	0			359.8	3
C207						3600.0	0	0	0			359.6	3
C208						3600.0	0	0	0			350.5	2
RC201	720.25		720.25		0.000	1.4	1	0	0	695.0	25.25	684.8	5
RC202	708.65		708.65		0.000	28.1	1	0	0	682.2	26.45	613.6	5
RC203						3600.0	0	0	0			555.3	4
RC204						3600.0	0	0	0			444.2	3
RC205	707.44		707.44		0.000	7.3	1	0	0	665.7	41.74	630.2	5
RC206	700.57		700.57		0.000	3.9	1	0	0	652.6	47.97	610.0	5
RC207						3600.0	0	0	0			558.6	4
RC208						3600.0	0	0	0			476.7	3

Table 19: Detailed results for the VRPTW-CNC with $\rho = 0.001$ for the $n = 50$-customer instances.

	VRPTW-CNC with Limited Fleet											VRPTW	
	Bounds							Cuts		Costs			
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	c^*	K^*
R101	1638.32		1634.59		0.000	7.0	13	5	4	1637.7	0.62	1637.7	20
R102	1522.14		1522.14		0.000	35.2	1	0	0	1484.2	37.94	1466.6	18
R103						3600.0	0	0	0			1208.7	14
R104						3600.0	0	0	0			971.5	11
R105	1363.92		1363.92		0.000	156.3	1	9	72	1355.3	8.62	1355.3	15
R106			1323.12	1323.12		3600.0	1	4	41			1234.6	13
R107						3600.0	0	0	0			1064.6	11
R108						3600.0	0	0	0			932.1	10
R109			1179.42	1179.42		3600.0	1	14	80			1146.9	13
R110			1148.17	1148.17		3600.0	1	0	20			1068.0	12
R111						3600.0	0	0	0			1048.7	12
R112						3600.0	0	0	0			948.6	10
C101	837.74		837.74		0.000	5.1	1	0	0	827.3	10.44	827.3	10
C102						3600.0	0	0	0			827.3	10
C103						3600.0	0	0	0			826.3	10
C104						3600.0	0	0	0			822.9	10
C105	868.90		868.90		0.000	19.9	1	0	0	827.3	41.60	827.3	10
C106	913.00		913.00		0.000	344.3	1	0	10	833.1	79.90	827.3	10
C107	908.63		908.63		0.000	171.7	1	0	0	827.3	81.33	827.3	10
C108			996.54	996.54		3600.0	1	0	10			827.3	10
C109						3600.0	0	0	0			827.3	10
RC101	1629.64		1629.64		0.000	24.5	1	78	19	1619.8	9.84	1619.8	15
RC102			1531.45	1531.45		3600.0	2	29	100			1457.4	14
RC103			1376.18	1376.18		3600.0	1	15	0			1258.0	11
RC104						3600.0	0	0	0			1132.3	10
RC105	1564.05		1563.93		0.000	2029.0	3	28	45	1517.8	46.25	1513.7	15
RC106			1387.02	1387.02		3600.0	1	14	20			1372.7	12
RC107						3600.0	0	0	0			1207.8	12
RC108						3600.0	0	0	0			1114.2	11
R201						3600.0	0	0	0			1143.2	8
R202						3600.0	0	0	0			1029.6	8
R203						3600.0	0	0	0			870.8	6
R204						3600.0	0	0	0			731.3	5
R205						3600.0	0	0	0			949.8	5
R206						3600.0	0	0	0			875.9	5
R207						3600.0	0	0	0			794.0	4
R208						3600.0	0	0	0			701.0	4
R209						3600.0	0	0	0			854.8	5
R210						3600.0	0	0	0			900.5	6
R211						3600.0	0	0	0			746.7	4
C201	613.99		613.99		0.000	1021.2	1	0	0	589.1	24.89	589.1	3
C202						3600.0	0	0	0			589.1	3
C203						3600.0	0	0	0			588.7	3
C204						3600.0	0	0	0			588.1	3
C205						3600.0	0	0	0			586.4	3
C206						3600.0	0	0	0			586.4	3
C207						3600.0	0	0	0			585.8	3
C208						3600.0	0	0	0			585.8	3
RC201						3600.0	0	0	0			1261.8	9
RC202						3600.0	0	0	0			1092.3	8
RC203						3600.0	0	0	0			923.7	5
RC204						3600.0	0	0	0			783.5	4
RC205						3600.0	0	0	0			1154.0	7
RC206						3600.0	0	0	0			1051.1	7
RC207						3600.0	0	0	0			962.9	6
RC208						3600.0	0	0	0			776.1	4

Table 20: Detailed results for the VRPTW-CNC with $\rho = 0.001$ for the $n = 100$-customer instances.

	VRPTW-CNC with Limited Fleet												VRPTW	
	Bounds							Cuts		Costs			
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	c^*	K^*
R101	618.20		618.20		0.000	0.0	1	0	0	617.1	1.10	617.1	8
R102	619.61		619.61		0.000	0.0	1	2	0	586.0	33.61	547.1	7
R103	671.74		671.74		0.000	0.9	1	3	0	513.9	157.84	454.6	5
R104	724.17		724.17		0.000	3.0	1	0	0	449.5	274.67	416.9	4
R105	544.30		544.30		0.000	0.0	1	0	0	530.7	13.60	530.5	6
R106	581.81		581.81		0.000	0.2	1	2	0	518.8	63.01	465.4	5
R107	700.60		700.60		0.000	3.2	1	0	5	498.0	202.60	424.3	4
R108	663.74		663.74		0.000	19.4	1	0	0	439.5	224.24	397.3	4
R109	505.31		505.31		0.000	0.1	1	0	0	449.3	56.01	441.3	5
R110	514.41		514.41		0.000	0.1	1	0	0	466.4	48.01	444.1	5
R111	667.12		667.12		0.000	2.8	1	0	13	454.9	212.22	428.8	4
R112	610.95		610.95		0.000	25.1	1	4	20	402.8	208.15	393.0	4
C101	213.38		213.38		0.000	0.1	1	0	0	191.3	22.08	191.3	3
C102	1851.89		1851.89		0.000	119.2	1	0	27	268.3	1583.59	190.3	3
C103	5465.59		5465.59		0.000	330.6	1	0	0	266.6	5198.99	190.3	3
C104			8127.70	8127.70		3600.0	1	0	20			186.9	3
C105	279.24		279.24		0.000	0.1	1	0	0	191.3	87.94	191.3	3
C106	213.51		213.51		0.000	0.2	1	0	0	191.3	22.21	191.3	3
C107	390.43		390.43		0.000	0.4	1	0	10	191.3	199.13	191.3	3
C108	483.75		483.75		0.000	2.5	1	0	19	220.3	263.45	191.3	3
C109	887.29		887.29		0.000	11.6	1	0	20	245.5	641.79	191.3	3
RC101	485.34		485.34		0.000	0.2	1	14	0	461.1	24.24	461.1	4
RC102	448.45		448.45		0.000	0.7	1	0	0	360.1	88.35	351.8	3
RC103	479.61		479.61		0.000	8.8	1	0	0	358.6	121.01	332.8	3
RC104	516.32		516.32		0.000	10.2	1	0	0	346.2	170.12	306.6	3
RC105	508.85		508.85		0.000	0.5	1	0	10	448.0	60.85	411.3	4
RC106	398.88		398.88		0.000	0.3	1	0	0	359.6	39.28	345.5	3
RC107	414.24		414.24		0.000	3.7	1	0	0	327.6	86.64	298.3	3
RC108	475.53		475.53		0.000	46.7	1	0	0	302.4	173.13	294.5	3
R201	503.97		503.97		0.000	0.2	1	0	0	501.5	2.47	463.3	4
R202	534.74		534.74		0.000	0.4	1	0	0	526.9	7.84	410.5	4
R203	656.14		656.14		0.000	37.8	1	0	0	497.0	159.14	391.4	3
R204						3600.0	0	0	0			355.0	2
R205	544.57		544.57		0.000	3.7	1	0	0	530.0	14.57	393.0	3
R206	564.52		564.52		0.000	9.4	1	0	0	532.9	31.62	374.4	3
R207	656.89		656.89		0.000	196.2	1	0	0	497.0	159.89	361.6	3
R208						3600.0	0	0	0			328.2	1
R209	659.52		659.52		0.000	309.8	1	0	20	521.2	138.32	370.7	2
R210	527.03		527.03		0.000	2.4	1	0	0	519.0	8.03	404.6	3
R211	670.51		670.51		0.000	1504.8	1	0	0	535.8	134.71	350.9	2
C201	258.02		258.02		0.000	1.4	1	0	0	214.7	43.32	214.7	2
C202	1211.52		1211.52		0.000	288.2	1	0	0	335.9	875.62	214.7	2
C203	5304.71		5304.71		0.000	3071.1	1	0	0	287.0	5017.71	214.7	2
C204						3600.0	0	0	0			213.1	1
C205	315.74		315.74		0.000	2.3	1	0	0	228.1	87.64	214.7	2
C206	251.66		251.66		0.000	1.7	1	0	0	235.8	15.86	214.7	2
C207	289.31		289.31		0.000	18.1	1	0	0	251.6	37.71	214.5	2
C208	490.12		490.12		0.000	2.6	1	0	0	226.7	263.42	214.5	2
RC201	395.10		395.10		0.000	0.3	1	0	0	395.1	0.00	360.2	3
RC202	431.12		431.12		0.000	4.2	1	0	0	379.7	51.42	338.0	3
RC203	446.72		446.72		0.000	3.4	1	0	0	368.3	78.42	326.9	3
RC204	447.33		447.33		0.000	12.9	1	0	0	351.1	96.22	299.7	3
RC205	385.91		385.91		0.000	0.9	1	0	0	385.4	0.51	338.0	3
RC206	395.10		395.10		0.000	0.8	1	0	0	395.1	0.00	324.0	3
RC207	386.81		386.81		0.000	4.5	1	0	0	378.6	8.20	298.3	3
RC208	625.84		625.84		0.000	1870.1	1	0	0	454.2	171.64	269.1	2

Table 21: Detailed results for the VRPTW-CNC with $\rho = 0.01$ for the $n = 25$-customer instances.

	VRPTW-CNC with Limited Fleet												
	Bounds							Cuts		Costs		VRPTW	
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	c^*	K^*
R101	1045.93		1045.93		0.000	0.1	1	3	0	1044.0	1.93	1044.0	12
R102	1035.52		1035.52		0.000	0.3	1	0	0	989.3	46.22	909.0	11
R103	1099.64		1099.64		0.000	163.0	1	0	20	860.9	238.74	772.9	9
R104						3600.0	0	0	0			625.4	6
R105	938.17		938.17		0.000	0.6	1	3	0	899.3	38.86	899.3	9
R106	1010.33		1010.33		0.000	15.3	1	6	5	913.8	96.53	793.0	8
R107	1175.98		1175.98		0.000	891.8	1	0	20	806.4	369.58	711.1	7
R108						3600.0	0	0	0			617.7	6
R109	927.77		927.77		0.000	31.6	1	5	60	814.1	113.67	786.8	8
R110		1004.94	999.20	1004.83	0.011	3600.0	13	4	100			697.0	7
R111	1090.45		1090.45		0.000	216.0	1	0	0	795.9	294.55	707.2	7
R112			1200.65	1200.65		3600.0	1	2	30			630.2	6
C101	413.36		413.36		0.000	0.4	1	0	0	362.4	50.96	362.4	5
C102			7959.11	7959.11		3600.0	1	0	10			361.4	5
C103						3600.0	0	0	0			361.4	5
C104						3600.0	0	0	0			358.0	5
C105	564.60		564.60		0.000	2.4	1	0	0	386.5	178.10	362.4	5
C106	511.61		511.61		0.000	0.9	1	0	0	365.0	146.61	362.4	5
C107	751.63		751.63		0.000	23.5	1	0	0	386.5	365.13	362.4	5
C108	1156.37		1156.37		0.000	533.3	1	0	0	398.3	758.07	362.4	5
C109						3600.0	0	0	0			362.4	5
RC101	979.86		979.86		0.000	1.1	1	62	0	945.1	34.76	944.0	8
RC102		1078.38	1038.07	1060.13	1.721	3600.0	235	19	100			822.5	7
RC103			1215.13	1215.13		3600.0	1	8	84			710.9	6
RC104						3600.0	0	0	0			545.8	5
RC105	1019.29		1015.61		0.000	139.6	11	27	100	920.5	98.79	855.3	8
RC106	874.23		874.23		0.000	22.9	1	7	21	742.5	131.73	723.2	6
RC107	988.46		984.71		0.000	3444.2	9	3	100	713.2	275.26	642.7	6
RC108			1038.89	1038.89		3600.0	1	2	30			598.1	6
R201	891.71		891.71		0.000	3.3	1	0	6	864.4	27.31	791.9	6
R202	907.23		907.23		0.000	250.8	1	0	10	873.0	34.23	698.5	5
R203						3600.0	0	0	0			605.3	5
R204						3600.0	0	0	0			506.4	2
R205	967.07		987.67		0.000	2532.8	1	0	0	929.2	37.87	690.1	4
R206						3600.0	0	0	0			632.4	4
R207						3600.0	0	0	0			575.5	3
R208						3600.0	0	0	0			487.7	2
R209						3600.0	0	0	0			600.6	4
R210						3600.0	0	0	0			645.6	4
R211						3600.0	0	0	0			535.5	3
C201						3600.0	0	0	0			360.2	3
C202						3600.0	0	0	0			360.2	3
C203						3600.0	0	0	0			359.8	3
C204						3600.0	0	0	0			588.7	2
C205	403.73		463.16		0.000	531.4	1	0	0	379.9	23.83	359.8	3
C206	393.43		466.27		0.000	1663.8	1	0	0	389.6	3.83	359.8	3
C207						3600.0	0	0	0			359.6	3
C208						3600.0	0	0	0			350.5	2
RC201	753.13		753.13		0.000	1.8	1	0	0	742.2	10.93	684.8	5
RC202	793.10		793.10		0.000	193.3	1	0	0	723.2	69.90	613.6	5
RC203						3600.0	0	0	0			555.3	4
RC204						3600.0	0	0	0			444.2	3
RC205	767.42		767.42		0.000	10.5	1	0	0	746.4	21.02	630.2	5
RC206	755.41		755.41		0.000	12.2	1	0	0	740.7	14.71	610.0	5
RC207						3600.0	0	0	0			558.6	4
RC208						3600.0	0	0	0			476.7	3

Table 22: Detailed results for the VRPTW-CNC with $\rho = 0.01$ for the $n = 50$-customer instances.

	VRPTW-CNC with Limited Fleet												
	Bounds							Cuts		Costs		VRPTW	
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	c^*	K^*
R101	1643.90		1639.88		0.000	6.9	13	5	4	1637.7	6.20	1637.7	20
R102	1702.24		1702.24		0.000	553.5	1	0	20	1564.5	137.74	1466.6	18
R103						3600.0	0	0	0			1208.7	14
R104						3600.0	0	0	0			971.5	11
R105	1435.42		1435.42		0.000	121.3	1	30	79	1365.9	69.52	1355.3	15
R106			1647.66	1647.66		3600.0	1	23	0			1234.6	13
R107						3600.0	0	0	0			1064.6	11
R108						3600.0	0	0	0			932.1	10
R109			1417.70	1417.70		3600.0	1	9	70			1146.9	13
R110						3600.0	0	0	0			1068.0	12
R111						3600.0	0	0	0			1048.7	12
R112						3600.0	0	0	0			948.6	10
C101	931.71		931.71		0.000	21.7	1	0	0	827.3	104.41	827.3	10
C102						3600.0	0	0	0			827.3	10
C103						3600.0	0	0	0			826.3	10
C104						3600.0	0	0	0			822.9	10
C105	1209.26		1209.26		0.000	86.2	1	0	0	848.0	361.26	827.3	10
C106	1470.48		1470.48		0.000	422.0	1	0	0	866.5	603.98	827.3	10
C107	1574.13		1574.13		0.000	241.4	1	0	0	848.7	725.43	827.3	10
C108						3600.0	0	0	0			827.3	10
C109						3600.0	0	0	0			827.3	10
RC101	1717.66		1717.66		0.000	385.4	1	85	58	1621.8	95.86	1619.8	15
RC102			1822.99	1822.99		3600.0	1	30	20			1457.4	14
RC103						3600.0	0	0	0			1258.0	11
RC104						3600.0	0	0	0			1132.3	10
RC105			1787.60	1787.60		3600.0	1	28	60			1513.7	15
RC106			1677.13	1677.13		3600.0	1	17	0			1372.7	12
RC107						3600.0	0	0	0			1207.8	12
RC108						3600.0	0	0	0			1114.2	11
R201						3600.0	0	0	0			1143.2	8
R202						3600.0	0	0	0			1029.6	8
R203						3600.0	0	0	0			870.8	6
R204						3600.0	0	0	0			731.3	5
R205						3600.0	0	0	0			949.8	5
R206						3600.0	0	0	0			875.9	5
R207						3600.0	0	0	0			794.0	4
R208						3600.0	0	0	0			701.0	4
R209						3600.0	0	0	0			854.8	5
R210						3600.0	0	0	0			900.5	6
R211						3600.0	0	0	0			746.7	4
C201	807.87		807.87		0.000	1183.1	1	0	0	621.2	186.67	589.1	3
C202						3600.0	0	0	0			589.1	3
C203						3600.0	0	0	0			588.7	3
C204						3600.0	0	0	0			588.1	3
C205						3600.0	0	0	0			586.4	3
C206						3600.0	0	0	0			586.4	3
C207						3600.0	0	0	0			585.8	3
C208						3600.0	0	0	0			585.8	3
RC201						3600.0	0	0	0			1261.8	9
RC202						3600.0	0	0	0			1092.3	8
RC203						3600.0	0	0	0			923.7	5
RC204						3600.0	0	0	0			783.5	4
RC205						3600.0	0	0	0			1154.0	7
RC206						3600.0	0	0	0			1051.1	7
RC207						3600.0	0	0	0			962.9	6
RC208						3600.0	0	0	0			776.1	4

Table 23: Detailed results for the VRPTW-CNC with $\rho = 0.01$ for the $n = 100$-customer instances.

	VRPTW-CNC with Limited Fleet												
	Bounds							Cuts		Costs		VRPTW	
Instance	Opt	UB	LB_{root}	LB_{tree}	$\%Gap$	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	c^*	K^*
R101	622.59		622.59		0.000	0.0	1	0	0	617.1	5.49	617.1	8
R102	700.22		700.22		0.000	0.1	1	2	0	605.3	94.92	547.1	7
R103	1257.89		1257.71		0.000	9.8	3	1	44	555.4	702.50	454.6	5
R104	1817.26		1817.26		0.000	4.6	1	0	0	461.0	1356.26	416.9	4
R105	593.84		593.84		0.000	0.0	1	0	0	538.0	55.84	530.5	6
R106	824.45		824.45		0.000	1.8	1	5	37	523.4	301.05	465.4	5
R107	1507.80		1507.80		0.000	5.4	1	0	10	501.8	1006.00	424.3	4
R108	1556.88		1556.88		0.000	19.6	1	0	10	443.1	1113.78	397.3	4
R109	626.40		626.40		0.000	0.1	1	0	0	507.2	119.20	441.3	5
R110	706.46		706.46		0.000	0.3	1	0	0	466.4	240.06	444.1	5
R111	1389.34		1389.34		0.000	16.0	1	0	39	491.1	898.24	428.8	4
R112	1357.64		1357.64		0.000	113.2	1	0	40	431.0	926.64	393.0	4
C101	301.70		301.70		0.000	0.1	1	0	0	191.3	110.40	191.3	3
C102	8162.11		8162.11		0.000	39.7	1	0	0	278.8	7883.31	190.3	3
C103	26261.60		26261.60		0.000	221.1	1	0	0	266.6	25995.00	190.3	3
C104			39629.10	39629.10		3600.0	1	0	20			186.9	3
C105	630.41		630.41		0.000	1.1	1	0	49	213.8	416.61	191.3	3
C106	302.34		302.34		0.000	0.1	1	0	0	191.3	111.04	191.3	3
C107	1137.37		1137.37		0.000	0.6	1	0	26	245.5	891.87	191.3	3
C108	1497.00		1497.00		0.000	2.9	1	0	20	252.7	1244.30	191.3	3
C109	3454.44		3454.44		0.000	11.7	1	0	0	245.5	3208.94	191.3	3
RC101	547.23		547.23		0.000	0.2	1	14	0	476.0	71.23	461.1	4
RC102	801.83		801.83		0.000	1.9	1	0	0	360.1	441.73	351.8	3
RC103	963.65		963.65		0.000	4.7	1	0	0	358.6	605.05	332.8	3
RC104	1194.08		1194.08		0.000	16.6	1	0	0	348.2	845.88	306.6	3
RC105	729.34		729.34		0.000	1.6	1	0	40	471.2	258.14	411.3	4
RC106	550.76		550.76		0.000	0.8	1	0	0	366.7	184.06	345.5	3
RC107	748.38		748.38		0.000	3.7	1	0	0	342.3	406.08	298.3	3
RC108	1120.81		1120.81		0.000	32.8	1	0	0	320.2	800.61	294.5	3
R201	513.34		513.34		0.000	0.2	1	0	0	502.5	10.84	463.3	4
R202	566.09		566.09		0.000	0.8	1	0	0	526.9	39.19	410.5	4
R203	1273.62		1273.62		0.000	181.5	1	0	0	502.9	770.72	391.4	3
R204						3600.0	0	0	0			355.0	2
R205	577.49		577.49		0.000	5.0	1	0	0	561.5	15.99	393.0	3
R206	663.25		663.25		0.000	26.3	1	0	0	546.0	117.25	374.4	3
R207	1278.54		1278.54		0.000	295.8	1	0	0	502.9	775.64	361.6	3
R208						3600.0	0	0	0			328.2	1
R209	986.62		986.62		0.000	626.9	1	0	0	622.8	363.82	370.7	2
R210	559.13		559.13		0.000	2.3	1	0	0	519.0	40.13	404.6	3
R211	1101.02		1101.02		0.000	1460.2	1	0	0	573.9	527.12	350.9	2
C201	427.52		427.52		0.000	1.1	1	0	0	252.6	174.92	214.7	2
C202	4714.01		4714.01		0.000	246.1	1	0	0	335.9	4378.11	214.7	2
C203						3600.0	0	0	0			214.7	2
C204						3600.0	0	0	0			213.1	1
C205	666.29		666.29		0.000	2.5	1	0	0	228.1	438.19	214.7	2
C206	315.09		315.09		0.000	2.5	1	0	0	235.8	79.29	214.7	2
C207	440.16		440.16		0.000	18.8	1	0	0	251.6	188.56	214.5	2
C208	1543.80		1543.80		0.000	3.5	1	0	0	226.7	1317.10	214.5	2
RC201	395.10		395.10		0.000	0.3	1	0	0	395.1	0.00	360.2	3
RC202	610.78		610.78		0.000	85.1	1	0	16	396.2	214.58	338.0	3
RC203	743.72		743.72		0.000	8.7	1	0	0	383.4	360.32	326.9	3
RC204	809.28		809.28		0.000	121.9	1	0	0	357.2	452.08	299.7	3
RC205	387.95		387.95		0.000	1.1	1	0	0	385.4	2.55	338.0	3
RC206	395.10		395.10		0.000	0.5	1	0	0	395.1	0.00	324.0	3
RC207	408.96		408.96		0.000	5.5	1	0	0	385.0	23.96	298.3	3
RC208			1250.49	1250.49		3600.0	1	0	10			269.1	2

Table 24: Detailed results for the VRPTW-CNC with $\rho = 0.05$ for the $n = 25$-customer instances.

	VRPTW-CNC with Limited Fleet												
	Bounds							Cuts		Costs		VRPTW	
Instance	Opt	UB	LB_{root}	LB_{tree}	$\%Gap$	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	c^*	K^*
R101	1053.65		1053.65		0.000	0.1	1	0	0	1044.0	9.65	1044.0	12
R102	1173.31		1173.31		0.000	1.5	1	0	10	1022.8	150.51	909.0	11
R103	1973.92		1958.71		0.000	844.3	15	0	20	903.8	1070.12	772.9	9
R104						3600.0	0	0	0			625.4	6
R105	1060.42		1060.42		0.000	1.0	1	3	10	951.6	108.82	899.3	9
R106	1386.29		1386.29		0.000	22.4	1	5	15	917.3	468.99	793.0	8
R107	2654.29		2654.29		0.000	2197.4	1	0	50	806.4	1847.89	711.1	7
R108						3600.0	0	0	0			617.7	6
R109	1247.35		1243.79		0.000	248.8	7	2	100	904.1	343.25	786.8	8
R110	1907.69		1891.35		0.000	2673.9	13	0	41	810.0	1097.69	697.0	7
R111	2096.82		2096.82		0.000	1825.1	1	0	58	860.2	1236.62	707.2	7
R112			3291.32	3291.32		3600.0	1	0	20			630.2	6
C101	610.70		610.70		0.000	0.7	1	0	0	386.5	224.20	362.4	5
C102						3600.0	0	0	0			361.4	5
C103						3600.0	0	0	0			361.4	5
C104						3600.0	0	0	0			358.0	5
C105	1276.98		1276.98		0.000	1.7	1	0	0	386.5	890.48	362.4	5
C106	1002.36		1002.36		0.000	1.3	1	0	0	391.8	610.56	362.4	5
C107	2201.95		2201.95		0.000	19.3	1	0	0	391.8	1810.15	362.4	5
C108	4188.50		4188.50		0.000	648.8	1	0	0	412.6	3775.90	362.4	5
C109						3600.0	0	0	0			362.4	5
RC101	1113.49		1113.49		0.000	1.6	1	69	0	960.1	153.39	944.0	8
RC102	1640.49		1587.49		0.000	1458.6	63	20	100	927.5	712.99	822.5	7
RC103			2760.06	2760.06		3600.0	1	8	70			710.9	6
RC104						3600.0	0	0	0			545.8	5
RC105	1324.88		1281.84		0.000	381.1	49	27	100	991.0	333.88	855.3	8
RC106	1399.53		1399.53		0.000	134.1	1	7	80	743.1	656.43	723.2	6
RC107			1983.91	1990.01		3600.0	2	3	97			642.7	6
RC108			2433.50	2433.50		3600.0	1	2	20			598.1	6
R201	930.63		930.63		0.000	3.7	1	0	0	910.6	20.03	791.9	6
R202	1035.65		1035.65		0.000	574.9	1	0	0	879.6	156.05	698.5	5
R203						3600.0	0	0	0			605.3	5
R204						3600.0	0	0	0			506.4	2
R205						3600.0	0	0	0			690.1	4
R206						3600.0	0	0	0			632.4	4
R207						3600.0	0	0	0			575.5	3
R208						3600.0	0	0	0			487.7	2
R209						3600.0	0	0	0			600.6	4
R210						3600.0	0	0	0			645.6	4
R211						3600.0	0	0	0			535.5	3
C201						3600.0	0	0	0			360.2	3
C202						3600.0	0	0	0			360.2	3
C203						3600.0	0	0	0			359.8	3
C204						3600.0	0	0	0			588.7	2
C205						3600.0	0	0	0			359.8	3
C206						3600.0	0	0	0			359.8	3
C207						3600.0	0	0	0			359.6	3
C208						3600.0	0	0	0			350.5	2
RC201	769.85		769.85		0.000	1.6	1	0	0	754.9	14.95	684.8	5
RC202	1020.38		1020.38		0.000	2013.3	1	0	18	744.6	275.78	613.6	5
RC203						3600.0	0	0	0			555.3	4
RC204						3600.0	0	0	0			444.2	3
RC205	796.43		796.43		0.000	44.2	1	0	0	773.8	22.63	630.2	5
RC206	787.27		787.27		0.000	25.2	1	0	0	753.4	33.87	610.0	5
RC207						3600.0	0	0	0			558.6	4
RC208						3600.0	0	0	0			476.7	3

Table 25: Detailed results for the VRPTW-CNC with $\rho = 0.05$ for the $n = 50$-customer instances.

	VRPTW-CNC with Limited Fleet												
	Bounds							Cuts		Costs		VRPTW	
Instance	Opt	UB	LB_{root}	LB_{tree}	$\%Gap$	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	c^*	K^*
R101	1666.31		1662.80		0.000	6.9	13	4	5	1639.9	26.41	1637.7	20
R102	2167.61		2167.61		0.000	1784.4	1	0	44	1625.8	541.82	1466.6	18
R103						3600.0	0	0	0			1208.7	14
R104						3600.0	0	0	0			971.5	11
R105	1627.28		1626.01		0.000	143.5	3	2	87	1464.6	162.68	1355.3	15
R106						3600.0	0	0	0			1234.6	13
R107						3600.0	0	0	0			1064.6	11
R108						3600.0	0	0	0			932.1	10
R109			1989.83	1989.83		3600.0	1	0	30			1146.9	13
R110						3600.0	0	0	0			1068.0	12
R111						3600.0	0	0	0			1048.7	12
R112						3600.0	0	0	0			948.6	10
C101	1302.45		1302.45		0.000	96.4	1	0	0	848.0	454.45	827.3	10
C102						3600.0	0	0	0			827.3	10
C103						3600.0	0	0	0			826.3	10
C104						3600.0	0	0	0			822.9	10
C105	2652.48		2652.48		0.000	261.8	1	0	0	848.7	1803.78	827.3	10
C106	3874.05		3874.05		0.000	2945.2	1	0	0	899.8	2974.25	827.3	10
C107	4474.14		4474.14		0.000	1348.1	1	0	0	850.0	3624.14	827.3	10
C108						3600.0	0	0	0			827.3	10
C109						3600.0	0	0	0			827.3	10
RC101	1992.10		1992.10		0.000	668.1	1	67	79	1703.3	288.80	1619.8	15
RC102			2631.22	2631.22		3600.0	1	12	20			1457.4	14
RC103						3600.0	0	0	0			1258.0	11
RC104						3600.0	0	0	0			1132.3	10
RC105			2297.47	2297.47		3600.0	1	36	30			1513.7	15
RC106						3600.0	0	0	0			1372.7	12
RC107						3600.0	0	0	0			1207.8	12
RC108						3600.0	0	0	0			1114.2	11
R201						3600.0	0	0	0			1143.2	8
R202						3600.0	0	0	0			1029.6	8
R203						3600.0	0	0	0			870.8	6
R204						3600.0	0	0	0			731.3	5
R205						3600.0	0	0	0			949.8	5
R206						3600.0	0	0	0			875.9	5
R207						3600.0	0	0	0			794.0	4
R208						3600.0	0	0	0			701.0	4
R209						3600.0	0	0	0			854.8	5
R210						3600.0	0	0	0			900.5	6
R211						3600.0	0	0	0			746.7	4
C201	1554.57		1554.57		0.000	1183.6	1	0	0	621.2	933.37	589.1	3
C202						3600.0	0	0	0			589.1	3
C203						3600.0	0	0	0			588.7	3
C204						3600.0	0	0	0			588.1	3
C205						3600.0	0	0	0			586.4	3
C206						3600.0	0	0	0			586.4	3
C207						3600.0	0	0	0			585.8	3
C208						3600.0	0	0	0			585.8	3
RC201						3600.0	0	0	0			1261.8	9
RC202						3600.0	0	0	0			1092.3	8
RC203						3600.0	0	0	0			923.7	5
RC204						3600.0	0	0	0			783.5	4
RC205						3600.0	0	0	0			1154.0	7
RC206						3600.0	0	0	0			1051.1	7
RC207						3600.0	0	0	0			962.9	6
RC208						3600.0	0	0	0			776.1	4

Table 26: Detailed results for the VRPTW-CNC with $\rho = 0.05$ for the $n = 100$-customer instances.

	VRPTW-CNC with Limited Fleet											VRPTW	
	Bounds							Cuts		Costs			
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	%*Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	c^*	K^*
R101	628.09		628.09		0.000	0.0	1	0	0	617.1	10.99	617.1	8
R102	791.37		791.37		0.000	0.1	1	2	0	610.9	180.47	547.1	7
R103	1960.39		1947.33		0.000	14.7	9	1	34	555.4	1404.99	454.6	5
R104	3173.53		3173.53		0.000	10.0	1	0	0	461.0	2712.53	416.9	4
R105	642.29		642.29		0.000	0.1	1	0	0	566.7	75.58	530.5	6
R106	1118.04		1107.45		0.000	3.4	5	5	54	556.8	561.24	465.4	5
R107	2513.80		2513.80		0.000	2.5	1	0	0	501.8	2012.00	424.3	4
R108	2662.98		2662.98		0.000	10.3	1	0	0	464.3	2198.68	397.3	4
R109	745.60		745.60		0.000	0.1	1	0	0	507.2	238.40	441.3	5
R110	935.71		935.71		0.000	0.6	1	0	0	483.3	452.41	444.1	5
R111	2271.24		2271.24		0.000	6.6	1	0	20	524.5	1746.74	428.8	4
R112	2284.28		2284.28		0.000	90.4	1	0	30	431.0	1853.28	393.0	4
C101	412.11		412.11		0.000	0.1	1	0	0	191.3	220.81	191.3	3
C102	16045.40		16045.40		0.000	31.6	1	0	0	278.8	15766.60	190.3	3
C103	52235.50		52235.50		0.000	246.9	1	0	0	295.2	51940.30	190.3	3
C104			78952.00	78952.00		3600.0	1	0	20			186.9	3
C105	1034.83		1034.83		0.000	0.6	1	0	26	245.5	789.33	191.3	3
C106	413.39		413.39		0.000	0.1	1	0	0	191.3	222.09	191.3	3
C107	2029.24		2029.24		0.000	0.6	1	0	20	245.5	1783.74	191.3	3
C108	2741.29		2741.29		0.000	1.2	1	0	0	252.7	2488.59	191.3	3
C109	6660.74		6660.74		0.000	15.3	1	0	0	252.3	6408.44	191.3	3
RC101	618.46		618.46		0.000	0.2	1	14	0	476.0	142.46	461.1	4
RC102	1243.56		1243.56		0.000	0.9	1	0	0	360.1	883.46	351.8	3
RC103	1568.70		1568.70		0.000	6.9	1	0	0	358.6	1210.10	332.8	3
RC104	2039.96		2039.96		0.000	17.5	1	0	0	348.2	1691.76	306.6	3
RC105	987.47		987.47		0.000	1.9	1	0	50	471.2	516.27	411.3	4
RC106	729.71		729.71		0.000	0.8	1	0	0	373.4	356.31	345.5	3
RC107	1154.46		1154.46		0.000	6.4	1	0	0	342.3	812.16	298.3	3
RC108	1917.07		1917.07		0.000	49.4	1	0	0	329.5	1587.57	294.5	3
R201	521.42		521.42		0.000	0.2	1	0	0	509.7	11.72	463.3	4
R202	605.28		605.28		0.000	0.5	1	0	0	526.9	78.38	410.5	4
R203	2044.34		2044.34		0.000	256.9	1	0	0	502.9	1541.44	391.4	3
R204						3600.0	0	0	0			355.0	2
R205	591.08		591.08		0.000	8.4	1	0	0	575.3	15.78	393.0	3
R206	780.08		780.08		0.000	25.5	1	0	0	556.0	224.08	374.4	3
R207	2054.18		2054.18		0.000	595.7	1	0	0	502.9	1551.28	361.6	3
R208						3600.0	0	0	0			328.2	1
R209	1350.44		1350.44		0.000	555.5	1	0	0	622.8	727.64	370.7	2
R210	599.25		599.25		0.000	6.2	1	0	0	519.0	80.25	404.6	3
R211	1628.15		1628.15		0.000	2499.0	1	0	10	573.9	1054.25	350.9	2
C201	602.45		602.45		0.000	1.2	1	0	0	252.6	349.85	214.7	2
C202	9092.12		9092.12		0.000	149.9	1	0	0	335.9	8756.22	214.7	2
C203						3600.0	0	0	0			214.7	2
C204						3600.0	0	0	0			213.1	1
C205	1104.48		1104.48		0.000	1.7	1	0	0	228.1	876.38	214.7	2
C206	394.39		394.39		0.000	2.2	1	0	0	235.8	158.59	214.7	2
C207	628.72		628.72		0.000	20.2	1	0	0	251.6	377.12	214.5	2
C208	2860.90		2860.90		0.000	3.9	1	0	0	226.7	2634.20	214.5	2
RC201	395.10		395.10		0.000	0.3	1	0	0	395.1	0.00	360.2	3
RC202	773.05		773.05		0.000	20.9	1	0	6	459.5	313.55	338.0	3
RC203	1097.70		1097.70		0.000	20.2	1	0	0	389.9	707.80	326.9	3
RC204	1261.35		1261.35		0.000	99.5	1	0	0	357.2	904.15	299.7	3
RC205	390.50		390.50		0.000	1.5	1	0	0	385.4	5.10	338.0	3
RC206	395.10		395.10		0.000	1.1	1	0	0	395.1	0.00	324.0	3
RC207	432.93		432.93		0.000	9.3	1	0	0	385.0	47.92	298.3	3
RC208	1976.22		1976.22		0.000	3580.9	1	0	10	549.9	1426.32	269.1	2

Table 27: Detailed results for the VRPTW-CNC with $\rho = 0.1$ for the $n = 25$-customer instances.

	VRPTW-CNC with Limited Fleet												
	Bounds							Cuts		Costs		VRPTW	
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	c^*	K^*
R101	1062.50		1062.50		0.000	0.1	1	0	0	1046.1	16.40	1044.0	12
R102	1323.82		1321.22		0.000	4.1	3	0	16	1022.8	301.03	909.0	11
R103	3040.03		2996.70		0.000	2421.0	39	0	22	957.2	2082.83	772.9	9
R104						3600.0	0	0	0			625.4	6
R105	1163.15		1163.15		0.000	2.5	1	6	40	965.0	198.15	899.3	9
R106	1854.48		1854.48		0.000	140.4	1	3	49	918.3	936.18	793.0	8
R107	4481.77		4481.77		0.000	2719.3	1	0	60	831.6	3650.17	711.1	7
R108						3600.0	0	0	0			617.7	6
R109	1589.14		1568.32		0.000	403.1	41	4	67	908.1	681.04	786.8	8
R110	3005.39		2985.36		0.000	2015.2	5	0	50	810.0	2195.39	697.0	7
R111	3324.60		3308.85		0.000	3432.8	5	0	64	893.0	2431.60	707.2	7
R112			5862.68	5862.68		3600.0	1	0	30			630.2	6
C101	834.89		834.89		0.000	0.4	1	0	0	386.5	448.39	362.4	5
C102						3600.0	0	0	0			361.4	5
C103						3600.0	0	0	0			361.4	5
C104						3600.0	0	0	0			358.0	5
C105	2167.47		2167.47		0.000	2.2	1	0	0	386.5	1780.97	362.4	5
C106	1612.92		1612.92		0.000	1.9	1	0	0	391.8	1221.12	362.4	5
C107	4012.09		4012.09		0.000	27.0	1	0	0	391.8	3620.29	362.4	5
C108	7939.61		7939.61		0.000	545.7	1	0	0	442.5	7497.11	362.4	5
C109						3600.0	0	0	0			362.4	5
RC101	1261.40		1261.40		0.000	1.5	1	66	0	973.8	287.60	944.0	8
RC102		2353.49	2059.37	2351.43	0.088	3600.0	253	8	100			822.5	7
RC103			4625.39	4625.39		3600.0	1	12	70			710.9	6
RC104						3600.0	0	0	0			545.8	5
RC105	1656.93		1565.72		0.000	732.6	143	28	100	999.3	657.63	855.3	8
RC106	2041.49		2041.49		0.000	122.0	1	7	70	796.0	1245.49	723.2	6
RC107			3211.32	3211.32		3600.0	1	5	100			642.7	6
RC108			4163.40	4163.40		3600.0	1	2	20			598.1	6
R201	950.66		950.66		0.000	4.6	1	0	0	910.6	40.05	791.9	6
R202	1191.69		1191.69		0.000	956.3	1	0	0	879.6	312.09	698.5	5
R203						3600.0	0	0	0			605.3	5
R204						3600.0	0	0	0			506.4	2
R205						3600.0	0	0	0			690.1	4
R206						3600.0	0	0	0			632.4	4
R207						3600.0	0	0	0			575.5	3
R208						3600.0	0	0	0			487.7	2
R209						3600.0	0	0	0			600.6	4
R210						3600.0	0	0	0			645.6	4
R211						3600.0	0	0	0			535.5	3
C201						3600.0	0	0	0			360.2	3
C202						3600.0	0	0	0			360.2	3
C203						3600.0	0	0	0			359.8	3
C204						3600.0	0	0	0			588.7	2
C205						3600.0	0	0	0			359.8	3
C206	427.89		625.13		0.000	767.2	1	0	0	389.6	38.29	359.8	3
C207						3600.0	0	0	0			359.6	3
C208						3600.0	0	0	0			350.5	2
RC201	784.79		784.79		0.000	1.0	1	0	0	754.9	29.89	684.8	5
RC202	1246.51		1246.51		0.000	2571.8	1	0	10	807.2	439.31	613.6	5
RC203						3600.0	0	0	0			555.3	4
RC204						3600.0	0	0	0			444.2	3
RC205	815.37		815.37		0.000	23.9	1	0	0	779.6	35.77	630.2	5
RC206	821.14		821.14		0.000	24.0	1	0	0	753.4	67.74	610.0	5
RC207						3600.0	0	0	0			558.6	4
RC208						3600.0	0	0	0			476.7	3

Table 28: Detailed results for the VRPTW-CNC with $\rho = 0.1$ for the $n = 50$-customer instances.

	VRPTW-CNC with Limited Fleet												
	Bounds							Cuts		Costs		VRPTW	
Instance	*Opt*	*UB*	LB_{root}	LB_{tree}	*%Gap*	Time	B&B	2-path	SR	$\sum c_P$	$\sum f_P$	c^*	K^*
R101	1692.72		1690.52		0.000	7.5	13	1	7	1639.9	52.82	1637.7	20
R102	2704.63		2704.63		0.000	2664.8	1	0	75	1650.1	1054.53	1466.6	18
R103						3600.0	0	0	0			1208.7	14
R104						3600.0	0	0	0			971.5	11
R105	1772.00		1767.77		0.000	377.6	17	3	82	1491.3	280.70	1355.3	15
R106						3600.0	0	0	0			1234.6	13
R107						3600.0	0	0	0			1064.6	11
R108						3600.0	0	0	0			932.1	10
R109						3600.0	0	0	0			1146.9	13
R110						3600.0	0	0	0			1068.0	12
R111						3600.0	0	0	0			1048.7	12
R112						3600.0	0	0	0			948.6	10
C101	1756.26		1756.26		0.000	202.5	1	0	0	848.7	907.56	827.3	10
C102						3600.0	0	0	0			827.3	10
C103						3600.0	0	0	0			826.3	10
C104						3600.0	0	0	0			822.9	10
C105	4456.26		4456.26		0.000	141.8	1	0	0	848.7	3607.56	827.3	10
C106						3600.0	0	0	0			827.3	10
C107	8098.27		8098.27		0.000	1722.1	1	0	0	850.0	7248.27	827.3	10
C108						3600.0	0	0	0			827.3	10
C109						3600.0	0	0	0			827.3	10
RC101	2255.46		2255.46		0.000	516.5	1	70	54	1752.0	503.46	1619.8	15
RC102			3568.49	3568.49		3600.0	1	16	10			1457.4	14
RC103						3600.0	0	0	0			1258.0	11
RC104						3600.0	0	0	0			1132.3	10
RC105			2809.45	2809.45		3600.0	1	36	10			1513.7	15
RC106						3600.0	0	0	0			1372.7	12
RC107						3600.0	0	0	0			1207.8	12
RC108						3600.0	0	0	0			1114.2	11
R201						3600.0	0	0	0			1143.2	8
R202						3600.0	0	0	0			1029.6	8
R203						3600.0	0	0	0			870.8	6
R204						3600.0	0	0	0			731.3	5
R205						3600.0	0	0	0			949.8	5
R206						3600.0	0	0	0			875.9	5
R207						3600.0	0	0	0			794.0	4
R208						3600.0	0	0	0			701.0	4
R209						3600.0	0	0	0			854.8	5
R210						3600.0	0	0	0			900.5	6
R211						3600.0	0	0	0			746.7	4
C201	2487.94		2487.94		0.000	2665.8	1	0	0	621.2	1866.74	589.1	3
C202						3600.0	0	0	0			589.1	3
C203						3600.0	0	0	0			588.7	3
C204						3600.0	0	0	0			588.1	3
C205						3600.0	0	0	0			586.4	3
C206						3600.0	0	0	0			586.4	3
C207						3600.0	0	0	0			585.8	3
C208						3600.0	0	0	0			585.8	3
RC201						3600.0	0	0	0			1261.8	9
RC202						3600.0	0	0	0			1092.3	8
RC203						3600.0	0	0	0			923.7	5
RC204						3600.0	0	0	0			783.5	4
RC205						3600.0	0	0	0			1154.0	7
RC206						3600.0	0	0	0			1051.1	7
RC207						3600.0	0	0	0			962.9	6
RC208						3600.0	0	0	0			776.1	4

Table 29: Detailed results for the VRPTW-CNC with $\rho = 0.1$ for the $n = 100$-customer instances.