Submitted to *Transportation Science*
manuscript TS-2015-0173

Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

Exact Algorithms for the Vehicle Routing Problem with Time Windows and Combinatorial Auction

Zhenzhen Zhang
Department of Industrial Systems Engineering and Management, National University of Singapore, 1 Engineering Drive 2, Singapore 117576, zhenzhenzhang222@gmail.com

Zhixing Luo
School of Management and Engineering, Nanjing University, Nanjing 210093, P.R. China, luozx.hkphd@gmail.com
Department of Industrial Systems Engineering and Management, National University of Singapore, 1 Engineering Drive 2, Singapore 117576

Hu Qin
School of Management, Huazhong University of Science and Technology, Wuhan 430000, PR China, tigerqin1980@gmail.com

Andrew Lim
Department of Industrial Systems Engineering and Management, National University of Singapore, 1 Engineering Drive 2, Singapore 117576, isealim@nus.edu.sg
International Center of Management Science and Engineering, School of Management and Engineering, Nanjing University, Nanjing 210093, P.R. China

In this paper, we address an interesting variant of the vehicle routing problem with time windows (VRPTW) faced by the China National Petroleum Corporation (CNPC). The CNPC owns a limited number of tanker trucks for delivering the petroleum to oil stations within specific time windows in the regular seasons. However, during the peak seasons, some requests need to be outsourced to external third-party logistics (3PL) companies. These companies provide several bids, each of which includes the oil stations to be served and the corresponding charge. The CNPC needs to select some bids and design routes for the self-owned trucks, so that all requests are satisfied and the total cost is minimized. To study this problem, we formulate it into an arc-flow model and a set-partitioning model, and propose six families of valid inequalities to strengthen the set-partitioning model. Based on the set-partitioning model, we propose a branch-and-price-and-cut algorithm and a branch-and-bound algorithm to solve the problem exactly. The proposed algorithms are tested on instances generated according to the well-known Solomon benchmark instances (Solomon 1987) for the VRPTW and read-world data of the CNPC. The computational experiments demonstrate the effectiveness of the proposed algorithms.

Key words: VRPTW; combinational auction; outsource; branch-and-price-and-cut; branch-and-bound
History: Received: August 2015; revision received: December 2016, February 2018.

1. Introduction

The vehicle routing problem with time windows and combinatorial auction (VRPTWCA) extends the classic vehicle routing problem with time windows (VRPTW) by allowing customers to be outsourced to third-party logistics (3PL) providers through combinatorial auction. Given a set of customers, a fleet of vehicles based at the depot, and a set of bids each of which consists of a group of customers and a charge to serve these customers, the VRPTWCA requires selecting some bids to outsource the corresponding customers to 3PL providers and designing routes for the self-owned vehicles to serve the remaining customers subject to the time windows and capacity constraints. The objective is to minimize the total travel cost of the vehicles and the total money charged by the 3PL providers. Each customer must be either served by a vehicle, or allocated exactly to one bid that contains the customer. The VRPTWCA is motivated by the oil distribution of the China National Petroleum Corporation (CNPC), which uses tank trucks to deliver oil from a depot to oil stations located in different places. The CNPC chooses to outsource part of its oil transportation tasks to 3PL providers for the following three reasons. First, the CNPC faces demand fluctuation within a year, and does not have enough vehicles to service all the oil stations during the peak seasons. Second, outsourcing the deliveries of some remote oil stations to regional 3PL providers may reduce costs. According to the Chinese government regulations, it is compulsory for state-owned companies (e.g. CNPC) to put outsourced services out to tender. To attract more 3PL providers to take part in the auction, the CNPC allows 3PL providers to submit bids that list the oil stations they are interested in and the charge to serve the oil stations, as the 3PL providers are more willing to serve groups of close oil stations. A simple example with 12 oil stations, two vehicles, and four bids is shown in Figure 1, where the bids are marked with ellipses and solid lines indicate that the corresponding bids are selected.

Outsourcing transportation tasks has become common practice in many enterprises in the past decades. However, there are limited vehicle routing problems involving outsourcing decisions in the literature. Chu (2005) introduced the vehicle routing problem with a private fleet and common carrier (VRPPC), where each customer is either served by private vehicles or outsourced to a common carrier, named the subcontractor, with a fixed charge. The total cost of the VRPPC consists of three parts: the total fixed cost and total routing cost of the private fleet, and the total sum charged by the subcontractor. Chu (2005) solved the VRPPC with a two-stage heuristic consisting of a modified savings-based construction heuristic (Clarke and Wright 1964) and a simple improvement heuristic. Bolduc et al. (2007) proposed a *SRI* (selection, routing, and improvement) heuristic for the VRPPC, which achieved better results than the heuristic proposed by Chu (2005). Bolduc et al. (2008) proposed a perturbation meta-heuristic where perturbation was invoked in the

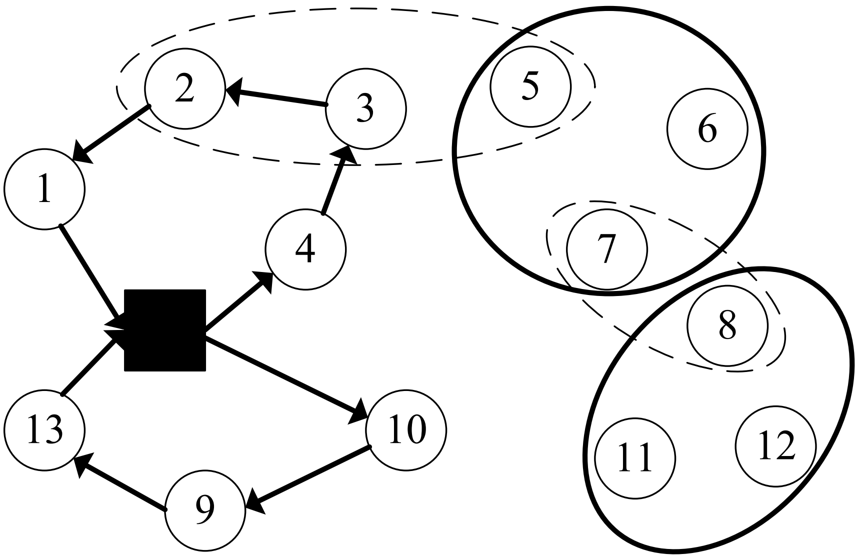

Figure 1 **An example of the VRPTWCA**

construction and improvement phases, and demonstrated the superiority of the proposed metaheuristic through computational experiments. The tabu search has also been applied to solve the VRPPC successfully, for example, the works of Côté and Potvin (2009) and Potvin and Naud (2011). Recently, Stenger et al. (2013) and Vidal et al. (2016) studied two vehicle routing problems which generalize the VRPPC, and their solution approaches, namely an adaptive large neighborhood search and a large neighborhood search with implicit customer selection, can also be applied to solve the VRPPC. To the best of our knowledge, no exact algorithms solve the VRPPC in the literature.

The VRPPC can be viewed as a special case of the VRPTWCA where the time window of each customer is sufficiently large, and each customer corresponds to exactly one bid. Note that the fixed cost of a vehicle can be transformed into the routing cost of the vehicle from a mathematical point of view. Compared with the VRPPC, the VRPTWCA has two major advantages. First, the VRPTWCA is easier to implement in practice. To implement the VRPPC, the fixed charge for outsourcing each individual customer to a 3PL provider has to be given in advance. However, the 3PL providers may not be willing to share or even know this information as they are not interested in each individual customer. Second, the VRPTWCA may pay less to the 3PL providers for outsourcing the same customers, because 3PL providers tend to offer a lower price for a group of customers as a whole than the total price for each individual customer.

The contributions of this paper are summarized as follows. First, we introduce a new variant of the VRPTW which outsources groups of customers to 3PL providers through a combinatorial auction. Compared with the VRPPC which also involves outsourcing decisions, the way of outsourcing in our problem may be easier to implement in practice and reduce the prices charged by the 3PL providers. Second, we formulate the problem into an arc-flow model and a set-partitioning model, and propose six families of valid inequalities to strengthen the set-partitioning model. The inequalities are demonstrated to be effective in improving the lower bounds yielded by the linear programming (LP) relaxation of the model. Third, we propose a branch-and-price-and-cut (BPC) algorithm and a branch-and-bound algorithm (BB) to solve the problem. The two proposed algorithms are tested on instances generated according to the classic Solomon benchmark instances for the VRPTW (Solomon 1987) and the real-world data of the CNPC, respectively. The computational results show that the BPC algorithm performs slightly better than the BB algorithm in general. However, the BPC algorithm does not dominate the BB algorithm overall, because the BB algorithm can solve some instances where the BPC algorithm fails to reach optimality in four hours of computational time. In addition, both algorithms can solve the practical instances to optimality in four hours of computational time.

The remainder of the paper is structured as follows. In Section 2, we present the arc-flow model and the set-partitioning model for the VRPTWCA, and the six families of valid inequalities to strengthen the set-partitioning model. In Section 3, we introduce the pricing problem, the label-setting algorithm to solve the pricing problem, and a tabu search to accelerate the convergence of the column generation. We also introduce other components of the branch-and-price-and-cut algorithm, including a heuristic to generate the initial solution, the separation algorithms for the valid inequalities and the branching strategies. Section 4 describes the branch-and-bound algorithm in detail. Section 5 is devoted to the computational experiments. Section 6 concludes the paper with some closing remarks.

2. Mathematical Models

In this section, we describe the arc-flow model and the set-partitioning model for the VRPTWCA as well as some valid inequalities for strengthening the set-partitioning model. First, we introduce some necessary notation.

Let $G = (V, A)$ be the complete directed graph in which the VRPTWCA is defined. $V = \{0, 1, \ldots, n\}$ is the node set, where node 0 indicates the depot and $N = \{1, \ldots, n\}$ represents the set of customers. $A = \{(i, j) \mid i, j \in V, i \neq j\}$ is the arc set. Each node $i \in V$ has a demand q_i, a service time s_i, and a time window $[e_i, l_i]$. For node 0, $q_0 = s_0 = 0$, e_0 is the earliest time for a vehicle to leave the depot, and l_0 is the latest time to return to the depot. For node $i \in N$, e_i and l_i are the earliest and latest service starting times of customer i, respectively. If a vehicle arrives at customer $i \in N$ before e_i, it must wait until e_i to start the service. Each arc $(i, j) \in A$ has a travel cost $c_{i,j}$ and a travel time $t_{i,j}$. We assume both $c_{i,j}$ and $t_{i,j}$ follow the triangle inequality, i.e. $c_{i,j} \leq c_{i,k} + c_{k,j}$ and $t_{i,j} \leq t_{i,k} + t_{k,j}$ for all $i, j, k \in V$. Let K be the set of homogeneous vehicles with capacity Q, and U be the set of bids. For each bid $u \in U$, let p_u be the price of u and $o_{i,u}$ ($i \in N$) be a binary indicator, which is equal to 1 if u contains customer i and 0 otherwise.

The VRPTWCA aims to minimize the total cost of the chosen bids and routes of the vehicles and satisfy the following constraints. (1) Every customer is served exactly once by either one bid or a vehicle; (2) the demand and time window of each customer are respected; (3) each vehicle starts and terminates at the depot; and (4) the vehicle capacity is not exceeded.

2.1. Arc-Flow Model

To build a two-index arc-flow model, we add a duplicate of node 0, namely node $n+1$, to graph G, and create a new graph $G' = (V', A')$ where $V' = V \cup \{n+1\}$ is the node set and $A' = A \cup \{(i, n+1) \mid i \in N\}$ is the arc set. The decision variables of the arc-flow model are listed as follows:

- $x_{i,j}$ ($(i,j) \in A'$): a binary decision variable which is equal to 1 if arc (i,j) is traveled through by a vehicle, and 0 otherwise;
- y_u ($u \in U$): a binary decision variable which is equal to 1 if bid u is selected, and 0 otherwise;
- a_i ($i \in V'$): service starting time at node i; and
- d_i ($i \in V'$): the load of a vehicle after serving customer i.

The arc-flow model for the VRPTWCA is formulated as follows:

$$\min \sum_{(i,j)\in A'} c_{i,j} x_{i,j} + \sum_{u \in U} p_u y_u, \tag{1}$$

$$\text{s.t.} \quad \sum_{(i,j)\in A'} x_{i,j} + \sum_{u \in U} o_{i,u} y_u = 1, \quad \forall i \in N, \tag{2}$$

$$\sum_{(0,i)\in A'} x_{0,i} = \sum_{(i,n+1)\in A'} x_{i,n+1} \le |K|, \tag{3}$$

$$\sum_{(j,i)\in A'} x_{j,i} = \sum_{(i,j)\in A'} x_{i,j}, \quad \forall i \in N, \tag{4}$$

$$a_j \ge (a_i + s_i + t_{i,j}) x_{i,j}, \quad \forall (i,j) \in A', \tag{5}$$

$$d_i \ge (d_j + q_j) x_{i,j}, \quad \forall (i,j) \in A', \tag{6}$$

$$e_i \le a_i \le l_i, \quad i \in V', \tag{7}$$

$$d_i \le Q, \quad \forall i \in V', \tag{8}$$

$$a_i \ge 0, \quad \forall i \in V', \tag{9}$$

$$d_i \ge 0, \quad \forall i \in V', \tag{10}$$

$$x_{i,j} \in \{0,1\}, \quad \forall (i,j) \in A', \tag{11}$$

$$y_u \in \{0,1\}, \quad \forall u \in U. \tag{12}$$

Objective (1) minimizes the total travel cost of the vehicles and the total amount charged by the 3PL providers. Constraints (2) ensure that each customer is either visited by a vehicle or allocated a bid. Constraints (3) guarantee that the number of vehicles used cannot exceed the fleet size. Constraints (4) are the flow conservation constraints. Constraints (5) and (6) ensure the consistency of the service starting time and the load of a vehicle at different nodes, respectively. Constraints (7) are the time windows constraints of the customers, and constraints (8) are the capacity constraints of the vehicles.

2.2. Set-Partitioning Model

Let R denote the set of feasible routes satisfying the time windows and capacity constraints. Let c_r be the cost of route r, and $\alpha_{i,r}$ be a binary number which is equal to 1 if route r visits customer i and 0 otherwise. The binary decision variable θ_r is set to 1 if route r is selected in an optimal solution and 0 otherwise. The set-partitioning model can be formulated as follows:

$$\min \sum_{r\in R} c_r\theta_r + \sum_{u\in U} p_u y_u, \tag{13}$$

$$\text{s.t.} \quad \sum_{r\in R} \alpha_{i,r}\theta_r + \sum_{u\in U} o_{i,u} y_u = 1, \quad \forall i \in N, \tag{14}$$

$$\sum_{r\in R} \theta_r \le |K|, \tag{15}$$

$$y_u \in \{0,1\}, \quad \forall u \in U, \tag{16}$$

$$\theta_r \in \{0,1\}, \quad \forall r \in R. \tag{17}$$

The objective function (13) minimizes the total cost of the selected routes and bids. Constraints (14) guarantee that each customer is either visited by a vehicle or outsourced to a 3PL company, whereas the number of available vehicles is limited by constraints (15).

2.3. Valid Inequalities

In this section, we present six families of valid inequalities for the set-partitioning model. Let $\beta_{i,j,r}$ be a binary number equal to 1 if and only if arc $(i,j) \in A'$ is traveled through by route $r \in R$. For a subset $S \subset N$, let $\delta^-(S) = \{(i,j) \in A' \mid i \notin S, j \in S\}$, $\delta^+(S) = \{(i,j) \in A' \mid i \in S, j \notin S\}$, and $A(S) = \{(i,j) \in A' \mid i,j \in S\}$.

2.3.1. K-Path Inequalities The k-path inequalities are a family of classic valid inequalities for the VRPTW introduced by Kohl et al. (1999). For a subset $S \subseteq N$, let k_S be the minimum number of vehicles required to serve all the customers in S. The k-path inequalities related to S are modified to include the bids as follows:

$$\sum_{(i,j)\in\delta^+(S)} \sum_{r\in R} \beta_{i,j,r}\theta_r + \sum_{u\in U} \eta_u y_u \ge k_S, \tag{18}$$

where

$$\eta_u = \begin{cases} k_S, \ S \subseteq u \\ k_S - 1, \ u \bigcap S \ne \emptyset \text{ and } S \not\subset u \\ 0, \ \text{otherwise.} \end{cases} \tag{19}$$

PROPOSITION 1. *Inequality (18) is valid for the VRPTWCA.*

Proof. The validity of inequality (18) can be proved by considering the following three cases.

In the first case where $y_u = 0$ for any bid u with $u \bigcap S \neq \emptyset$, inequality (18) reduces to the classic k-path inequality.

For the second case in which a bid u with $S \subseteq u$ and $y_u = 1$ exists, the left-hand-side value of inequality (18) is at least k_S. Thus, inequality (18) holds.

Last, if there exists at least one bid having $u \bigcap S \neq \emptyset$, $S \not\subseteq u$, and $y_u = 1$, we have $\sum_{u \in U} \eta_u y_u \geq k_S - 1$. If not all the customers in S are covered by the bids, we have $\sum_{(i,j) \in \delta^+(S)} \sum_{r \in R} \beta_{i,j,r} \theta_r \geq 1$. Thus, inequality (18) holds. If all the customers in S are only covered by different bids, then among these bids there must exist at least two bids having $u \bigcap S \neq \emptyset$, $S \not\subseteq u$, and $y_u = 1$. Because $k_S \geq 2$, $\sum_{u \in U} \eta_u y_u \geq k_S - 1 + (k_S - 1) \geq k_S$, and therefore, inequality (18) holds. □

2.3.2. Capacity Inequalities The capacity inequalities are modified from the classic capacity inequalities for the VRP (Naddef and Rinaldi 2001) with consideration of the bids. The capacity inequalities share the same form as the k-path inequalities, but may be stronger than the k-path inequalities when the lower bound for the number of necessary vehicles in the k-path inequalities is determined by the capacity constraint. Given a subset $S \subseteq N$, the capacity inequality with respect to S is defined as:

$$\sum_{(i,j) \in \delta^+(S)} \sum_{r \in R} \beta_{i,j,r} \theta_r + \sum_{u \in U} \left\lceil \frac{1}{Q} \sum_{i \in u \bigcap S} q_i \right\rceil y_u \geq \left\lceil \frac{1}{Q} \sum_{i \in S} q_i \right\rceil. \tag{20}$$

For the k-path inequality (18), if $k_S = \left\lceil \frac{1}{Q} \sum_{i \in S} q_i \right\rceil$, it is possible that $\left\lceil \frac{1}{Q} \sum_{i \in u \bigcap S} q_i \right\rceil < k_S - 1$ and then the capacity inequality is stronger than the k-path inequality. Therefore, the capacity inequalities are not dominated by the k-path inequalities.

PROPOSITION 2. *Inequality (20) is valid for the VRPTWCA.*

Proof. The following inequality naturally holds for the VRPTWCA:

$$\sum_{(i,j) \in \delta^+(S)} q_i x_{i,j} + \sum_{u \in U} y_u \sum_{i \in u \bigcap S} q_i \geq \sum_{i \in S} q_i. \tag{21}$$

By dividing both sides of inequality (21) by the vehicle capacity Q and taking the ceiling values, we have

$$\sum_{(i,j) \in \delta^+(S)} \left\lceil \frac{q_i}{Q} \right\rceil x_{i,j} + \sum_{u \in U} \left\lceil \frac{1}{Q} \sum_{i \in u \bigcap S} q_i \right\rceil y_u \geq \left\lceil \frac{1}{Q} \sum_{i \in S} q_i \right\rceil. \tag{22}$$

Because $\left\lceil \frac{q_i}{Q} \right\rceil = 1$, inequality (20) is valid for the VRPTWCA. □

2.3.3. Strengthened Capacity Inequalities The capacity inequalities can be strengthened by lifting the left-hand-side coefficients associated with decision variables $\{y\}$ from $\left\lceil \frac{1}{Q}\sum_{i\in u\bigcap S} q_i \right\rceil$ to $\left\lceil \frac{1}{Q}\sum_{i\in S} q_i \right\rceil - \left\lceil \frac{1}{Q}\sum_{i\in S\backslash u} q_i \right\rceil$. However, not all the coefficients of variables $\{y\}$ in the capacity inequalities can be lifted. Whether the coefficient of a variable y_u can be lifted depends on the coefficients of other variables $\{y\}$. More specifically, let $\gamma(u,S)$ be the coefficient associated with variable y_u in inequality (20), which is $\left\lceil \frac{1}{Q}\sum_{i\in u\bigcap S} q_i \right\rceil$ initially. Then, according to a given order of bids, we try to lift the coefficient $\gamma(u,S)$ one by one if it respects the following proposition.

PROPOSITION 3. *For a bid $u\in U$, if $\gamma(u',S)\geq\gamma(u',S\backslash u)$ for all $u'\bigcap u=\emptyset$, then $\gamma(u,S)$ can be lifted to* $\left\lceil \frac{1}{Q}\sum_{i\in S} q_i \right\rceil - \left\lceil \frac{1}{Q}\sum_{i\in S\backslash u} q_i \right\rceil$.

Proof. We only need to prove the validity of lifting $\gamma(u,S)$ to $\left\lceil \frac{1}{Q}\sum_{i\in S} q_i \right\rceil - \left\lceil \frac{1}{Q}\sum_{i\in S\backslash u} q_i \right\rceil$ when $y_u=1$, since the coefficient of a bid u with $y_u=0$ does not have any effect on the inequality. Suppose $y_u=1$ for a bid $u\in U$ and the capacity inequality (20) holds for the subset S before $\gamma(u,S)$ is lifted. Then $y_{u'}=0$ for all $u'\bigcap u\neq\emptyset$. Let $U'=\{u' \mid u'\in U, u'\bigcap u=\emptyset\}$ and $S'=S\backslash u$. Then the capacity inequality for S', i.e.,

$$\sum_{(i,j)\in\delta^+(S')}\sum_{r\in R}\beta_{i,j,r}\theta_r+\sum_{u'\in U'}\gamma(u',S')y_{u'}\geq\left\lceil\frac{1}{Q}\sum_{i\in S'}q_i\right\rceil, \tag{23}$$

holds. Because

$$\sum_{(i,j)\in\delta^+(S)}\sum_{r\in R}\beta_{i,j,r}\theta_r\geq\sum_{(i,j)\in\delta^+(S')}\sum_{r\in R}\beta_{i,j,r}\theta_r \tag{24}$$

and

$$\gamma(u',S)\geq\gamma(u',S'),\ \forall\ u'\in U', \tag{25}$$

we have

$$\sum_{(i,j)\in\delta^+(S)}\sum_{r\in R}\beta_{i,j,r}\theta_r+\sum_{u'\in U'}\gamma(u',S)y_{u'}\geq\left\lceil\frac{1}{Q}\sum_{i\in S\backslash u}q_i\right\rceil, \tag{26}$$

which is equivalent to

$$\sum_{(i,j)\in\delta^+(S)}\sum_{r\in R}\beta_{i,j,r}\theta_r+\sum_{u'\in U'}\gamma(u',S)y_{u'}+(\left\lceil\frac{1}{Q}\sum_{i\in S}q_i\right\rceil-\left\lceil\frac{1}{Q}\sum_{i\in S\backslash u}q_i\right\rceil)y_u\geq\left\lceil\frac{1}{Q}\sum_{i\in S}q_i\right\rceil. \tag{27}$$

So lifting $\gamma(u,S)$ to $\left\lceil \frac{1}{Q}\sum_{i\in S} q_i \right\rceil - \left\lceil \frac{1}{Q}\sum_{i\in S\backslash u} q_i \right\rceil$ is valid. □

Remark: the sequence to lift the coefficients of variables $\{y\}$ in a capacity inequality affects the quality of the resulting strengthened capacity inequality. For example, given two bids u_1 and u_2, if $\gamma(u_1, S)$ is lifted first, then it may not be possible to lift $\gamma(u_2, S)$, and vice versa. The detailed computation of lifted coefficients is explained in Section 3.3.

The strengthened capacity inequalities are stronger than or at least equivalent to the capacity inequalities (20). Consider the following case where $S = \{1, 2, 3, 4\}$, $q_1 = q_2 = q_3 = q_4 = 10$, $Q = 15$, $u_1 = \{1, 2, 5\}$ and $u_2 = \{3, 4, 6\}$. The capacity inequality with respect to S is $\sum_{(i,j)\in\delta^+(S)} \beta_{i,j,r}\theta_r + 2y_{u_1} + 2y_{u_2} \geq 3$, while the strengthened capacity inequality with respect to S can be $\sum_{(i,j)\in\delta^+(S)} \beta_{i,j,r}\theta_r + y_{u_1} + 2y_{u_2} \geq 3$ or $\sum_{(i,j)\in\delta^+(S)} \beta_{i,j,r}\theta_r + 2y_{u_1} + y_{u_2} \geq 3$. Here, not both $\gamma(u_1, S)$ and $\gamma(u_2, S)$ can be lifted to 1. Suppose $\gamma(u_1, S)$ is lifted to 1, i.e. $\gamma(u_1, S) = \left\lceil \frac{\sum_{i\in S} q_i}{Q} \right\rceil - \left\lceil \frac{\sum_{i\in S\setminus u_1} q_i}{Q} \right\rceil$. Then, $\gamma(u_2, S)$ cannot be lifted to 1 according to Proposition 3, because $\gamma(u_1, S\setminus u_2) = 2 > \gamma(u_1, S) = 1$. On the contrary, if $\gamma(u_2, S)$ is first lifted to 1, then $\gamma(u_1, S)$ cannot also be lifted to 1.

2.3.4. Bid Partitioning Inequalities The bid partitioning inequalities are a family of simple but useful inequalities for the VRPTWCA. For a subset $S \subseteq N$, let $U^1 = \{u \in U | S \subseteq u\}$ and $U^2 = U \setminus U^1$. If there exist no feasible partitions from U^2 to cover each customer in S exactly once, at least one route or a bid in U^1 must be selected to cover nodes in S, and hence the following inequality is valid:

$$\sum_{(i,j)\in\delta^+(S)} \sum_{r\in R} \beta_{i,j,r}\theta_r + \sum_{u\in U^1} y_u \geq 1. \tag{28}$$

2.3.5. Clique Inequalities According to the characteristics of the VRPTWCA, we design the following two types of clique inequalities. The first type only involves the bids. For a subset $W \subseteq U$, if the bids in W form a clique, i.e., any bid in W has at least one customer in common with all the other bids in W, then the following inequality is valid:

$$\sum_{u\in W} y_u \leq 1. \tag{29}$$

The second type involves two bids and one arc. For two bids u_1 and u_2, and an arc $(i, j) \in A'$, if $u_1 \cap u_2 \neq \emptyset$, $i \in u_1$, and $j \in u_2$, then bids u_1, u_2, and arc (i, j) form a clique, and therefore, the following inequality is valid:

$$y_{u_1} + y_{u_2} + \sum_{r\in R} (\beta_{i,j,r} + \beta_{j,i,r})\theta_r \leq 1. \tag{30}$$

2.3.6. Subset-row Inequalities Although the subset-row inequalities were first introduced for the VRPTW by Jepsen et al. (2008), they are valid for the general set-partitioning model. Therefore, the subset-row inequalities can be applied in the VRPTWCA with a slight modification. For a subset $S \subseteq N$ and an integer k $(1 < k \leq |S|)$, the modified subset-row inequalities with inclusion of the bids are defined as follows:

$$\sum_{r\in R}\left\lfloor \frac{1}{k}\sum_{i\in S}\alpha_{i,r}\right\rfloor \theta_r + \sum_{u\in U}\left\lfloor \frac{1}{k}\sum_{i\in S} o_{i,u}\right\rfloor y_u \leq \left\lfloor \frac{|S|}{k}\right\rfloor. \tag{31}$$

The idea behind the subset-row inequalities is intuitive. For example, when $k=2$ and $|S|=3$, the subset-row inequalities simply ensure that at most one route or bid covering more than one node in S can be selected.

Handling the dual value of inequality (31) in the pricing problem requires additional effort, and becomes more and more difficult as the sizes of $|S|$ and k increase. To achieve a good balance between the complexity of the pricing problem and the strength of the subset-row inequality, a common approach in the literature is to limit the sizes of $|S|$ and k. In our implementation, we restrict the size of $|S|$ to 3 and k to 2. Then, the subset-row inequality (31) reduces to the following inequality:

$$\sum_{r\in R}\rho(r,S)\theta_r + \sum_{u\in U}\rho(u,S)z_u \leq 1, \tag{32}$$

where $\rho(r,S)$ and $\rho(u,S)$ are equal to 1 if route r and bid u cover more than one node in S, respectively, and 0 otherwise.

3. Branch-and-Price-and-Cut Algorithm

In this section, we introduce the branch-and-price-and-cut algorithm for the VRPTWCA. The core of the branch-and-price-and-cut algorithm is column generation (Desrosiers and Lübbecke 2005), which is used to solve the LP relaxation of the set-partitioning model. Column generation is an iteration algorithm which dynamically generates the part of columns that possibly constitute an optimal solution. Note that a column is actually a route in the VRP. Initially, the LP relaxation of the set-partitioning model is restricted to a subset of columns, i.e., $R' \subset R$, leading to the restricted linear master problem (RLMP). The RLMP can be quickly solved by the simplex method to provide primal and dual solutions. Based on the dual solution of the incumbent RLMP, a pricing subproblem is solved to identify other columns of negative reduced cost, which are possibly part of an optimal solution. If such columns are found, they are added to R' and the new RLMP is solved again. Otherwise, the procedure terminates and the optimal solution of the incumbent RLMP is also the optimal solution of the LP relaxation of the original set-partitioning model.

In the remainder of this section, we first propose a backtracking heuristic to construct an initial feasible solution. Then, we introduce the pricing problem derived from the set-partitioning model, the label-setting algorithm to solve the pricing problem, three techniques for accelerating the label-setting algorithm, and a tabu search to heuristically solve the pricing problem to speed up the convergence of column generation. Later, we present the algorithms used to separate the violated inequalities. Last, we describe the branching strategies that guarantee the branch-and-price-and-cut algorithm will eventually achieve an optimal integer solution.

3.1. Initial Feasible Solution

An initial feasible solution not only provides initial columns required by the column generation, but also an initial upper bound for the branch-and-price-and-cut algorithm. The initial solution of our branch-and-price-and-cut algorithm is generated by a backtracking heuristic. The details of the heuristic are as follows. First, the bids are sorted in decreasing order according to the number of customers in a bid. Then, the algorithm greedily selects compatible bids according to the given sequence to cover as many customers as possible. Here, two bids are compatible if they do not have any overlap. After obtaining a set of compatible bids, the regret insertion proposed by Potvin and Rousseau (1993) is invoked to generate routes to cover all the remaining customers. If a feasible plan is obtained, the procedure terminates; otherwise, the procedure returns to the last step to attempt a different selection of bids. In our experiments, this backtracking method can obtain feasible initial solutions for all instances quickly.

3.2. Pricing Problem

Let π_i $(i \in N)$ denote the dual value of constraint (14) and π_0 denote the dual value of constraint (15). Let τ_S $(S \subseteq N)$, υ_S $(S \subseteq N)$, ϕ_S $(S \subseteq N)$, $\varphi_{u_1,u_2,i,j}$ $(u_1, u_2 \in U, (i,j) \in A')$ and λ_S $(S \subseteq N)$ denote the dual values of the k-path inequalities (18), the capacity inequalities (20), the bid partitioning inequalities (28), the clique inequalities (30), and the subset-row inequalities (32), respectively. Let $\mu_{i,j} = \sum_{(i,j)\in\delta^+(S), S\subseteq N}(\tau_S + \upsilon_S + \phi_S) + \sum_{u_1\in U}\sum_{u_2\in U\setminus\{u_1\}}\varphi_{u_1,u_2,i,j}$ and $\mathcal{S}$ denote the set of the subset-row inequalities (32) generated. The pricing problem can be formulated as follows:

$$\min_{r\in R}\; c_r - \sum_{i\in N}\alpha_{i,r}\pi_i - \pi_0 - \sum_{(i,j)\in A'}\beta_{i,j,r}\mu_{i,j} - \sum_{S\in\mathcal{S}}\rho(r,S)\lambda_S, \tag{33}$$

where the objective (33) minimizes the reduced cost of a route. The pricing problem is equivalent to the elementary shortest path problem with resource constraints (ESPPRC) (Feillet et al. 2004). A path is called elementary only if each node is visited once at most. The ESPPRC can typically be solved by the label-setting algorithm discussed below (Feillet et al. 2004, Righini and Salani 2008).

3.2.1. Label-Setting Algorithm The label-setting algorithm is a type of dynamic programming algorithm that enumerates all non-dominated routes by state propagation. A state is a partial route from node 0 to any node $i \in V'$, represented by a label $L_i = (C_i, T_i, D_i, \{\Omega_{i,S}\}_{S\in\mathcal{S}}, \{V_{i,k}\}_{k\in V'})$, where

- C_i is the reduced cost of the partial route;
- T_i is the earliest time to start the service at node i along the path;
- D_i is the accumulated demand at node i along the path;
- $\Omega_{i,S}$ is the number of nodes in S that have been visited by the partial route; and
- $V_{i,k}$ is a binary resource which is equal to 1 if node k can be reached by the partial route, and 0 otherwise.

Note that $\Omega_{i,S}$ can only be 0, 1, 2, or 3 since $|S|$ is restricted to 3 in our implementation.

For an arc $(i,j) \in A'$, a label L_i associated with node i can be extended to node j if $V_{i,j} = 1$. If so, a new label $L_j = (C_j, T_j, L_j, \{\Omega_{j,S}\}_{S\in\mathcal{S}}, \{V_{j,k}\}_{k\in V'})$ is created, where

$$\Omega_{j,S} = \begin{cases} \Omega_{i,S} + 1, & j \in S \\ \Omega_{i,S}, & j \notin S, \end{cases} \tag{34}$$

$$C_j = C_i + c_{i,j} - \pi_j - \mu_{i,j} - \sum_{S\in\mathcal{S}, \Omega_{i,S}<2, \Omega_{j,S}=2} \lambda_S, \tag{35}$$

$$T_j = \max\{e_j, T_i + s_i + t_{i,j}\}, \tag{36}$$

$$D_j = D_i + q_j, \tag{37}$$

$$V_{j,k} = \begin{cases} 0, & k = j \text{ or } V_{i,k} = 0 \text{ or } D_j + q_k > Q \text{ or } T_j + s_j + t_{j,k} > l_k \\ 1, & \text{otherwise} \end{cases} \tag{38}$$

During the label extension, dominance rules are applied to eliminate labels that are dominated by other labels. The dominated labels can be safely discarded and the label-setting algorithm is still guaranteed to get an optimal solution. For two labels $L_i^1 = (C_i^1, T_i^1, D_i^1, \{\Omega_{i,S}^1\}_{S\in\mathcal{S}}, \{V_{i,k}^1\}_{k\in V'})$ and $L_i^2 = (C_i^2, T_i^2, D_i^2, \{\Omega_{i,S}^2\}_{S\in\mathcal{S}}, \{V_{i,k}^2\}_{k\in V'})$ associated with the same node i, let $\mathcal{S}' = \{S \mid S \in \mathcal{S}, \Omega_{i,S}^1 \text{mod} 2 > \Omega_{i,S}^2 \text{mod} 2\}$ be the set of subset-row inequalities that may affect the partial route represented by L_i^1 but may not affect the partial route represented by L_i^2. Then, L_i^1 dominates L_i^2 if

$$C_i^1 - \sum_{S\in\mathcal{S}'} \lambda_S \le C_i^2, \tag{39}$$

$$T_i^1 \le T_i^2, \tag{40}$$

$$D_i^1 \le D_i^2, \tag{41}$$

$$V_{i,k}^1 \ge V_{i,k}^2, \ \forall\, k \in V', \tag{42}$$

and at least one of the above inequalities is strict. Condition (39) can be further tightened by reducing the number of λ_S added to the left-hand side of the inequality. Let $N(L_i^1, L_i^2)$ be the set of nodes that either have been visited by L_i^1 or are reachable from L_i^2, and $\mathcal{S}'' = \mathcal{S}' \setminus \{S \mid S \in \mathcal{S}, |S \cap N(L_i^1, L_i^2)| < 2\}$. For a subset $S \in \mathcal{S}$, if $|S \cap N(L_i^1, L_i^2)| < 2$, it means $\rho(r, S) = 0$ for any route r generated by connecting L_i^1 and a completion of L_i^2 to the end depot. Therefore, we can replace $\mathcal{S}'$ in condition (39) by $\mathcal{S}''$.

Many techniques have been proposed to speed up label-setting algorithms in the literature. In our implementation, we have adopted the following techniques to accelerate our label-setting algorithm. First, we adopt the bounded bidirectional search, which divides the extension of the label-setting algorithm into the *forward extension* and the *backward extension* according to a chosen critical resource. In the forward extension, labels are extended from node 0 to node $n+1$, while in the backward extension, labels are extended in the opposite direction from node $n+1$ to node 0. Both the forward extension and the backward extension stop when the consumption of the critical resource reaches a half. Then the forward partial routes and the backward partial routes associated with the same node are joined together pair by pair to find out the route with the minimal cost. In our implementation, we choose the time as the critical resource. For details about the bounded bidirectional search, the reader is referred to Righini and Salani (2006, 2008).

Second, we adopt the method proposed by Martinelli et al. (2014) which uses ng-route relaxation to speed up the pricing of elementary routes. The ng-routes, first proposed by Baldacci et al. (2011), are a compromise between elementary routes and non-elementary routes. In the ng-route relaxation, each customer is associated beforehand with a set of nearest customers including itself, called *ng-set*. When generating a ng-route, a customer i cannot be visited if i has been visited and belongs to the intersection of the ng-sets of the customers after i. On the one hand, the larger the size of ng-sets is, the closer the ng-route is to an elementary route. On the other hand, the larger the size of ng-sets is, the more difficult it is to generate an optimal ng-route since the dominance relationship between the ng-routes becomes weaker. Martinelli et al. (2014) used ng-route relaxation to speed up a label-setting algorithm as follows. Initially, each customer is associated with a small-size ng-set. Then, an optimal ng-route is generated by the label-setting algorithm. If the optimal ng-route is not elementary, the ng-sets of the customers in the cycles of the optimal ng-route are enlarged to destroy these cycles, and the label-setting algorithm is invoked again. This process iterates until an optimal ng-route obtained by the label-setting algorithm becomes elementary.

Third, we apply the q-route relaxation (Christofides et al. 1981) to compute a lower bound on the reduced cost of a partial route to reach node $n+1$ (or node 0 in the backward extension), and use this lower bound to prune labels that cannot be extended to a route with negative reduced cost. A q-route is a non-elementary route which considers only either the capacity constraints or the time windows constraints, and hence can be computed fast by dynamic programming. The q-route relaxation can be strengthened by eliminating cycles of 2 customers at the expense of a slight increase in the time complexity of the dynamic programming algorithm. For the details, please see Christofides et al. (1981). Note that the dual values from the subset-row inequalities are dropped in the q-route relaxation, since they cannot be handled in the dynamic programming algorithm.

3.2.2. Tabu Search Column Generator For the pricing problem, it is not necessary to invoke the time-consuming label-setting algorithm as long as routes with negative reduced cost can be found quickly in other ways. In the literature, one common way to speed up the convergence of column generation is to use a heuristic to find columns with negative reduced cost first, and to invoke the label-setting algorithm only after the heuristic fails (Desaulniers et al. 2008, Archetti et al. 2011, Luo et al. 2014). We follow the same approach, and propose a tabu search heuristic to find columns with negative reduced cost.

The tabu search is run several times with different promising initial routes, which are usually the basic variables of the current RLMP. At each iteration, the search moves to the best feasible neighboring solution defined by the given operators. In our work, three types of operators are used: (1) remove a node from the current route, (2) insert a node into the current route by considering all possible insertion places, and (3) exchange two nodes with exactly one node visited by the current route. To check the feasibility of the resulting route, we use the advanced segment-based evaluation procedure (Vidal et al. 2014). It can examine the feasibility in constant time, and has been proved to be quite efficient for VRP variants with complex constraints (Zhang et al. 2015, Lim et al. 2017). To avoid cycling, the tabu list is used to forbid the reverse moves. Only the allowable operators are considered.

3.3. Cutting Separation Algorithms

The k-path inequalities (18) and the capacity inequalities (20) can be separated simultaneously as they are of the same form. Therefore, we use four heuristics, namely the partial enumeration scheme proposed by Desaulniers (2010), the shrinking heuristic and the route-based connected component heuristic proposed by Archetti et al. (2011), and the tabu search proposed by Cordeau (2006), to separate these inequalities simultaneously. Note that in the k-path inequalities, it is difficult to compute the minimum number of vehicles to serve a subset of customers in general, because it is equivalent to solving a VRP for the subset of customers. However, determining whether a subset of customers can be served by a single route is easier as it is equivalent to finding a feasible route serving the customers, which can be done by a label-setting algorithm. Therefore, when checking a k-path inequality related to a subset of customers, we invoke the label-setting algorithm to test whether the customers can be served in a single feasible route. If so, the right-hand-side value is set to 1; otherwise, it is set to 2. Also, the number of customers in a subset is limited to 20 to avoid situations in which the label-setting algorithm takes a lot of time to terminate.

Let $(\bar{\theta}_r, \bar{y}_u)$ $(r \in R, u \in U)$ be the optimal solution of the LP relaxation of the set-partitioning model, and $\bar{x}_{i,j} = \sum_{r \in R} \beta_{i,j,r} \bar{\theta}_r$. When implementing the four heuristics, we adjust them slightly to make them work better in our problem. The modifications are as follows. First, when implementing the partial enumeration scheme, the size of a subset of customers is limited to 15. Second, when implementing the shrinking heuristic, two super nodes i and j are chosen to shrink if they maximize $\bar{x}_{i,j} + \bar{x}_{j,i} - \sum_{u \in U} \left\lceil \frac{1}{Q} \sum_{i \in u \cap k} q_i \right\rceil \bar{y}_u$ where k is the super node into which i and j are shrunk.

For the capacity inequalities (20), an additional greedy procedure is invoked in the above heuristics to lift the coefficients of the bids to obtain the strengthened capacity inequalities. First, the bids are sorted in decreasing value of $\bar{y}_u$, and ties are broken in increasing value of $\sum_{i \in u \cap S} q_i$, where S is the subset of customers being considered. Then, the coefficients of the bids are lifted one by one according to the sorted order.

For the bid partitioning inequality (28), we only need to consider the set S of customers covered by at least one bid, because the other customers must be served by a vehicle and the inequality obviously holds. Thus, the bid partitioning inequality is separated into three steps. First, we select the set of customers covered by at least one bid u with $\bar{y}_u > 0$, denoted by $\bar{N}$. Then, we enumerate all subsets of $\bar{N}$ with size no larger than 3, and test whether there exists a bid partition for each subset by full enumeration. Next, we check whether the bid partitioning inequality is violated for each such subset.

The separation problem for the first type of clique inequality (29) is equivalent to the maximum weighted clique problem (Baldacci et al. 2008). Therefore, we adopt the branch-and-bound algorithm proposed by Östergård (2002) to separate these inequalities. The second type of clique inequality (30) is so simple it can be separated by complete enumeration.

For the subset-row inequality (32), we also use a full enumeration algorithm to separate the subset. The full enumeration algorithm is very fast because the size of the considered subsets of customers is limited to 3.

3.4. Branching Strategies

In the branch-and-price-and-cut algorithm, four types of branching rules are used in a hierarchy to ensure that the algorithm eventually gets an integer optimal solution. The four types of branching rules are compatible with the structure of the pricing problem and do not have any influence on the label-setting algorithm.

Branching on the number of vehicles. Let $(\bar{\theta}_r, \bar{y}_u)$ $(r \in R, u \in U)$ be an optimal solution of the LP relaxation of the set-partitioning model. Let $\bar{m}$ be the number of vehicles used in the optimal solution, i.e., $\sum_{r\in R}\bar{\theta}_r$. If $\bar{m}$ is fractional, we branch on the value of $\bar{m}$ to obtain two child nodes by forcing $\sum_{r\in R}\theta_r \le \lfloor \bar{m} \rfloor$ and $\sum_{r\in R}\theta_r \ge \lceil \bar{m} \rceil$, respectively.

Branching on the number of bids covering a customer. Let $\bar{y}_i$ be the number of bids covering customer $i \in N$, i.e., $\sum_{u\in U} o_{i,u}\bar{y}_u$. We branch on the value of $\bar{y}_{i^*}$ where $\bar{y}_{i^*}$ is fractional and closest to 0.5. One child node is obtained by forcing $\sum_{u\in U} o_{i^*,u}y_u = 0$, namely, dropping bid u for each $o_{i^*,u} = 1$. The other node is generated by forcing $\sum_{u\in U} o_{i^*,u}y_u = 1$, which can be enforced by deleting all the arcs associated with customer i^*. In addition, if all the bids covering customer i^* also cover another customer j^*, then all arcs associated with customer j^* and the bids which cover customer j^* but not customer i^* can be deleted.

Branching on bids. We branch on bid u^* for which $\bar{y}_{u^*}$ is fractional and closest to 0.5. Two child nodes are generated by forcing $y_{u^*} = 0$ and $y_{u^*} = 1$, respectively. Here, $y_{u^*} = 1$ can be enforced by deleting all the arcs (i,j) and (j,i) for each customer $i \in N$ with $o_{i,u^*} = 1$, and bid $u \in U\backslash\{u^*\}$ with $u \cap u^* \neq \emptyset$.

Branching on arcs. Let $\bar{x}_{i,j}$ be the number of vehicles traveling through arc (i,j) in the optimal solution, i.e., $\sum_{r\in R}\beta_{i,j,r}\bar{\theta}_r$. We branch on the value of arc (i^*,j^*) with $\bar{x}_{i^*,j^*}$ closest to 0.5. Two child nodes are generated by forbidding and forcing vehicles to travel through arc (i^*,j^*), respectively. Forcing vehicles to travel through an arc can be achieved by deleting arcs (i^*,k) where $k \neq j^*$ and (k,j^*) where $k \neq i^*$.

4. Branch-and-Bound Algorithm

In this section, we propose a branch-and-bound algorithm to solve the VRPTWCA, based on the observation that the decision space of the VRPTWCA can be divided into two different levels: the selection of bids in the first level and the routing plan making in the second level. Suppose that we can compute a lower bound on the cost associated with a combination of bids. Then the problem to determine the optimal selection of bids based on these lower bounds is actually the weighted maximum independent set problem (WMIS) (Hifi 1997) under the lower bound scheme. Once the selection of bids is fixed, the VRPTWCA reduces to the classic VRPTW if $|K| > 1$ or the classic traveling salesman problem with time windows (TSPTW) (Baldacci et al. 2012) if $|K| = 1$. The WMIS is equivalent to the weighted maximum clique problem (WMCP), which can be solved efficiently by the branch-and-bound algorithm proposed by Östergård (2002). For the VRPTW, we can use the branch-and-price-and-cut algorithm proposed in Section 3. For the TSPTW, we can use the bi-directional dynamic programming proposed by Li (2009).

The lower bound scheme is based on the LP-relaxed set-covering model for the VRPTW. Let $\mathcal{S}^1$, $\mathcal{S}^2$ and $\mathcal{S}^3$ be the set of k-path inequalities, capacity inequalities, and subset-row inequalities that have been introduced to strengthen the LP-relaxed set-covering model, respectively. Then the LP-relaxed set-covering model for the VRPTW is as follows:

$$\textbf{SC:} \quad \min \sum_{r \in R} c_r x_r \tag{43}$$

$$\text{s.t.} \quad \sum_{r \in R} \alpha_{i,r} x_r \geq 1, \qquad i \in N \tag{44}$$

$$\sum_{(i,j) \in \delta^+(S)} \sum_{r \in R} \beta_{i,j,r} x_r \geq k_S, \qquad S \in \mathcal{S}^1 \tag{45}$$

$$\sum_{(i,j) \in \delta^+(S)} \sum_{r \in R} \beta_{i,j,r} x_r \geq \left\lceil \frac{1}{Q} \sum_{i \in S} q_i \right\rceil, \qquad S \in \mathcal{S}^2 \tag{46}$$

$$\sum_{r \in R} \rho(r,S) x_r \leq 1, \qquad S \in \mathcal{S}^3 \tag{47}$$

$$x_r \geq 0, \qquad r \in R \tag{48}$$

Let $\bar{\mu}_i$ $(i \in N)$, $\bar{\tau}_S$ $(S \in \mathcal{S}^1)$, $\bar{\upsilon}_S$ $(S \in \mathcal{S}^2)$ and $\bar{\lambda}_S$ $(S \in \mathcal{S}^3)$ be the optimal dual values of constraints (44), (45), (46) and (47), respectively. Let $OPT(SC)$ be the optimal cost of model **SC**. For a bid $u \in U$, let

$$\bar{p}_u = p_u - \sum_{i \in u} \bar{\mu}_i - \sum_{S \in \mathcal{S}^1} \bar{\tau}_S \eta_u - \sum_{S \in \mathcal{S}^2} \bar{\upsilon}_S \left\lceil \frac{1}{Q} \sum_{i \in u \bigcap S} q_i \right\rceil - \sum_{S \in \mathcal{S}^3} \bar{\lambda}_S \rho(u,S). \tag{49}$$

Let $\{y_u^*\}_{u \in U}$ be a solution satisfying the following constraints:

$$\sum_{u \in U} \alpha_{i,u} y_u^* \leq 1, \quad i \in N. \tag{50}$$

THEOREM 1. *$OPT(SC) + \sum_{u \in U} \bar{p}_u y_u^*$ is a valid lower bound on the cost of any feasible VRPTWCA solution where a bid $u \in U$ is selected if and only if $y_u^* = 1$.*

Proof. Let $\{x'_r, y_u^*\}_{r \in R, u \in U}$ be a feasible solution for the VRPTWCA. Then $\{x'_r\}_{r \in R}$ is a feasible solution for the following problem:

$$\min \sum_{r \in R} c_r x_r \tag{51}$$

$$\text{s.t.} \quad \sum_{r \in R} \alpha_{i,r} x_r = 1 - \sum_{u \in U} \alpha_{i,u} y_u^*, \quad i \in N \tag{52}$$

$$\sum_{(i,j) \in \delta^+(S)} \sum_{r \in R} \beta_{i,j,r} x_r \geq k_S - \sum_{u \in U} \eta_u y_u^*, \quad S \in \mathcal{S}^1 \tag{53}$$

$$\sum_{(i,j) \in \delta^+(S)} \sum_{r \in R} \beta_{i,j,r} x_r \geq \left\lceil \frac{1}{Q} \sum_{i \in S} q_i \right\rceil - \sum_{u \in U} \left\lceil \frac{1}{Q} \sum_{i \in u \bigcap S} q_i \right\rceil y_u^*, \quad S \in \mathcal{S}^2 \tag{54}$$

$$\sum_{r \in R} \rho(r,S) x_r \leq 1 - \sum_{u \in U} \rho(u,S) y_u^*, \quad S \in \mathcal{S}^3 \tag{55}$$

$$x_r \geq 0, \quad r \in R \tag{56}$$

Since $\bar{\mu}_i$ $(i \in N)$, $\bar{\tau}_S$ $(S \in \mathcal{S}^1)$, $\bar{\upsilon}_S$ $(S \in \mathcal{S}^2)$ and $\bar{\lambda}_S$ $(S \in \mathcal{S}^3)$ is a dual feasible solution for the above problem, we have:

$$\sum_{i \in N} \bar{\mu}_i (1 - \sum_{u \in U} \alpha_{i,u} y_u^*) + \sum_{S \in \mathcal{S}^1} \bar{\tau}_S (k_S - \sum_{u \in U} \eta_u y_u^*) + \sum_{S \in \mathcal{S}^2} \bar{\upsilon}_S (\left\lceil \frac{1}{Q} \sum_{i \in S} q_i \right\rceil - \sum_{u \in U} \left\lceil \frac{1}{Q} \sum_{i \in u \bigcap S} q_i \right\rceil y_u^*) + \sum_{S \in \mathcal{S}^3} \bar{\lambda}_S (1 - \sum_{u \in U} \rho(u,S) y_u^*) \leq \sum_{r \in R} c_r x_r', \tag{57}$$

which is equivalent to

$$OPT(SC) + \sum_{u \in U} \bar{p}_u y_u^* \leq \sum_{r \in R} c_r x_x' + \sum_{u \in U} p_u y_u^*. \tag{58}$$

Since inequality (58) holds for any feasible solution $\{x_r', y_u^*\}_{r \in R, u \in U}$ of the VRPTWCA, Theorem (1) holds. □

Model **SC** can be solved by the column generation introduced in Section 3. Meanwhile, the k-path inequalities, the capacity inequalities, and the subset-row inequalities can also be separated by the algorithms introduced in Section 3.3.

The branch-and-bound algorithm is detailed in Algorithm 1, where parameter B stores the selected bids, i is the index of the bid to be determined and UB is the upper bound. The algorithm starts from B&B($\emptyset, 1, \infty$), and then goes through each bid and decides if the current bid should be included in set B or not. When all bids are done, it is checked if a better upper bound can be found. As we can see, if the pruning condition (Line 2) is removed, the branch-and-bound algorithm simply enumerates all the feasible combinations of bids. Therefore, the branch-and-bound algorithm is guaranteed to achieve an optimal solution eventually. However, the efficiency of the branch-and-bound algorithm heavily depends on the qualities of the lower bound and the upper bound, and the speed to solve the VRPTW or the TSPTW. If these three components are good enough, the branch-and-bound may possibly outperform the branch-and-price-and-cut algorithm in Section 3. Here, to get an initial upper bound of high quality, we enumerate all possible selections of bids, then we solve the reduced VRPTW problems with a simple tabu search. The idea of tabu search is similar to the one described in Section 3.2.2 but uses the classic neighborhood operators for VRPTW: relocate one customer, swap two customers and 2-opt (Bräysy and Gendreau 2005). This procedure might be quite time-consuming for large instances, thus we limit its computational time to 600 seconds.

5. Computational Results

To evaluate the performance of the proposed exact algorithms, experiments are conducted on two classes of VRPTWCA instances derived from the well-known Solomon's VRPTW instances (Solomon 1987) and the real-world data of the CNPC, respectively. The algorithm are coded in

Algorithm 1 The branch-and-bound algorithm B&B

B&B(Set B, Index i, Upper bound UB)

if $i > |U|$
    if $OPT(SC) + \sum_{u \in B} \bar{p}_u y_u < UB$
        Solve the VRPTW or the TSPTW defined on the set of customers $V \backslash \{j \in N | \alpha_{j,u} = 1, u \in B\}$
        Update UB if a better solution is achieved
    return
if bid u_i is compatible with any bid $u' \in B$
    B&B($B \cup \{u_i\}, i+1, UB$) // select the bid u_i
B&B($B, i+1, UB$) // ignore bid u_i

Java and performed on a workstation equipped with an Intel(R) Xeon(R) CPU E5-1603 clocked at 2.80 GHz (Quad Core, but only a single thread is used) with 64 GB RAM running Linux operating system. The RLMP is solved by ILOG CPLEX solver 12.5.1. The time limit for each instance is four hours in the experiments. The generated instances and optimal solutions can be downloaded from our website `http://www.computational-logistics.org/orlib/VRPTWCA`.

5.1. Test Instances

Two classes of instances are used to test the performance of the proposed algorithms. The detailed information about the instances are described in this section.

5.1.1. Instances Derived from the Solomon Instances The Solomon's VRPTW instances are divided into three classes according to the characteristics of the geographical distribution of the customers: c (clustered distribution), r (random distribution), and rc (a mix of clustered and random distribution). Each class is further divided into two types: type-1 with narrow time windows and type-2 with wider time windows. The name $r203$ stands for the third type-2 instance of class r. In total, there are 29 type-1 instances and 27 type-2 instances. Because every instance contains 100 customers, they are usually called 100-customer instances. Based on these instances, 25- and 50-customer instances can be built by keeping the first 25 and 50 customers, respectively.

Our VRPTWCA instances are generated by introducing the bid information into the VRPTW instances. Each bid is actually a route that has a different depot and vehicle capacity specified by a 3PL company. More precisely, for each instance, the total number of bids is first specified. A location is randomly selected as the depot of a 3PL company, and the vehicle capacity is increased up to $Q*(1+\beta)$ where β is randomly chosen from the interval $[0, 0.8]$. Then, the efficient variable neighborhood search (VNS) framework proposed by Wei et al. (2014) is used to solve the updated instance. Each route in the final solution can be chosen as a bid and the corresponding route cost is set as the charge. A route can be chosen if it satisfies the constraint that every customer cannot be contained in more than a third of the total bids. This process is repeated until a given number of bids are generated.

We only generate the bids for the type-1 instances. Then, every type-2 instance shares the same bids than the corresponding type-1 instance. For example, instances $r103$ and $r203$ have the same bids. Moreover, the total number of bids increases with the size of an instance, e.g., the number of bids is between $[6,11]$, $[8,21]$, and $[24,33]$ for the 25-, 50-, and 100-customer instances, respectively. In addition, the number of available vehicles is set to $BNV/2$ where BNV is the minimum number of vehicles needed to serve all customers, as reported in the literature. The vehicle capacity and the customer information remain the same. These instances are marked as group-1 instances. To study the effect of the number of vehicles, we also test the group-2 instances where the number of available vehicles is 25, the same as in the original VRPTW instances. Obviously, the group-1 instances force the bids to be selected in the solutions, while the group-2 instances have enough vehicles.

Finally, the travel cost and travel time of each arc are the Euclidian distance between the corresponding nodes. These values are not truncated in our implementation.

5.1.2. Instances Derived from Real-World Data The real-world data comes from a branch of the CNPC based in Dalian, a city located in north China. The data consists of one warehouse and 100 oil stations, including the locations of the warehouse and the oil stations, the demand of each oil station, and the vehicle capacity used in the oil distribution. However, some necessary information for the VRPTWCA does not exist in the data, such as the time window of the warehouse, the service time windows of oil stations and the set of bids submitted by the 3PL providers. Thus, the missing information is generated manually. For the warehouse, we generate a time window with length equal to eight hours. For the service time windows of the oil stations, we generate two time windows which correspond to the working times in the morning and the afternoon, respectively, and assign one of them to each oil station randomly. To generate the set of bids, the method described in Section 5.1.1 is used. The travel distances between the warehouse and the oil stations are achieved using the Baidu Map API (`http://lbsyun.baidu.com/index.php?title=jspopular`). Similar to the instances derived from the Solomon benchmark instances, we generate one real-world instance with a limited number of vehicles and another with a sufficient number of vehicles. In addition, because the time windows of oil stations may be too strict, we also generate two instances without any time windows for the oil stations. Therefore, we generate a total of four real-world instances, which are the same except for the number of available vehicles and the service time windows of oil stations.

5.2. Comparison of Inequalities

To assess the impact of different inequalities, we performed a series of experiments based on the instances derived from the Solomon benchmark to obtain the lower bound at the root node using the branch-and-price-and-cut (BPC) algorithm with a single family of inequalities. The average results of each class are summarized in Tables 1 and 2, where the first three columns give the information about the instances. Column UB is the best-known solution obtained by the BPC algorithm or the branch-and-bound (BB) algorithm in the experiments, while LB is the lower bound obtained without any inequalities. The next columns present the results of the proposed algorithm with only the capacity inequalities (CC), the strengthened capacity inequalities (SC), the k-path inequalities (KP), the bid partitioning inequalities (BP), the clique inequalities (Cli), and the subset-row inequalities (SR), respectively. The final column gives the results of the algorithm using all the inequalities simultaneously. In addition, the last six rows show the percentage of gap closed by the corresponding inequalities, i.e., $(CUT-LB)/(UB-LB)*100$, where CUT is the lower bound for the corresponding inequalities. Note that after the inequalities are introduced, not all the instances can be solved to optimality at the root node within the given time limit. So only those instances that can be solved to optimality in all settings are used for comparison.

From Tables 1 and 2, we can observe some interesting results. First, using all inequalities together can close the optimal gaps for certain classes of instances. Second, compared to the results obtained with a single family of inequalities, using them together produces better results on most of the instances. However, on some classes of instances, the subset-row inequalities (SR) alone are sufficiently powerful to close the gaps to optimality. Third, the strengthened capacity inequalities (SC) can indeed get tighter lower bounds than the capacity inequalities (CC) on all the instances. Fourth, the bid partitioning inequalities (BP) and the clique inequalities (Cli) turn out to be quite effective even if they are simple. Fifth, for the type-2 instances, no valid capacity (CC, SC) and k-path (KP) inequalities are obtained, which concurs with the observations on the VRPTW instances reported in the literature (Desaulniers et al. 2008). Finally, the lower bounds obtained by the BPC algorithm at the root nodes are quite close to the optimal solutions.

5.3. Integer Solutions

In this section, we apply the BPC algorithm and the BB algorithm to solve the instances to optimality. The BPC algorithm uses all the inequalities introduced in Section 2.3 except the capacity inequalities since they are dominated by strengthened capacity inequalities.

Table 3 shows the number of Solomon instances solved by the two algorithms, where column $Inst$ is the total number of instances and column Tot is the number of instances solved to optimality by either the BPC algorithm or the BB algorithm. Some interesting results are observed. First, out of the 168 instances of each group, 137 group-1 instances and 156 group-2 instances are optimally solved by the BPC algorithm, while the BB algorithm optimally solves 135 and 150 instances, respectively. In this regard, the BPC algorithm performs slightly better than the BB algorithm.

Table 1 The effects of different inequalities for the BPC algorithm on the Solomon Group-1 instances

Scale	Type	Class	*UB*	*LB*	*CC*	*SC*	*KP*	*BP*	*Cli*	*SR*	*All*
25	1	c	179.47	174.14	174.16	174.64	174.14	176.45	176.21	179.47	179.47
		r	470.50	463.37	463.37	463.37	464.35	464.35	464.14	469.73	469.73
		rc	307.66	281.99	281.99	284.46	282.56	287.93	287.02	307.66	307.66
	2	c	176.39	172.90	172.90	172.90	172.90	175.51	175.54	176.39	176.39
		r	397.40	393.03	393.03	393.03	393.03	393.62	393.62	397.21	397.21
		rc	277.69	261.98	261.98	261.98	261.98	265.35	266.64	277.69	277.69
50	1	c	345.50	334.45	335.16	339.92	334.45	334.84	336.15	343.53	343.82
		r	739.48	725.33	725.33	725.33	725.62	727.02	727.05	738.93	738.99
		rc	561.51	521.70	521.70	523.18	534.15	525.36	529.63	556.54	556.54
	2	c	315.53	313.11	313.11	313.11	313.11	314.25	314.41	315.53	315.53
		r	686.74	674.94	674.94	674.94	674.94	674.98	675.25	686.74	686.74
		rc	512.80	495.23	495.23	495.23	495.23	498.90	502.30	512.80	512.80
100	1	c	767.97	753.83	754.87	756.46	753.93	754.58	756.39	764.80	765.03
		r	1130.06	1113.47	1113.47	1113.50	1113.88	1114.35	1115.26	1126.58	1126.68
		rc	1192.43	1163.04	1166.63	1166.71	1165.85	1164.27	1166.29	1187.74	1188.04
	2	c	671.43	652.70	652.70	652.70	652.70	652.70	655.41	669.44	669.43
Imp(%)	1	c			5.84	28.20	0.32	11.31	20.77	83.16	84.85
		r			0.00	0.07	4.45	9.37	11.32	87.35	87.77
		rc			3.79	8.02	16.69	11.42	17.08	89.81	90.14
	2	c			0.00	0.00	0.00	15.21	26.97	91.92	91.87
		r			0.00	0.00	0.00	3.86	5.52	98.83	98.83
		rc			0.00	0.00	0.00	21.13	35.21	100.00	100.00

Table 2 The effects of different inequalities for the BPC algorithm on the Solomon Group-2 instances

Scale	Type	Class	*UB*	*LB*	*CC*	*SC*	*KP*	*BP*	*Cli*	*SR*	*All*
25	1	c	178.04	174.02	174.02	174.45	174.02	176.29	176.03	178.04	178.04
		r	454.54	450.84	450.84	450.84	451.68	451.04	451.20	454.46	454.46
		rc	294.33	277.50	277.50	283.17	278.07	281.75	282.53	294.33	294.33
	2	c	176.39	172.82	172.82	172.82	172.82	175.43	175.45	176.39	176.39
		r	382.52	380.06	380.06	380.06	380.06	380.06	380.06	382.52	382.52
		rc	275.55	261.06	261.06	261.06	261.06	264.31	265.70	275.55	275.55
50	1	c	339.61	331.26	332.18	336.82	331.26	331.65	332.88	339.29	339.51
		r	719.43	710.44	710.48	710.48	711.01	711.34	711.41	719.20	719.20
		rc	554.49	519.25	519.25	520.74	530.50	521.75	527.02	554.49	554.49
	2	c	312.71	310.67	310.67	310.67	310.67	311.80	311.96	312.71	312.71
		r	623.56	617.67	617.67	617.67	617.67	617.67	617.69	623.56	623.56
		rc	509.69	492.67	492.67	492.67	492.67	494.44	498.20	509.69	509.69
100	1	c	740.19	731.81	732.88	734.18	731.81	732.37	733.36	739.30	739.58
		r	1108.99	1095.36	1095.38	1095.38	1095.55	1095.72	1096.11	1107.58	1107.62
		rc	1158.40	1136.72	1141.98	1142.05	1137.35	1136.96	1137.60	1157.13	1157.13
	2	c	584.22	584.10	584.10	584.10	584.10	584.10	584.10	584.22	584.22
		r	998.34	991.05	991.05	991.05	991.05	991.05	991.05	997.07	997.07
		rc	1024.15	1013.96	1013.96	1013.96	1013.96	1013.97	1014.46	1024.15	1024.14
Imp(%)	1	c			9.60	40.28	0.00	15.52	24.94	94.18	96.54
		r			0.22	0.22	6.06	5.53	7.88	93.47	93.60
		rc			7.13	16.93	16.88	9.48	18.55	98.28	98.29
	2	c			0.00	0.00	0.00	65.32	68.51	100.00	100.00
		r			0.00	0.00	0.00	0.00	0.16	91.85	91.85
		rc			0.00	0.00	0.00	12.06	25.58	100.00	99.99

However, the BB algorithm can optimally solve some instances which cannot be solved by the BPC algorithm. In these instances, the column-and-cut generation procedure in the BPC algorithm usually fails to solve the LP relaxed set-partitioning model at the root node in the given time limit. Second, on the whole, the proposed algorithms obtain optimal solutions for 140 and 158 instances of group-1 and group-2, respectively. Obviously, more group-2 instances are solved to optimality. Third, more type-1 instances are solved, especially for the large-scale instances, because the type-2 instances have wider time windows, and the label-setting algorithm needs to explore many more states and hence consumes more time. Finally, class r contains the most difficult instances and class c are the easiest.

Table 3 **Number of instances solved by the BPC and BB algorithms**

				Group-1			Group-2		
Scale	Type	Class	Inst	BPC	BB	Tot	BPC	BB	Tot
25	1	c	9	9	9	9	9	9	9
		r	12	12	12	12	12	12	12
		rc	8	8	8	8	8	8	8
	2	c	8	8	8	8	8	8	8
		r	11	11	11	11	11	11	11
		rc	8	8	8	8	8	8	8
50	1	c	9	9	9	9	9	9	9
		r	12	12	12	12	12	12	12
		rc	8	8	8	8	8	8	8
	2	c	8	8	8	8	8	8	8
		r	11	6	7	7	10	9	10
		rc	8	7	8	8	7	8	8
100	1	c	9	9	9	9	9	9	9
		r	12	11	12	12	11	12	12
		rc	8	8	6	8	8	8	8
	2	c	8	3	0	3	8	5	8
		r	11	0	0	0	4	3	4
		rc	8	0	0	0	6	3	6
	Tot		168	137	135	140	156	150	158

Tables 4 and 5 compare the average results on each class of the Solomon instances for the lower bound at the root node (LB), the number of nodes in the branching tree ($Nodes$) and the total computational time to solve the instances (*Tot-Time*). To be more specific, columns *BPC-LP* and *BPC-LPC* report the lower bounds obtained by the BPC algorithm without and with the inequalities at the root node, respectively. Note that only instances solved by both the BPC algorithm and the BB algorithm are taken into account, and column $Inst*$ gives the corresponding number. According to the results, the lower bounds (*BPC-LP*) obtained by the BPC algorithm without inequalities are much better than the lower bounds (*LB-BB*) obtained by the BB algorithm. Moreover, the lower bounds (*BPC-LPC*) obtained by the BPC algorithm are quite close to the optimal solutions. Furthermore, the BB algorithm needs to explore many more nodes (more than 600 on average) to obtain an integer solution, while the BPC algorithm only explores about two nodes on average. Considering the total computational time, the BPC algorithm requires less time to solve the instances than the BB algorithm on average. In general, both the BPC and BB algorithms require less time for the type-1 instances, but more time to solve the type-2 instances, because the label-setting procedure in the BPC algorithm takes more time to solve the type-2 instances with wider time windows. Moreover, comparing the UB in Tables 4 and 5, we find that lower costs are incurred on the group-2 instances, because more vehicles provide more choices.

Integer results on the real-world instances are summarized in Table 6. Column *Veh* gives the number of available vehicles in an instance and column TW indicates whether an instance involves service time windows for the oil stations. Columns BPC presents the results obtained by the branch-and-price-and-cut (BPC) algorithm, including the lower bound at the root node obtained with and without inequalities (LPC and LP, respectively), the number of nodes in the branching tree ($Nodes$), the number of inequalities added to the tree (SC, Cli, SR), the total time to solve the instance ($TotT$), and the time to identify the inequalities ($SepT$). Because the BPC algorithm cannot find any violated k-path inequalities and the bid partitioning inequalities, they are not presented in the table. Columns BB present the results obtained by the branch-and-bound (BB) algorithm, including the lower bound (LB) of the instance, the number of nodes in the branching tree ($Nodes$), the time to achieve the lower bound (LBT) and the total time to solve the instance ($TotT$). From the table, we can see that all the four instances can be solved to optimality by both algorithms within four hours of computational time. The computational time on the instances with time windows is even less than three minutes. Meanwhile, the BPC algorithm and the BB algorithm are comparable on the instances with a limited number of vehicles, but the BPC algorithm outperforms the BB algorithm on the instances with a sufficient number of vehicles. In addition, except for the strengthened capacity inequalities and the subset-row inequalities, the other inequalities have little influence on the set-partitioning model. Here, many violated strengthened capacities are found mainly because the capacity constraints in these instances are quite tight.

Table 4 Comparison of the results of the BPC and BB algorithms on the Solomon Group-1 instances

					LB			$Nodes$		$Tot-Time$	
Scale	Type	Class	$Inst*$	UB	$BPC-LP$	$BPC-LPC$	BB	BPC	BB	BPC	BB
25	1	c	9	179.47	174.14	179.47	133.65	1.00	17.78	4.00	3.48
		r	12	470.50	463.37	469.73	447.80	1.67	9.92	1.59	2.53
		rc	8	307.66	281.99	307.66	209.07	1.00	19.50	2.18	2.46
	2	c	8	176.39	172.90	176.39	126.75	1.00	14.88	2.65	2.98
		r	11	397.40	393.03	397.21	377.61	1.18	10.45	248.45	12.45
		rc	8	277.69	261.98	277.69	194.57	1.00	18.63	18.47	6.58
50	1	c	9	345.50	334.45	343.82	291.01	2.33	83.22	71.24	47.14
		r	12	739.48	725.33	738.99	689.60	1.50	76.92	84.87	28.47
		rc	8	561.51	521.70	556.54	464.73	1.75	200.25	11.06	20.58
	2	c	8	315.53	313.11	315.53	269.61	1.00	81.63	41.87	48.52
		r	6	686.74	674.94	686.74	635.14	1.00	226.17	1023.54	381.38
		rc	7	512.80	495.23	512.80	368.45	1.00	378.57	72.38	1898.09
100	1	c	9	767.97	753.83	765.03	682.39	3.67	4881.56	128.69	1700.37
		r	11	1132.06	1118.00	1129.10	1047.45	4.27	4132.00	1852.03	3720.15
		rc	6	1216.32	1189.83	1211.69	1115.67	8.00	2354.83	143.53	1577.82
Avg				539.13	524.92	537.89	470.23	2.09	833.75	247.10	630.20

6. Conclusions

In this paper, we introduce a new extension of the classic VRPTW where some customers can be outsourced to the 3PL providers through combinatorial auction. This new extension is called

Table 5 Comparison of the results of the BPC and BB algorithms for the Solomon Group-2 instances

Scale	Type	Class	$Inst*$	UB	LB			$Nodes$		$Tot-Time$	
					$BPC-LP$	$BPC-LPC$	BB	BPC	BB	BPC	BB
25	1	c	9	178.04	174.02	178.04	133.65	1.00	17.89	1.42	1.99
		r	12	454.54	450.84	454.46	441.28	1.67	7.50	1.05	1.76
		rc	8	294.33	277.50	294.33	207.34	1.00	19.88	1.44	1.57
	2	c	8	176.39	172.82	176.39	126.75	1.00	14.88	4.09	2.65
		r	11	382.52	380.06	382.52	377.61	1.00	3.36	2.86	4.31
		rc	8	275.55	261.06	275.55	194.57	1.00	18.50	12.19	5.61
50	1	c	9	339.61	331.26	339.51	291.01	1.44	70.56	12.47	18.22
		r	12	719.43	710.44	719.20	685.54	1.67	78.58	10.93	23.40
		rc	8	554.49	519.25	554.49	464.73	1.00	175.38	7.45	18.09
	2	c	8	312.71	310.67	312.71	269.61	1.00	72.63	15.28	38.73
		r	9	636.26	630.09	636.26	609.51	1.00	74.33	95.45	265.12
		rc	7	509.69	492.67	509.69	368.45	1.00	364.00	103.21	942.94
100	1	c	9	740.19	731.81	739.58	679.18	1.67	1981.44	81.47	1289.03
		r	11	1127.46	1114.69	1126.17	1059.78	11.18	5083.73	1630.19	1127.46
		rc	8	1158.40	1136.72	1157.13	1087.52	4.25	1621.88	699.38	1306.70
	2	c	5	585.10	585.10	585.10	542.05	1.00	375.20	2894.54	6160.66
		r	3	1016.08	1009.41	1014.38	995.26	5.00	360.00	1770.40	9418.91
		rc	3	1034.58	1025.70	1034.58	960.18	1.00	899.33	1078.37	10461.77
Avg				583.08	573.01	582.78	527.44	2.10	624.39	467.90	1727.16

Table 6 Detailed results for the real-world instances

Inst.	Veh	TW	UB	BPC								BB			
				LP	LPC	$Nodes$	SC	Cli	SR	$SepT$	$TotT$	LB	$Nodes$	LBT	$TotT$
1	10	Yes	713.11	710.25	713.11	1	114	0	20	8.5	95.0	697.94	1	40.89	130.92
2	10	No	704.11	700.53	704.08	2	168	0	30	14.2	146.7	684.13	3	58.57	139.62
3	25	Yes	647.04	642.93	645.89	29	164	0	180	55.8	197.28	633.03	62	41.39	706.67
4	25	No	636.84	631.92	634.78	191	586	1	239	287.1	2312.93	615.86	166	59.68	9014.56

the vehicle routing problem with time windows and combinatorial auction (VRPTWCA). We formulate two mathematical models for the VRPTWCA: a compact arc-flow model and a strong set-partitioning model. To further strengthen the set-partitioning model, we propose six families of valid inequalities derived from the properties of the problem. Based on the set-partitioning model and the valid inequalities, we propose a branch-and-price-and-cut algorithm and a branch-and-bound algorithm to solve the problem exactly. To test the proposed algorithms, we generate test instances according to the classic Solomon benchmark instances (Solomon 1987) for the VRPTW and real-world data of the CNPC. The computational results show that the valid inequalities can substantially improve the lower bounds yielded by the LP relaxations of the set-partitioning model. In addition, both algorithms can optimally solve Solomon instances with up to 100 customers and all real-world instances in four hours of computational time.

In this paper, we focus on exact algorithms for the VRPTWCA. According to the computational results, the exact algorithms take a large amount of time to solve large instances to optimality. So, in practice, heuristics are more suitable because heuristics can usually solve large instances quickly. Therefore, our future work on the VRPTWCA will focus on the design of efficient heuristics. The computational results presented in this paper can serve as reference for our future research.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (Grant Nos. 71501091, 71390335, 71501075, 71531009, 71571077, 71401059), Guangdong Natural Science Funds for Distinguished Young Scholar (Grant No. 2015A030306007), NRF Singapore [Grant NRF-RSS2016-004], MOE-AcRF-Tier 1 [Grants R-266-000-096-133, R-266-000-096-731, and R-266-000-100-646], and the Fundamental Research Funds for the Central Universities.

References

Archetti, C., M. Bouchard, G. Desaulniers. 2011. Enhanced branch and price and cut for vehicle routing with split deliveries and time windows. *Transportation Science* **45**(3) 285–298.

Baldacci, R., N. Christofides, A. Mingozzi. 2008. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming* **115**(2) 351–385.

Baldacci, R., A. Mingozzi, R. Roberti. 2011. New route relaxation and pricing strategies for the vehicle routing problem. *Operations research* **59**(5) 1269–1283.

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

Bolduc, M.-C., J. Renaud, F. Boctor. 2007. A heuristic for the routing and carrier selection problem. *European Journal of Operational Research* **183**(2) 926–932.

Bolduc, M.-C., J. Renaud, F. Boctor, G. Laporte. 2008. A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. *Journal of the Operational Research Society* **59**(6) 776–787.

Bräysy, O., M. Gendreau. 2005. Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science* **39**(1) 119–139.

Christofides, N., A. Mingozzi, P. Toth. 1981. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical programming* **20**(1) 255–282.

Chu, C.-W. 2005. A heuristic algorithm for the truckload and less-than-truckload problem. *European Journal of Operational Research* **165**(3) 657–667.

Clarke, G., J. W. Wright. 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research* **12**(4) 568–581.

Cordeau, J.-F. 2006. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research* **54**(3) 573–586.

Côté, J.-F., J.-Y. Potvin. 2009. A tabu search heuristic for the vehicle routing problem with private fleet and common carrier. *European Journal of Operational Research* **198**(2) 464–469.

Desaulniers, G. 2010. Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations Research* **58**(1) 179–192.

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

Desrosiers, J., M. E. Lübbecke. 2005. A primer in column generation. Guy Desaulniers, Jacques Desrosiers, Marius M. Solomon, eds., *Column Generation*. Springer US, 1–32.

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

Hifi, M. 1997. A genetic algorithm-based heuristic for solving the weighted maximum independent set and some equivalent problems. *Journal of the Operational Research Society* **48**(6) 612–622.

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

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

Li, J.-Q. 2009. A computational study of bi-directional dynamic programming for the traveling salesman problem with time windows. *Working paper, University of California, Berkeley* .

Lim, A., Z. Zhang, H. Qin. 2017. Pickup and delivery service with manpower planning in Hong Kong public hospitals. *Transportation Science* **51**(2) 688–705.

Luo, Z., H. Qin, A. Lim. 2014. Branch-and-price-and-cut for the multiple traveling repairman problem with distance constraints. *European Journal of Operational Research* **234**(1) 49–60.

Martinelli, R., D. Pecin, M. Poggi. 2014. Efficient elementary and restricted non-elementary route pricing. *European Journal of Operational Research* **239**(1) 102–111.

Naddef, D., G. Rinaldi. 2001. Branch-and-cut algorithms for the capacitated vrp. *The vehicle routing problem*. Society for Industrial and Applied Mathematics, 53–84.

Östergård, P. R. 2002. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics* **120**(1-3) 197–207.

Potvin, J.-Y., M.-A. Naud. 2011. Tabu search with ejection chains for the vehicle routing problem with private fleet and common carrier. *Journal of the Operational Research Society* **62** 326–336.

Potvin, J.-Y., J.-M. Rousseau. 1993. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research* **66**(3) 331–340.

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

Righini, G., M. Salani. 2008. New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks* **51**(3) 155–170.

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

Stenger, A., D. Vigo, S. Enz, M. Schwind. 2013. An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transportation Science* **47**(1) 64–80.

Vidal, T., T. G. Crainic, M. Gendreau, C. Prins. 2014. A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research* **234**(3) 658–673.

Vidal, T., N. Maculan, L. S. Ochi, P. H. Vaz Penna. 2016. Large neighborhoods with implicit customer selection for vehicle routing problems with profits. *Transportation Science* **50**(2) 720–734.

Wei, L., Z. Zhang, A. Lim. 2014. An adaptive variable neighborhood search for a heterogeneous fleet vehicle routing problem with three-dimensional loading constraints. *Computational Intelligence Magazine, IEEE* **9**(4) 18–30.

Zhang, Z., M. Liu, A. Lim. 2015. A memetic algorithm for the patient transportation problem. *Omega* **54** 60–71.

Appendix A: Detailed Lower Bounds of Different Inequalities

Tables I-XII give the detailed lower bounds obtained by the branch-and-price-and-cut (BPC) algorithm with different inequalities. Column UB lists the best-known solutions (non-optimal solutions are marked with *) obtained by the BPC algorithm or the branch-and-bound (BB) algorithm in the experiments, while LB is the lower bound obtained without any inequalities. The next columns present the results of the proposed algorithm with the capacity inequalities (CC), the strengthened capacity inequalities (SC), the k-path inequalities (KP), the bid partitioning inequalities (BP), the clique inequalities (Cli), and the subset-row inequalities (SR), respectively. A dash (-) indicates that the corresponding value cannot be obtained within the specified computational time. Moreover, the bold numbers indicate that the lower bound is an optimal integer solution.

Table I Lower bounds of BPC on the Solomon Type-1 instances with 25 customers in Group-1

Inst.	UB	LB	CC	SC	KP	BP	Cli	SR	All
c101	138.10	131.99	131.99	131.99	131.99	134.95	131.99	**138.10**	**138.10**
c102	195.69	191.79	191.79	191.79	191.79	191.79	192.05	**195.69**	**195.69**
c103	177.91	174.77	174.77	174.77	174.77	174.77	174.77	**177.91**	**177.91**
c104	192.93	187.46	187.70	187.70	187.46	187.46	187.46	**192.93**	**192.93**
c105	182.99	181.40	181.40	182.06	181.40	182.32	182.93	**182.99**	**182.99**
c106	190.21	169.93	169.93	169.93	169.93	184.83	184.83	**190.21**	**190.21**
c107	194.27	187.81	187.81	191.49	187.81	188.73	188.73	**194.27**	**194.27**
c108	176.63	175.58	175.58	175.58	175.58	**176.63**	**176.63**	**176.63**	**176.63**
c109	166.54	166.49	166.49	166.49	166.49	**166.54**	**166.54**	**166.54**	**166.54**
r101	565.28	557.61	557.61	557.61	**565.28**	557.61	557.61	**565.28**	**565.28**
r102	558.20	551.82	551.82	551.82	552.04	551.82	551.82	**558.20**	**558.20**
r103	454.09	453.68	453.68	453.68	453.68	453.68	453.68	453.68	453.68
r104	422.98	421.61	421.61	421.61	421.61	421.61	421.61	**422.98**	**422.98**
r105	502.55	499.54	499.54	499.54	499.54	500.35	499.94	**502.55**	**502.55**
r106	472.12	**472.12**	**472.12**	**472.12**	**472.12**	**472.12**	**472.12**	472.12	472.12
r107	477.28	475.03	475.03	475.03	476.67	475.89	475.89	**477.28**	**477.28**
r108	410.56	401.40	401.40	401.40	401.64	401.40	401.40	406.09	406.09
r109	442.40	437.67	437.67	437.67	439.73	437.67	437.67	**442.40**	**442.40**
r110	460.31	439.41	439.41	439.41	439.41	445.78	445.44	457.83	457.83
r111	430.35	421.46	421.46	421.46	421.46	421.46	421.46	428.43	428.43
r112	449.91	429.06	429.06	429.06	429.06	432.75	431.09	**449.91**	**449.91**
rc101	350.66	322.67	322.67	324.19	327.23	325.31	339.43	**350.66**	**350.66**
rc102	259.61	255.04	255.04	256.77	255.04	256.77	255.04	**259.61**	**259.61**
rc103	324.15	323.93	323.93	323.93	323.93	**324.15**	**324.15**	**324.15**	**324.15**
rc104	239.85	227.45	227.45	231.81	227.45	232.85	231.33	**239.85**	**239.85**
rc105	455.00	357.83	357.83	357.83	357.83	381.21	363.08	**455.00**	**455.00**
rc106	303.92	285.62	285.62	285.62	285.62	291.23	292.55	**303.92**	**303.92**
rc107	236.47	200.84	200.84	212.97	200.84	204.30	204.45	**236.47**	**236.47**
rc108	291.59	282.52	282.52	282.52	282.52	287.59	286.09	**291.59**	**291.59**

Table II Lower bounds of BPC on the Solomon Type-2 instances with 25 customers in Group-1

Inst.	*UB*	*LB*	*CC*	*SC*	*KP*	*BP*	*Cli*	*SR*	*All*
c201	148.32	139.32	139.32	139.32	139.32	148.15	**148.32**	**148.32**	**148.32**
c202	200.73	197.01	197.01	197.01	197.01	**200.73**	**200.73**	**200.73**	**200.73**
c203	177.20	**177.20**	**177.20**	**177.20**	**177.20**	**177.20**	**177.20**	**177.20**	**177.20**
c204	151.27	148.30	148.30	148.30	148.30	148.30	148.30	**151.27**	**151.27**
c205	151.48	**151.48**	**151.48**	**151.48**	**151.48**	**151.48**	**151.48**	**151.48**	**151.48**
c206	207.24	203.20	203.20	203.20	203.20	203.40	203.40	**207.24**	**207.24**
c207	179.68	**179.68**	**179.68**	**179.68**	**179.68**	**179.68**	**179.68**	**179.68**	**179.68**
c208	195.17	186.99	186.99	186.99	186.99	**195.17**	**195.17**	**195.17**	**195.17**
r201	485.76	479.06	479.06	479.06	479.06	483.88	484.03	**485.76**	**485.76**
r202	447.46	440.47	440.47	440.47	440.47	442.13	441.96	445.38	445.38
r203	400.40	**400.40**	**400.40**	**400.40**	**400.40**	**400.40**	**400.40**	**400.40**	**400.40**
r204	385.54	379.78	379.78	379.78	379.78	379.78	379.78	**385.54**	**385.54**
r205	405.98	401.83	401.83	401.83	401.83	401.83	401.83	**405.98**	**405.98**
r206	378.18	**378.18**	**378.18**	**378.18**	**378.18**	**378.18**	**378.18**	**378.18**	**378.18**
r207	362.79	361.30	361.30	361.30	361.30	361.30	361.30	**362.79**	**362.79**
r208	329.33	**329.33**	**329.33**	**329.33**	**329.33**	**329.33**	**329.33**	**329.33**	**329.33**
r209	403.65	385.92	385.92	385.92	385.92	385.92	385.92	**403.65**	**403.65**
r210	410.60	405.70	405.70	405.70	405.70	405.70	405.70	**410.60**	**410.60**
r211	361.69	361.38	361.38	361.38	361.38	361.38	361.38	**361.69**	**361.69**
rc201	334.04	313.44	313.44	313.44	313.44	316.79	321.93	**334.04**	**334.04**
rc202	245.26	**245.26**	**245.26**	**245.26**	**245.26**	**245.26**	**245.26**	**245.26**	**245.26**
rc203	307.79	307.55	307.55	307.55	307.55	307.55	307.55	**307.79**	**307.79**
rc204	228.85	217.82	217.82	217.82	217.82	220.65	219.41	**228.85**	**228.85**
rc205	345.39	309.09	309.09	309.09	309.09	316.70	314.54	**345.39**	**345.39**
rc206	286.60	261.98	261.98	261.98	261.98	270.12	275.43	**286.60**	**286.60**
rc207	222.39	193.16	193.16	193.16	193.16	195.94	200.08	**222.39**	**222.39**
rc208	251.19	247.57	247.57	247.57	247.57	249.78	248.90	**251.19**	**251.19**

Table III Lower bounds of BPC on the Solomon Type-1 instances with 25 customers in Group-2

Inst.	*UB*	*LB*	*CC*	*SC*	*KP*	*BP*	*Cli*	*SR*	*All*
c101	138.10	131.99	131.99	131.99	131.99	134.95	131.99	**138.10**	**138.10**
c102	190.74	**190.74**	**190.74**	**190.74**	**190.74**	**190.74**	**190.74**	**190.74**	**190.74**
c103	177.91	174.77	174.77	174.77	174.77	174.77	174.77	**177.91**	**177.91**
c104	187.45	**187.45**	**187.45**	**187.45**	**187.45**	**187.45**	**187.45**	**187.45**	**187.45**
c105	182.99	181.40	181.40	182.06	181.40	182.32	182.93	**182.99**	**182.99**
c106	190.21	169.93	169.93	169.93	169.93	184.83	184.83	**190.21**	**190.21**
c107	191.81	187.81	187.81	191.03	187.81	188.35	188.35	**191.81**	**191.81**
c108	176.63	175.58	175.58	175.58	175.58	**176.63**	**176.63**	**176.63**	**176.63**
c109	166.54	166.49	166.49	166.49	166.49	**166.54**	**166.54**	**166.54**	**166.54**
r101	565.15	557.53	557.53	557.53	**565.15**	557.53	557.53	**565.15**	**565.15**
r102	538.51	537.93	537.93	537.93	537.93	537.93	537.93	**538.51**	**538.51**
r103	453.27	**453.27**	**453.27**	**453.27**	**453.27**	**453.27**	**453.27**	**453.27**	**453.27**
r104	409.30	**409.30**	**409.30**	**409.30**	**409.30**	**409.30**	**409.30**	**409.30**	**409.30**
r105	501.72	497.41	497.41	497.41	497.41	497.41	499.28	**501.72**	**501.72**
r106	453.52	451.80	451.80	451.80	451.80	451.80	451.80	**453.52**	**453.52**
r107	425.27	**425.27**	**425.27**	**425.27**	**425.27**	**425.27**	**425.27**	**425.27**	**425.27**
r108	398.29	397.24	397.24	397.24	397.24	397.24	397.24	**398.29**	**398.29**
r109	442.40	437.67	437.67	437.67	438.88	437.67	437.67	**442.40**	**442.40**
r110	445.18	433.01	433.01	433.01	433.01	435.48	435.48	444.19	444.19
r111	427.77	421.46	421.46	421.46	421.46	421.46	421.46	**427.77**	**427.77**
r112	394.10	388.16	388.16	388.16	389.38	388.16	388.16	**394.10**	**394.10**
rc101	350.66	322.67	322.67	324.19	327.23	325.31	339.43	**350.66**	**350.66**
rc102	259.61	255.04	255.04	256.67	255.04	256.77	255.04	**259.61**	**259.61**
rc103	324.15	323.93	323.93	323.93	323.93	**324.15**	**324.15**	**324.15**	**324.15**
rc104	239.85	227.45	227.45	231.81	227.45	232.85	231.33	**239.85**	**239.85**
rc105	348.42	321.94	321.94	347.61	321.94	331.81	327.20	**348.42**	**348.42**
rc106	303.92	285.62	285.62	285.62	285.62	291.23	292.55	**303.92**	**303.92**
rc107	236.47	200.84	200.84	212.97	200.84	204.30	204.45	**236.47**	**236.47**
rc108	291.59	282.52	282.52	282.52	282.52	287.59	286.09	**291.59**	**291.59**

Table IV Lower bounds of BPC on the Solomon Type-2 instances with 25 customers in Group-2

Inst.	*UB*	*LB*	*CC*	*SC*	*KP*	*BP*	*Cli*	*SR*	*All*
c201	148.32	139.32	139.32	139.32	139.32	148.15	**148.32**	**148.32**	**148.32**
c202	200.73	197.01	197.01	197.01	197.01	**200.73**	**200.73**	**200.73**	**200.73**
c203	177.20	**177.20**	**177.20**	**177.20**	**177.20**	**177.20**	**177.20**	**177.20**	**177.20**
c204	151.27	148.30	148.30	148.30	148.30	148.30	148.30	**151.27**	**151.27**
c205	151.48	**151.48**	**151.48**	**151.48**	**151.48**	**151.48**	**151.48**	**151.48**	**151.48**
c206	207.24	202.57	202.57	202.57	202.57	202.69	202.69	**207.24**	**207.24**
c207	179.68	**179.68**	**179.68**	**179.68**	**179.68**	**179.68**	**179.68**	**179.68**	**179.68**
c208	195.17	186.99	186.99	186.99	186.99	**195.17**	**195.17**	**195.17**	**195.17**
r201	457.56	454.82	454.82	454.82	454.82	454.82	454.82	**457.56**	**457.56**
r202	411.49	**411.49**	**411.49**	**411.49**	**411.49**	**411.49**	**411.49**	**411.49**	**411.49**
r203	392.33	**392.33**	**392.33**	**392.33**	**392.33**	**392.33**	**392.33**	**392.33**	**392.33**
r204	355.89	352.92	352.92	352.92	352.92	352.92	352.92	**355.89**	**355.89**
r205	394.06	391.67	391.67	391.67	391.67	391.67	391.67	**394.06**	**394.06**
r206	375.48	374.62	374.62	374.62	374.62	374.62	374.62	**375.48**	**375.48**
r207	362.63	361.14	361.14	361.14	361.14	361.14	361.14	**362.63**	**362.63**
r208	329.33	**329.33**	**329.33**	**329.33**	**329.33**	**329.33**	**329.33**	**329.33**	**329.33**
r209	371.56	365.03	365.03	365.03	365.03	365.03	365.03	**371.56**	**371.56**
r210	405.48	405.05	405.05	405.05	405.05	405.05	405.05	**405.48**	**405.48**
r211	351.91	342.31	342.31	342.31	342.31	342.31	342.31	**351.91**	**351.91**
rc201	334.04	313.44	313.44	313.44	313.44	316.79	321.93	**334.04**	**334.04**
rc202	245.26	**245.26**	**245.26**	**245.26**	**245.26**	**245.26**	**245.26**	**245.26**	**245.26**
rc203	307.79	307.55	307.55	307.55	307.55	307.55	307.55	**307.79**	**307.79**
rc204	228.85	217.82	217.82	217.82	217.82	220.65	219.41	**228.85**	**228.85**
rc205	328.25	301.68	301.68	301.68	301.68	308.35	307.03	**328.25**	**328.25**
rc206	286.60	261.98	261.98	261.98	261.98	270.12	275.43	**286.60**	**286.60**
rc207	222.39	193.16	193.16	193.16	193.16	195.94	200.08	**222.39**	**222.39**
rc208	251.19	247.57	247.57	247.57	247.57	249.78	248.90	**251.19**	**251.19**

Table V Lower bounds of BPC on the Solomon Type-1 instances with 50 customers in Group-1

Inst.	*UB*	*LB*	*CC*	*SC*	*KP*	*BP*	*Cli*	*SR*	*All*
c101	356.67	332.47	332.47	346.32	332.47	332.47	334.45	355.61	355.61
c102	362.67	343.31	347.83	352.68	343.31	343.31	343.31	358.56	356.83
c103	344.16	324.24	325.45	330.19	324.24	325.71	324.35	335.27	339.59
c104	343.78	342.01	342.01	**343.78**	342.01	342.01	342.83	**343.78**	**343.78**
c105	334.80	334.07	334.07	334.07	334.07	334.07	334.07	**334.80**	**334.80**
c106	315.94	303.43	303.43	315.94	303.43	303.43	313.98	**315.94**	**315.94**
c107	373.95	369.84	369.84	370.28	369.84	369.84	370.02	370.28	370.28
c108	309.29	305.59	305.59	309.29	305.59	307.63	307.29	**309.29**	**309.29**
c109	368.28	355.06	355.78	356.75	355.06	355.06	355.06	**368.28**	**368.28**
r101	941.99	934.44	934.44	934.44	934.44	940.63	940.63	**941.99**	**941.99**
r102	800.74	798.88	798.88	798.88	798.88	798.88	798.88	**800.74**	**800.74**
r103	710.96	707.71	707.71	707.71	707.71	707.71	707.71	**710.96**	**710.96**
r104	726.12	696.35	696.35	696.35	696.35	696.35	696.35	723.09	723.73
r105	881.38	857.90	857.90	857.90	859.42	867.90	868.50	**881.38**	**881.38**
r106	718.37	700.76	700.76	700.76	700.76	700.76	700.76	716.05	716.05
r107	679.09	670.34	670.34	670.34	670.96	670.34	671.10	**679.09**	**679.09**
r108	659.58	639.13	639.13	639.13	640.44	639.13	639.13	**659.58**	**659.58**
r109	815.88	770.42	770.42	770.42	770.42	774.51	773.44	814.70	814.79
r110	672.50	672.47	672.47	672.47	672.47	672.47	**672.50**	**672.50**	**672.50**
r111	640.99	638.10	638.10	638.10	638.10	638.10	638.10	**640.99**	**640.99**
r112	626.10	617.51	617.51	617.51	617.51	617.51	617.51	**626.10**	**626.10**
rc101	632.55	623.69	623.69	623.69	**632.55**	**632.55**	**632.55**	**632.55**	**632.55**
rc102	701.64	615.32	615.32	621.28	638.64	616.57	624.12	661.88	661.88
rc103	532.70	491.75	491.75	491.75	491.75	493.76	500.44	**532.70**	**532.70**
rc104	515.11	485.53	485.53	485.53	485.53	486.83	488.22	**515.11**	**515.11**
rc105	629.58	577.32	577.32	577.32	608.99	584.48	595.46	**629.58**	**629.58**
rc106	484.59	422.20	422.20	422.20	457.83	430.62	435.39	**484.59**	**484.59**
rc107	501.20	470.26	470.26	475.81	470.26	470.44	472.75	**501.20**	**501.20**
rc108	494.73	487.50	487.50	487.83	487.67	487.64	488.08	**494.73**	**494.73**

Table VI Lower bounds of BPC on the Solomon Type-2 instances with 50 customers in Group-1

Inst.	*UB*	*LB*	*CC*	*SC*	*KP*	*BP*	*Cli*	*SR*	*All*
c201	384.41	381.37	381.37	381.37	381.37	381.37	381.42	**384.41**	**384.41**
c202	277.55	**277.55**	**277.55**	**277.55**	**277.55**	**277.55**	**277.55**	**277.55**	**277.55**
c203	277.94	**277.94**	**277.94**	**277.94**	**277.94**	**277.94**	**277.94**	**277.94**	**277.94**
c204	284.94	280.82	280.82	280.82	280.82	280.82	**284.94**	**284.94**	**284.94**
c205	345.61	339.06	339.06	339.06	339.06	**345.61**	341.24	**345.61**	**345.61**
c206	309.32	307.07	307.07	307.07	307.07	307.07	307.72	**309.32**	**309.32**
c207	344.53	343.64	343.64	343.64	343.64	343.64	**344.53**	**344.53**	**344.53**
c208	299.96	297.45	297.45	297.45	297.45	**299.96**	**299.96**	**299.96**	**299.96**
r201	796.23	793.64	793.64	793.64	793.64	793.64	795.24	**796.23**	**796.23**
r202	695.31	689.71	689.71	689.71	689.71	689.71	689.71	**695.31**	**695.31**
r203	619.52	607.27	607.27	607.27	607.27	607.27	607.27	**619.52**	**619.52**
r204	654.40*	-	-	-	-	-	-	-	-
r205	730.53	714.27	714.27	714.27	714.27	714.48	714.48	**730.53**	**730.53**
r206	641.11	627.11	627.11	627.11	627.11	627.11	627.49	-	-
r207	634.57*	-	-	-	-	-	-	-	-
r208	515.76*	-	-	-	-	-	-	-	-
r209	638.65	617.18	617.18	617.18	617.18	617.18	617.18	**638.65**	**638.65**
r210	640.18	627.58	627.58	627.58	627.58	627.58	627.60	**640.18**	**640.18**
r211	550.20*	540.69	540.69	540.69	540.69	540.69	540.69	-	-
rc201	627.76	613.52	613.52	613.52	613.52	626.77	627.38	**627.76**	**627.76**
rc202	569.44	550.68	550.68	550.68	550.68	550.68	561.72	**569.44**	**569.44**
rc203	482.32	467.98	467.98	467.98	467.98	467.98	473.02	**482.32**	**482.32**
rc204	434.06	417.30	417.30	417.30	417.30	419.15	419.79	-	-
rc205	575.01	553.43	553.43	553.43	553.43	557.39	562.17	**575.01**	**575.01**
rc206	446.07	410.38	410.38	410.38	410.38	418.64	420.60	**446.07**	**446.07**
rc207	461.03	445.62	445.62	445.62	445.62	445.81	446.17	**461.03**	**461.03**
rc208	428.00	425.02	425.02	425.02	425.02	425.02	425.02	**428.00**	**428.00**

Table VII Lower bounds of BPC on the Solomon Type-1 instances with 50 customers in Group-2

Inst.	*UB*	*LB*	*CC*	*SC*	*KP*	*BP*	*Cli*	*SR*	*All*
c101	355.61	332.47	332.47	346.32	332.47	332.47	334.45	354.67	354.67
c102	351.26	340.53	344.52	**351.26**	340.53	340.53	340.53	**351.26**	**351.26**
c103	337.82	324.23	325.41	330.20	324.23	325.71	324.35	335.87	**337.82**
c104	342.46	341.26	341.26	**342.46**	341.26	341.26	342.08	**342.46**	**342.46**
c105	334.80	334.07	334.07	334.07	334.07	334.07	334.07	**334.80**	**334.80**
c106	310.76	300.84	300.84	309.67	300.84	300.84	**310.76**	**310.76**	**310.76**
c107	353.90	**353.90**	**353.90**	**353.90**	**353.90**	**353.90**	**353.90**	**353.90**	**353.90**
c108	309.29	305.59	305.59	**309.29**	305.59	307.63	307.29	**309.29**	**309.29**
c109	360.62	348.46	351.57	354.20	348.46	348.46	348.46	**360.62**	**360.62**
r101	941.99	934.44	934.44	934.44	934.44	940.63	940.63	**941.99**	**941.99**
r102	800.74	798.88	798.88	798.88	798.88	798.88	798.88	**800.74**	**800.74**
r103	710.46	707.71	707.71	707.71	707.71	707.71	707.71	**710.46**	**710.46**
r104	631.32	621.49	621.94	621.94	624.20	621.49	621.49	**631.32**	**631.32**
r105	850.06	829.11	829.11	829.11	829.93	833.68	833.68	849.78	849.78
r106	716.57	700.76	700.76	700.76	700.76	700.76	700.76	716.05	716.05
r107	679.09	670.34	670.34	670.34	670.96	670.34	671.10	**679.09**	**679.09**
r108	592.05	585.11	585.11	585.11	587.79	585.11	585.11	**592.05**	**592.05**
r109	771.29	753.56	753.56	753.56	753.56	753.56	753.56	**771.29**	**771.29**
r110	672.50	672.47	672.47	672.47	672.47	672.47	**672.50**	**672.50**	**672.50**
r111	640.99	638.10	638.10	638.10	638.10	638.10	638.10	**640.99**	**640.99**
r112	626.10	613.36	613.36	613.36	613.36	613.36	613.36	624.09	624.09
rc101	620.96	604.36	604.36	604.36	604.36	604.36	614.17	**620.96**	**620.96**
rc102	657.05	615.11	615.11	620.60	637.77	615.89	623.74	**657.05**	**657.05**
rc103	532.70	491.75	491.75	491.75	491.75	493.76	498.36	**532.70**	**532.70**
rc104	515.11	485.53	485.53	485.53	485.53	486.83	488.22	**515.11**	**515.11**
rc105	629.58	577.32	577.32	577.32	608.99	584.48	595.46	**629.58**	**629.58**
rc106	484.59	422.20	422.20	422.20	457.63	430.62	435.39	**484.59**	**484.59**
rc107	501.20	470.26	470.26	476.34	470.26	470.44	472.75	**501.20**	**501.20**
rc108	494.73	487.50	487.50	487.83	487.67	487.64	488.08	**494.73**	**494.73**

Table VIII Lower bounds of BPC on the Solomon Type-2 instances with 50 customers in Group-2

Inst.	*UB*	*LB*	*CC*	*SC*	*KP*	*BP*	*Cli*	*SR*	*All*
c201	361.80	**361.80**	**361.80**	**361.80**	**361.80**	**361.80**	**361.80**	**361.80**	**361.80**
c202	277.55	**277.55**	**277.55**	**277.55**	**277.55**	**277.55**	**277.55**	**277.55**	**277.55**
c203	277.94	**277.94**	**277.94**	**277.94**	**277.94**	**277.94**	**277.94**	**277.94**	**277.94**
c204	284.94	280.82	280.82	280.82	280.82	280.82	**284.94**	**284.94**	**284.94**
c205	345.61	339.06	339.06	339.06	339.06	**345.61**	341.24	**345.61**	**345.61**
c206	309.32	307.07	307.07	307.07	307.07	307.07	307.72	**309.32**	**309.32**
c207	344.53	343.64	343.64	343.64	343.64	343.64	**344.53**	**344.53**	**344.53**
c208	299.96	297.45	297.45	297.45	297.45	**299.96**	**299.96**	**299.96**	**299.96**
r201	763.05	**763.05**	**763.05**	**763.05**	**763.05**	**763.05**	**763.05**	**763.05**	**763.05**
r202	687.54	685.29	685.29	685.29	685.29	685.29	685.29	**687.54**	**687.54**
r203	607.65	600.77	600.77	600.77	600.77	600.77	600.77	**607.65**	**607.65**
r204	509.25	505.84	505.84	505.84	505.84	505.84	505.84	**509.25**	**509.25**
r205	692.40	685.08	685.08	685.08	685.08	685.08	685.08	**692.40**	**692.40**
r206	629.83	620.83	620.83	620.83	620.83	620.83	620.96	**629.83**	**629.83**
r207	568.74	563.38	563.38	563.38	563.38	563.38	563.38	**568.74**	**568.74**
r208	490.05*	-	-	-	-	-	-	-	-
r209	603.10	600.28	600.28	600.28	600.28	600.28	600.28	**603.10**	**603.10**
r210	636.09	620.43	620.43	620.43	620.43	620.43	620.55	**636.09**	**636.09**
r211	537.98	531.70	531.70	531.70	531.70	531.70	531.70	**537.98**	**537.98**
rc201	605.96	598.94	598.94	598.94	598.94	598.94	603.41	**605.96**	**605.96**
rc202	569.44	550.68	550.68	550.68	550.68	550.68	561.00	**569.44**	**569.44**
rc203	482.32	464.61	464.61	464.61	464.61	464.61	469.00	**482.32**	**482.32**
rc204	434.06	417.30	417.30	417.30	417.30	419.15	419.79	-	-
rc205	575.01	553.43	553.43	553.43	553.43	557.39	562.17	**575.01**	**575.01**
rc206	446.07	410.38	410.38	410.38	410.38	418.64	420.60	**446.07**	**446.07**
rc207	461.03	445.62	445.62	445.62	445.62	445.81	446.17	**461.03**	**461.03**
rc208	428.00	425.02	425.02	425.02	425.02	425.02	425.02	**428.00**	**428.00**

Table IX Lower bounds of BPC on the Solomon Type-1 instances with 100 customers in Group-1

Inst.	*UB*	*LB*	*CC*	*SC*	*KP*	*BP*	*Cli*	*SR*	*All*
c101	679.64	663.92	666.96	668.42	663.92	663.92	663.92	668.42	668.42
c102	931.81	926.05	926.05	931.81	926.05	926.58	927.60	**931.81**	**931.81**
c103	747.02	720.96	720.96	720.96	720.96	726.01	724.70	**747.02**	746.57
c104	669.32	666.23	666.54	666.54	666.23	666.23	668.74	**669.32**	**669.32**
c105	787.59	778.87	779.81	780.13	778.87	778.87	782.72	786.57	786.20
c106	839.19	819.17	821.05	821.05	819.17	819.17	825.69	835.48	835.19
c107	827.43	824.77	824.77	824.77	825.66	824.77	824.77	**827.43**	**827.43**
c108	695.98	654.66	655.80	660.66	654.66	655.87	659.52	683.38	686.54
c109	733.77	729.82	731.87	**733.77**	729.82	729.82	729.82	**733.77**	**733.77**
r101	1574.33	1562.27	1562.27	1562.27	1562.40	1562.29	1562.44	1567.66	1567.66
r102	1397.62	1391.11	1391.11	1391.11	1391.11	1391.11	1391.74	1393.80	1393.80
r103	1186.60	1171.67	1171.67	1171.67	1171.70	1174.55	1173.68	1183.88	1183.93
r104	978.83	953.19	953.19	953.46	954.39	953.48	956.76	976.13	976.58
r105	1171.56	1159.61	1159.61	1159.61	1161.49	1160.45	1159.86	**1171.56**	**1171.56**
r106	1130.98	1126.34	1126.34	1126.34	1126.34	1126.34	1128.25	**1130.98**	**1130.98**
r107	1108.03	1063.60	1063.60	1063.60	1065.04	1066.15	1066.90	1100.48	1100.13
r108	916.49	895.75	895.75	895.75	895.75	899.00	900.05	913.95	914.13
r109	1160.64	1139.43	1139.43	1139.43	1139.67	1139.45	1142.39	1159.94	1160.60
r110	1045.75	1015.42	1015.42	1015.42	1015.42	1016.15	1016.98	1031.13	1031.32
r111	1018.71	1012.70	1012.70	1012.70	1012.70	1012.70	1013.43	1018.38	1018.38
r112	871.12	870.55	870.55	870.58	870.55	870.55	870.65	**871.12**	**871.12**
rc101	1283.82	1261.37	1265.60	1265.60	1277.03	1261.37	1270.01	**1283.82**	**1283.82**
rc102	1410.60	1355.03	1355.03	1355.03	1356.84	1357.52	1362.87	1396.21	1395.85
rc103	1116.95	1100.61	1108.93	1108.93	1102.11	1100.89	1100.89	**1116.95**	**1116.95**
rc104	1132.02	1105.35	1117.56	1117.72	1105.35	1105.35	1105.39	1130.22	1131.56
rc105	1268.34	1257.07	1257.07	1257.07	1260.53	1257.07	1257.07	1263.10	1263.10
rc106	1216.71	1187.65	1189.16	1189.31	1187.65	1191.83	1194.04	1208.12	1209.54
rc107	1001.52	977.23	979.11	979.11	977.28	977.23	977.23	1001.52	1000.89
rc108	1109.49	1060.01	1060.60	1060.88	1060.01	1062.91	1062.80	1101.96	1102.63

Table X Lower bounds of BPC on the Solomon Type-2 instances with 100 customers in Group-1

Inst.	*UB*	*LB*	*CC*	*SC*	*KP*	*BP*	*Cli*	*SR*	*All*
c201	682.67	**682.67**	**682.67**	**682.67**	**682.67**	**682.67**	**682.67**	**682.67**	**682.67**
c202	829.35*	746.68	746.68	746.68	746.68	759.96	755.68	-	-
c203	666.30*	624.06	624.06	624.06	624.06	629.31	635.05	-	-
c204	604.78*	581.30	581.30	581.30	581.30	581.30	581.86	-	-
c205	675.30	639.93	639.93	639.93	639.93	639.93	645.02	**675.30**	**675.30**
c206	744.82*	651.38	651.38	651.38	651.38	651.38	666.49	-	-
c207	757.66*	715.89	715.89	715.89	715.89	715.89	717.84	-	-
c208	656.32	635.50	635.50	635.50	635.50	635.50	638.53	650.35	650.31
r201	1581.66*	1498.16	1498.16	1498.16	1498.16	1506.16	1509.40	-	-
r202	1319.94*	1240.96	1240.96	1240.96	1240.96	1245.97	1246.49	-	-
r203	1066.99*	-	-	-	-	-	-	-	-
r204	963.78*	-	-	-	-	-	-	-	-
r205	1046.88*	996.16	996.16	996.16	996.16	996.36	996.37	-	-
r206	1060.63*	-	-	-	-	-	-	-	-
r207	1384.28*	-	-	-	-	-	-	-	-
r208	886.70*	-	-	-	-	-	-	-	-
r209	1039.99*	944.02	944.02	944.02	-	-	-	-	-
r210	996.24*	934.83	934.83	934.83	934.83	934.86	935.79	-	-
r211	999.28*	-	-	-	-	-	-	-	-
rc201	1367.44*	1311.19	1311.19	1311.19	1311.38	1312.08	1318.62	-	1356.00
rc202	1307.11*	1260.02	1260.02	1260.02	1260.02	1260.02	1260.02	-	-
rc203	1055.32*	985.42	985.42	985.42	985.42	987.33	989.12	-	-
rc204	911.25*	-	-	-	-	-	-	-	-
rc205	1271.52*	1221.79	1221.79	1221.79	1221.79	1221.79	1225.00	-	-
rc206	1201.03*	1124.45	1124.45	1124.45	1124.45	1124.46	1127.45	-	-
rc207	960.95*	922.11	922.11	922.11	922.11	922.14	922.14	-	-
rc208	931.19*	-	-	-	-	-	-	-	-

Table XI Lower bounds of BPC on the Solomon Type-1 instances with 100 customers in Group-2

Inst.	*UB*	*LB*	*CC*	*SC*	*KP*	*BP*	*Cli*	*SR*	*All*
c101	639.18	**639.18**	**639.18**	**639.18**	**639.18**	**639.18**	**639.18**	**639.18**	**639.18**
c102	817.14	**817.14**	**817.14**	**817.14**	**817.14**	**817.14**	**817.14**	**817.14**	**817.14**
c103	747.02	720.96	720.96	720.96	720.96	726.01	724.70	**747.02**	746.57
c104	669.32	666.23	666.54	666.54	666.23	666.23	668.74	**669.32**	**669.32**
c105	784.63	777.14	778.21	778.81	777.14	777.14	778.52	**784.63**	**784.63**
c106	796.49	789.15	792.22	794.68	789.15	789.15	791.31	**796.49**	**796.49**
c107	798.45	**798.45**	**798.45**	**798.45**	**798.45**	**798.45**	**798.45**	**798.45**	**798.45**
c108	675.71	648.20	651.36	658.09	648.20	648.20	652.39	667.72	670.63
c109	733.77	729.82	731.87	**733.77**	729.82	729.82	729.82	**733.77**	**733.77**
r101	1524.84	1519.93	1519.93	1519.93	1519.93	1519.93	1519.93	**1524.84**	**1524.84**
r102	1392.88	1388.39	1388.39	1388.39	1388.42	1388.39	1388.60	1390.13	1390.41
r103	1163.44	1150.77	1150.77	1150.77	1150.77	1151.97	1151.66	**1163.44**	**1163.44**
r104	959.20	937.92	937.95	937.95	938.46	937.92	938.54	957.87	957.93
r105	1168.84	1159.13	1159.13	1159.13	1160.52	1159.73	1159.31	**1168.84**	**1168.84**
r106	1130.98	1118.57	1118.57	1118.57	1118.57	1118.57	1119.33	**1130.98**	**1130.98**
r107	1041.79	1020.75	1020.75	1020.75	1020.75	1020.75	1021.85	1038.46	1038.51
r108	905.73	882.80	883.00	883.00	882.80	885.12	885.95	903.82	903.49
r109	1122.44	1096.98	1096.98	1096.98	1097.21	1096.98	1097.96	1114.95	1115.34
r110	1008.52	992.87	992.87	992.87	992.93	992.98	993.52	**1008.52**	**1008.52**
r111	1018.04	1005.69	1005.69	1005.69	1005.69	1005.69	1006.03	**1018.04**	1017.98
r112	871.12	870.55	870.55	870.58	870.55	870.55	870.65	**871.12**	**871.12**
rc101	1263.36	1243.74	1250.14	1250.14	1244.48	1243.74	1246.26	**1263.36**	**1263.36**
rc102	1296.04	1275.90	1277.62	1277.62	1275.90	1277.08	1277.20	**1296.04**	**1296.04**
rc103	1116.95	1100.21	1108.81	1108.82	1101.48	1100.27	1100.27	**1116.95**	**1116.95**
rc104	1082.31	1053.99	1070.77	1070.78	1053.99	1053.99	1053.99	1081.63	**1082.31**
rc105	1265.10	1253.87	1253.87	1253.87	1256.34	1253.87	1253.87	1258.87	1258.87
rc106	1190.90	1173.63	1175.49	1175.87	1174.23	1174.24	1176.83	**1190.90**	**1190.90**
rc107	1001.52	976.77	978.57	978.57	976.77	976.77	976.77	1000.32	999.87
rc108	1050.98	1015.64	1020.56	1020.75	1015.64	1015.70	1015.64	1048.97	1048.77

Table XII Lower bounds of BPC on the Solomon Type-2 instances with 100 customers in Group-2

Inst.	UB	LB	CC	SC	KP	BP	Cli	SR	All
c201	591.37	**591.37**	**591.37**	**591.37**	**591.37**	**591.37**	**591.37**	**591.37**	**591.37**
c202	591.56	**591.56**	**591.56**	**591.56**	**591.56**	**591.56**	**591.56**	**591.56**	**591.56**
c203	590.99	590.06	590.06	590.06	590.06	590.06	590.06	**590.99**	**590.99**
c204	565.78	**565.78**	**565.78**	**565.78**	**565.78**	**565.78**	**565.78**	**565.78**	**565.78**
c205	588.88	**588.88**	**588.88**	**588.88**	**588.88**	**588.88**	**588.88**	**588.88**	**588.88**
c206	588.49	**588.49**	**588.49**	**588.49**	**588.49**	**588.49**	**588.49**	**588.49**	**588.49**
c207	588.29	**588.29**	**588.29**	**588.29**	**588.29**	**588.29**	**588.29**	**588.29**	**588.29**
c208	568.38	**568.38**	**568.38**	**568.38**	**568.38**	**568.38**	**568.38**	**568.38**	**568.38**
r201	1139.81	1134.53	1134.53	1134.53	1134.53	1134.53	1134.53	1136.38	1136.38
r202	1033.55	1022.75	1022.75	1022.75	1022.75	1022.75	1022.75	1031.88	1031.88
r203	874.87	870.96	870.96	870.96	870.96	870.96	870.96	**874.87**	**874.87**
r204	749.40*	729.28	729.28	729.28	729.28	729.28	729.28	-	-
r205	945.14	935.95	935.95	935.95	935.95	935.95	935.95	**945.14**	**945.14**
r206	896.39*	867.34	867.34	867.34	867.34	867.34	867.34	-	-
r207	797.99*	794.59	794.59	794.59	794.59	794.59	794.59	-	-
r208	722.33*	-	-	-	-	-	-	-	-
r209	859.39*	843.99	843.99	843.99	843.99	843.99	843.99	856.83	856.83
r210	927.90*	888.04	888.04	888.04	888.04	888.04	888.04	-	-
r211	849.01*	740.81	740.81	740.81	740.81	740.81	740.81	-	-
rc201	1138.46	1137.50	1137.50	1137.50	1137.50	1137.56	1137.68	**1138.46**	**1138.46**
rc202	1075.29	1061.70	1061.70	1061.70	1061.70	1061.70	1064.29	**1075.29**	**1075.29**
rc203	906.36	891.46	891.46	891.46	891.46	891.46	891.46	**906.36**	**906.36**
rc204	812.29*	-	-	-	-	-	-	-	-
rc205	1087.54	1084.40	1084.40	1084.40	1084.40	1084.40	1084.40	**1087.54**	1087.52
rc206	1047.24	1030.80	1030.80	1030.80	1030.80	1030.80	1031.02	**1047.24**	**1047.24**
rc207	889.99	877.89	877.89	877.89	877.89	877.89	877.89	**889.99**	**889.99**
rc208	817.12*	758.59	758.59	758.59	758.59	758.59	758.59	-	-

Appendix B: Detailed Integer Results

Tables XIII-XXIV summarize the detailed integer results. Columns *BPC* presents the results obtained by the branch-and-price-and-cut (BPC) algorithm, including the lower bound at the root node obtained with and without inequalities (*LPC* and *LP*, respectively), the number of nodes in the branching tree (*Nodes*), the number of inequalities added to the tree (*Cuts*), the total time to solve the instance (*TotT*), and the time to identify the inequalities (*SepT*). Columns *BB* present the results obtained by the branch-and-bound (BB) algorithm, including the lower bound (*LB*) of the instance, the number of nodes in the branching tree (*Nodes*), the time to achieve the lower bound (*LBT*) and the total time to solve the instance (*TotT*). The dashes (-) indicate that the corresponding instance cannot be solved within the given time or physical memory.

Table XIII Detailed results for the Solomon Type-1 instances with 25 customers in Group-1

		BPC						BB			
Inst.	*UB*	*LP*	*LPC*	*Nodes*	*Cuts*	*SepT*	*TotT*	*LB*	*Nodes*	*LBT*	*TotT*
c101	138.10	131.99	138.10	1	10	0.5	1.6	114.25	9	0.4	0.7
c102	195.69	191.79	195.69	1	43	1.1	3.9	144.37	35	2.8	4.5
c103	177.91	174.77	177.91	1	22	0.6	3.8	140.73	17	3.9	5.6
c104	192.93	187.46	192.93	1	61	4.4	18.7	156.90	10	9.4	11.4
c105	182.99	181.40	182.99	1	5	0.2	0.8	139.07	9	0.5	1.0
c106	190.21	169.93	190.21	1	56	0.7	1.9	117.87	14	0.5	1.1
c107	194.27	187.81	194.27	1	15	0.4	1.8	117.34	30	0.6	2.1
c108	176.63	175.58	176.63	1	3	0.2	1.3	140.46	28	1.1	2.7
c109	166.54	166.49	166.54	1	3	0.3	2.1	131.84	8	1.8	2.3
r101	565.28	557.61	565.28	1	3	0.1	0.5	526.21	10	0.6	1.0
r102	558.20	551.82	558.20	1	12	0.2	0.7	554.52	1	0.7	0.9
r103	454.09	453.68	453.68	3	0	0.1	0.9	444.60	17	0.9	1.8
r104	422.98	421.61	422.98	1	20	0.5	1.7	392.40	15	0.7	2.1
r105	502.55	499.54	502.55	1	6	0.2	0.7	485.60	7	1.0	1.5
r106	472.12	472.12	472.12	1	0	0.0	0.7	469.64	1	1.5	1.6
r107	477.28	475.03	477.28	1	5	0.3	1.5	444.32	7	2.1	2.6
r108	410.56	401.40	406.09	3	44	1.3	4.0	400.09	10	2.9	3.5
r109	442.40	437.67	442.40	1	11	0.1	0.8	405.54	5	1.2	1.8
r110	460.31	439.41	457.83	3	34	0.6	2.1	426.56	16	2.9	3.6
r111	430.35	421.46	428.43	3	12	0.3	1.5	411.48	19	2.3	4.3
r112	449.91	429.06	449.91	1	46	1.6	4.0	412.66	11	4.7	5.9
rc101	350.66	322.67	350.66	1	23	0.5	1.3	301.58	8	0.7	0.9
rc102	259.61	255.04	259.61	1	20	0.7	1.9	166.88	16	0.5	1.1
rc103	324.15	323.93	324.15	1	2	0.3	1.2	234.38	18	0.7	1.7
rc104	239.85	227.45	239.85	1	25	1.9	4.8	182.50	14	1.0	1.9
rc105	455.00	357.83	455.00	1	31	1.5	2.9	304.86	10	0.5	1.0
rc106	303.92	285.62	303.92	1	15	0.3	1.1	172.80	28	0.8	1.8
rc107	236.47	200.84	236.47	1	36	0.6	2.1	148.13	18	2.3	3.1
rc108	291.59	282.52	291.59	1	16	0.2	2.2	161.43	44	3.7	8.3

Table XIV Detailed results for the Solomon Type-2 instances with 25 customers in Group-1

		BPC						BB			
Inst.	*UB*	*LP*	*LPC*	*Nodes*	*Cuts*	*SepT*	*TotT*	*LB*	*Nodes*	*LBT*	*TotT*
c201	148.32	139.32	148.32	1	1	0.0	0.6	94.69	11	0.6	1.0
c202	200.73	197.01	200.73	1	2	0.0	2.5	133.20	30	1.5	4.4
c203	177.20	177.20	177.20	1	0	0.0	3.0	130.19	15	2.5	4.1
c204	151.27	148.30	151.27	1	10	0.1	6.5	128.91	4	4.4	5.5
c205	151.48	151.48	151.48	1	0	0.0	0.9	111.82	8	0.8	1.4
c206	207.24	203.20	207.24	1	31	0.2	3.6	136.41	9	1.0	1.8
c207	179.68	179.68	179.68	1	0	0.0	2.4	124.43	12	1.4	2.9
c208	195.17	186.99	195.17	1	1	0.0	1.7	154.36	30	1.2	2.7
r201	485.76	479.06	485.76	1	12	0.1	1.6	450.42	28	1.7	5.2
r202	447.46	440.47	445.38	3	52	0.2	5.4	384.55	9	1.4	5.2
r203	400.40	400.40	400.40	1	0	0.0	2.9	388.88	10	1.6	4.2
r204	385.54	379.78	385.54	1	20	0.1	813.4	355.89	19	5.2	69.4
r205	405.98	401.83	405.98	1	30	0.1	2.4	391.40	14	2.0	6.8
r206	378.18	378.18	378.18	1	0	0.0	2.2	375.48	5	1.9	4.3
r207	362.79	361.30	362.79	1	20	0.1	3.6	362.63	1	3.8	5.9
r208	329.33	329.33	329.33	1	0	0.0	703.8	329.33	1	4.5	4.6
r209	403.65	385.92	403.65	1	20	0.1	114.8	371.56	10	5.1	6.7
r210	410.60	405.70	410.60	1	50	0.3	4.5	391.69	13	2.2	4.8
r211	361.69	361.38	361.69	1	10	0.1	1078.5	351.91	5	11.7	19.9
rc201	334.04	313.44	334.04	1	22	0.1	1.6	195.66	28	0.6	2.3
rc202	245.26	245.26	245.26	1	0	0.0	3.3	175.80	10	1.8	2.4
rc203	307.79	307.55	307.79	1	10	0.1	2.8	289.60	10	1.9	3.5
rc204	228.85	217.82	228.85	1	20	0.1	17.2	170.87	11	2.1	3.3
rc205	345.39	309.09	345.39	1	29	0.2	3.9	215.20	19	1.3	4.3
rc206	286.60	261.98	286.60	1	16	0.1	3.2	157.40	28	1.3	3.2
rc207	222.39	193.16	222.39	1	34	0.2	11.4	142.07	13	3.3	4.0
rc208	251.19	247.57	251.19	1	13	0.1	104.4	209.94	30	16.6	29.6

Table XV Detailed results for the Solomon Type-1 instances with 25 customers in Group-2

		BPC						BB			
Inst.	*UB*	*LP*	*LPC*	*Nodes*	*Cuts*	*SepT*	*TotT*	*LB*	*Nodes*	*LBT*	*TotT*
c101	138.10	131.99	138.10	1	10	0.4	1.5	114.25	9	0.4	0.6
c102	190.74	190.74	190.74	1	0	0.0	1.0	144.37	32	0.9	2.1
c103	177.91	174.77	177.91	1	22	0.4	2.8	140.73	17	2.1	3.9
c104	187.45	187.45	187.45	1	0	0.0	1.5	156.90	11	2.0	2.9
c105	182.99	181.40	182.99	1	5	0.2	0.8	139.07	10	0.4	1.1
c106	190.21	169.93	190.21	1	56	0.6	1.7	117.87	15	0.4	1.1
c107	191.81	187.81	191.81	1	15	0.3	1.1	117.34	31	0.6	2.7
c108	176.63	175.58	176.63	1	3	0.2	1.0	140.46	28	0.8	1.9
c109	166.54	166.49	166.54	1	3	0.2	1.6	131.84	8	1.1	1.6
r101	565.15	557.53	565.15	1	5	0.1	0.4	526.21	14	0.4	0.9
r102	538.51	537.93	538.51	1	7	0.1	0.6	535.83	1	0.6	0.9
r103	453.27	453.27	453.27	1	0	0.0	0.4	444.60	15	0.6	1.7
r104	409.30	409.30	409.30	1	0	0.0	0.5	392.40	9	0.6	1.2
r105	501.72	497.41	501.72	1	5	0.3	0.9	485.60	8	0.5	0.8
r106	453.52	451.80	453.52	1	10	0.1	0.7	453.19	2	1.3	1.7
r107	425.27	425.27	425.27	1	0	0.0	0.6	425.27	1	0.7	0.8
r108	398.29	397.24	398.29	1	10	0.7	1.7	394.55	3	1.2	1.8
r109	442.40	437.67	442.40	1	10	0.1	0.7	405.54	5	0.6	1.0
r110	445.18	433.01	444.19	9	51	0.6	3.3	426.56	14	2.1	5.3
r111	427.77	421.46	427.77	1	10	0.2	1.0	411.48	17	1.1	2.8
r112	394.10	388.16	394.10	1	33	0.3	1.8	394.10	1	2.3	2.4
rc101	350.66	322.67	350.66	1	23	0.3	0.7	301.58	8	0.6	0.8
rc102	259.61	255.04	259.61	1	20	0.7	1.5	166.88	16	0.5	0.9
rc103	324.15	323.93	324.15	1	2	0.1	0.7	234.38	18	0.7	2.1
rc104	239.85	227.45	239.85	1	25	1.4	3.1	182.50	14	0.8	1.4
rc105	348.42	321.94	348.42	1	7	0.1	0.7	291.02	8	0.5	0.9
rc106	303.92	285.62	303.92	1	15	0.3	0.9	172.80	32	0.6	2.1
rc107	236.47	200.84	236.47	1	36	0.6	1.7	148.13	18	0.7	1.4
rc108	291.59	282.52	291.59	1	16	0.6	2.3	161.43	45	0.9	3.0

Table XVI Detailed results for the Solomon Type-2 instances with 25 customers in Group-2

		BPC						BB			
Inst.	*UB*	*LP*	*LPC*	*Nodes*	*Cuts*	*SepT*	*TotT*	*LB*	*Nodes*	*LBT*	*TotT*
c201	148.32	139.32	148.32	1	1	0.0	0.6	94.69	11	0.5	1.0
c202	200.73	197.01	200.73	1	2	0.0	2.3	133.20	30	1.4	4.0
c203	177.20	177.20	177.20	1	0	0.0	2.6	130.19	15	2.3	3.2
c204	151.27	148.30	151.27	1	10	0.2	17.8	128.91	4	3.9	4.5
c205	151.48	151.48	151.48	1	0	0.0	1.2	111.82	8	0.8	1.5
c206	207.24	202.57	207.24	1	31	0.2	3.4	136.41	9	0.9	1.7
c207	179.68	179.68	179.68	1	0	0.0	2.4	124.43	12	1.6	2.5
c208	195.17	186.99	195.17	1	1	0.0	2.4	154.36	30	1.4	2.9
r201	457.56	454.82	457.56	1	6	0.0	0.9	450.42	6	1.7	2.8
r202	411.49	411.49	411.49	1	0	0.0	1.0	384.55	3	1.6	2.5
r203	392.33	392.33	392.33	1	0	0.0	2.0	388.88	2	2.2	2.6
r204	355.89	352.92	355.89	1	30	0.1	3.6	355.89	1	5.2	5.4
r205	394.06	391.67	394.06	1	10	0.0	1.4	391.40	2	2.0	2.5
r206	375.48	374.62	375.48	1	10	0.0	2.0	375.48	1	1.9	2.1
r207	362.63	361.14	362.63	1	30	0.1	4.2	362.63	1	3.9	4.1
r208	329.33	329.33	329.33	1	0	0.0	3.1	329.33	1	3.8	3.9
r209	371.56	365.03	371.56	1	40	0.2	3.4	371.56	1	4.0	4.1
r210	405.48	405.05	405.48	1	10	0.1	3.0	391.69	18	2.0	5.0
r211	351.91	342.31	351.91	1	50	0.2	6.9	351.91	1	12.4	12.7
rc201	334.04	313.44	334.04	1	22	0.1	2.0	195.66	28	0.6	2.0
rc202	245.26	245.26	245.26	1	0	0.0	4.2	175.80	10	1.7	2.1
rc203	307.79	307.55	307.79	1	10	0.1	3.4	289.60	10	1.7	2.9
rc204	228.85	217.82	228.85	1	20	0.2	16.3	170.87	11	2.1	3.7
rc205	328.25	301.68	328.25	1	19	0.1	3.6	215.20	15	1.3	2.3
rc206	286.60	261.98	286.60	1	16	0.1	3.1	157.40	28	1.2	3.1
rc207	222.39	193.16	222.39	1	34	0.2	8.7	142.07	13	3.0	3.5
rc208	251.19	247.57	251.19	1	13	0.1	56.2	209.94	33	13.6	25.1

Table XVII Detailed results for the Solomon Type-1 instances with 50 customers in Group-1

		BPC						BB			
Inst.	*UB*	*LP*	*LPC*	*Nodes*	*Cuts*	*SepT*	*TotT*	*LB*	*Nodes*	*LBT*	*TotT*
c101	356.67	332.47	355.61	3	62	2.3	7.1	287.11	44	1.1	4.7
c102	362.67	343.31	356.83	5	161	6.4	49.8	312.03	47	2.6	10.8
c103	344.16	324.24	339.59	5	180	21.3	530.1	284.77	105	7.0	93.0
c104	343.78	342.01	343.78	1	4	0.2	19.2	313.93	91	17.2	246.6
c105	334.80	334.07	334.80	1	15	0.5	2.1	276.95	101	1.3	8.8
c106	315.94	303.43	315.94	1	15	1.5	3.5	242.88	81	2.1	7.8
c107	373.95	369.84	370.28	3	2	1.0	3.7	297.46	214	1.9	26.4
c108	309.29	305.59	309.29	1	4	0.2	3.9	292.73	11	4.1	6.6
c109	368.28	355.06	368.28	1	103	2.7	21.7	311.19	55	5.0	19.7
r101	941.99	934.44	941.99	1	3	0.6	1.2	899.94	211	1.2	10.4
r102	800.74	798.88	800.74	1	10	0.4	1.5	751.09	34	1.8	7.4
r103	710.96	707.71	710.96	1	10	0.6	2.4	688.24	59	6.4	20.5
r104	726.12	696.35	723.73	3	220	7.5	872.2	648.51	26	48.0	56.0
r105	881.38	857.90	881.38	1	26	1.1	2.6	823.70	179	4.8	16.4
r106	718.37	700.76	716.05	3	59	2.5	9.5	642.06	161	5.4	14.2
r107	679.09	670.34	679.09	1	69	5.5	11.8	655.19	45	15.7	34.8
r108	659.58	639.13	659.58	1	64	6.4	79.8	587.84	13	53.7	66.8
r109	815.88	770.42	814.79	3	171	6.9	20.2	719.09	67	9.0	15.8
r110	672.50	672.47	672.50	1	1	0.4	2.6	645.51	62	8.1	19.0
r111	640.99	638.10	640.99	1	10	0.8	3.1	598.25	43	17.9	32.6
r112	626.10	617.51	626.10	1	71	3.6	11.6	615.76	23	40.5	47.7
rc101	632.55	623.69	632.55	1	4	0.5	1.1	528.09	54	7.2	5.3
rc102	701.64	615.32	661.88	7	250	10.5	25.7	602.10	388	19.7	4.7
rc103	532.70	491.75	532.70	1	27	1.8	4.4	413.61	160	29.9	16.2
rc104	515.11	485.53	515.11	1	74	2.9	28.5	386.76	309	309.3	39.0
rc105	629.58	577.32	629.58	1	17	1.4	2.9	569.22	68	9.5	3.5
rc106	484.59	422.20	484.59	1	32	2.1	4.3	417.92	188	14.9	7.9
rc107	501.20	470.26	501.20	1	43	1.2	6.3	411.50	74	30.3	15.8
rc108	494.73	487.50	494.73	1	45	2.1	15.4	388.63	361	648.6	72.1

Table XVIII Detailed results for the Solomon Type-2 instances with 50 customers in Group-1

		BPC						BB			
Inst.	*UB*	*LP*	*LPC*	*Nodes*	*Cuts*	*SepT*	*TotT*	*LB*	*Nodes*	*LBT*	*TotT*
c201	384.41	381.37	384.41	1	13	0.2	6.8	330.27	102	5.3	76.4
c202	277.55	277.55	277.55	1	0	0.0	9.7	255.83	11	7.5	13.8
c203	277.94	277.94	277.94	1	0	0.0	17.8	255.93	10	9.4	15.0
c204	284.94	280.82	284.94	1	4	0.0	243.1	265.28	10	25.2	73.3
c205	345.61	339.06	345.61	1	3	0.0	8.9	243.45	228	6.2	49.0
c206	309.32	307.07	309.32	1	22	0.4	18.5	262.08	68	6.7	14.8
c207	344.53	343.64	344.53	1	2	0.0	13.7	279.48	206	12.1	128.1
c208	299.96	297.45	299.96	1	2	0.0	16.5	264.55	18	11.5	18.0
r201	796.23	793.64	796.23	1	14	0.1	8.6	718.56	842	7.0	495.5
r202	695.31	689.71	695.31	1	80	0.5	33.8	646.59	63	10.2	93.8
r203	619.52	607.27	619.52	1	150	1.0	797.1	572.22	109	39.0	421.4
r204	654.40*	-	-	-	-	-	-	508.66	3	-	-
r205	730.53	714.27	730.53	1	133	0.9	554.3	681.62	212	22.6	592.9
r206	641.11	627.11	-	-	142	1.0	-	588.43	380	99.5	1165.9
r207	634.57*	-	-	-	-	-	-	549.14	7	-	-
r208	515.76*	-	-	-	-	-	-	-	-	-	-
r209	638.65	617.18	638.65	1	141	1.6	3764.1	581.70	74	11.9	303.9
r210	640.18	627.58	640.18	1	71	0.5	983.3	610.17	57	167.1	380.7
r211	550.20*	540.69	-	-	30	0.4	-	537.16	32	355.0	-
rc201	627.76	613.52	627.76	1	14	0.1	5.9	437.68	520	2.5	177.1
rc202	569.44	550.68	569.44	1	28	0.2	16.8	418.53	247	5.2	128.1
rc203	482.32	467.98	482.32	1	16	0.2	40.4	393.15	122	8.3	282.9
rc204	434.06	417.30	-	-	25	0.5	-	271.33	181	52.8	10560.5
rc205	575.01	553.43	575.01	1	41	0.3	13.4	380.10	641	3.8	318.0
rc206	446.07	410.38	446.07	1	40	0.5	31.4	306.49	515	6.8	108.7
rc207	461.03	445.62	461.03	1	23	0.3	44.0	331.26	105	18.4	206.8
rc208	428.00	425.02	428.00	1	21	0.2	354.8	311.94	500	1004.0	12065.0

Table XIX Detailed results for the Solomon Type-1 instances with 50 customers in Group-2

		BPC						BB			
Inst.	*UB*	*LP*	*LPC*	*Nodes*	*Cuts*	*SepT*	*TotT*	*LB*	*Nodes*	*LBT*	*TotT*
c101	355.61	332.47	354.67	5	65	2.5	8.6	287.11	57	1.2	6.2
c102	351.26	340.53	351.26	1	28	0.5	4.0	312.03	38	1.8	12.1
c103	337.82	324.23	337.82	1	121	14.4	56.9	284.77	109	3.0	40.4
c104	342.46	341.26	342.46	1	4	0.2	18.5	313.93	83	6.8	45.8
c105	334.80	334.07	334.80	1	15	0.5	2.0	276.95	101	1.0	9.4
c106	310.76	300.84	310.76	1	11	0.6	2.5	242.88	78	1.0	6.6
c107	353.90	353.90	353.90	1	0	0.0	1.5	297.46	116	1.9	20.7
c108	309.29	305.59	309.29	1	4	0.2	3.5	292.73	11	2.0	4.9
c109	360.62	348.46	360.62	1	76	1.9	14.7	311.19	42	5.6	18.1
r101	941.99	934.44	941.99	1	4	0.8	1.4	899.94	298	1.0	13.5
r102	800.74	798.88	800.74	1	10	0.4	1.6	751.09	34	1.7	9.5
r103	710.46	707.71	710.46	1	10	0.4	2.2	688.24	40	5.4	19.5
r104	631.32	621.49	631.32	1	87	3.4	11.0	624.91	8	18.7	22.9
r105	850.06	829.11	849.78	3	75	2.0	6.2	817.28	69	3.2	13.0
r106	716.57	700.76	716.05	3	86	2.6	8.1	642.06	218	2.4	18.4
r107	679.09	670.34	679.09	1	86	5.6	12.6	655.19	49	9.6	29.6
r108	592.05	585.11	592.05	1	34	2.4	7.6	569.78	13	17.4	54.1
r109	771.29	753.56	771.29	1	93	2.1	6.2	719.09	66	6.1	15.4
r110	672.50	672.47	672.50	1	1	1.2	3.6	645.51	72	3.9	19.0
r111	640.99	638.10	640.99	1	10	0.9	4.0	598.25	43	13.2	23.5
r112	626.10	613.36	624.09	5	189	5.9	66.7	615.11	33	21.8	42.4
rc101	620.96	604.36	620.96	1	14	1.0	1.8	528.09	44	2.3	6.9
rc102	657.05	615.11	657.05	1	58	2.4	4.9	602.10	91	3.8	12.1
rc103	532.70	491.75	532.70	1	26	1.4	3.9	413.61	196	3.3	12.0
rc104	515.11	485.53	515.11	1	61	3.0	21.7	386.76	330	2.4	40.8
rc105	629.58	577.32	629.58	1	17	1.4	3.1	569.22	71	2.2	10.6
rc106	484.59	422.20	484.59	1	30	1.9	4.4	417.92	188	3.4	11.6
rc107	501.20	470.26	501.20	1	33	1.1	5.1	411.50	85	5.6	13.5
rc108	494.73	487.50	494.73	1	45	2.5	14.7	388.63	398	6.3	37.2

Table XX Detailed results for the Solomon Type-2 instances with 50 customers in Group-2

		BPC						BB			
Inst.	*UB*	*LP*	*LPC*	*Nodes*	*Cuts*	*SepT*	*TotT*	*LB*	*Nodes*	*LBT*	*TotT*
c201	361.80	361.80	361.80	1	0	0.0	3.9	330.27	31	5.2	20.0
c202	277.55	277.55	277.55	1	0	0.0	10.1	255.83	11	7.3	13.3
c203	277.94	277.94	277.94	1	0	0.0	18.2	255.93	10	9.5	14.5
c204	284.94	280.82	284.94	1	4	0.0	34.9	265.28	9	22.8	53.7
c205	345.61	339.06	345.61	1	6	0.1	10.0	243.45	228	6.2	44.7
c206	309.32	307.07	309.32	1	22	0.4	19.0	262.08	68	6.5	14.0
c207	344.53	343.64	344.53	1	2	0.0	13.3	279.48	206	12.0	123.2
c208	299.96	297.45	299.96	1	2	0.0	13.0	264.55	18	13.1	26.5
r201	763.05	763.05	763.05	1	0	0.0	4.0	718.56	276	6.7	132.1
r202	687.54	685.29	687.54	1	40	0.2	16.0	646.59	48	10.4	67.9
r203	607.65	600.77	607.65	1	60	0.4	31.0	572.22	36	37.2	97.1
r204	509.25	505.84	509.25	1	50	0.4	12196.3	508.66	7	7445.1	-
r205	692.40	685.08	692.40	1	60	0.4	17.8	681.62	17	22.6	56.8
r206	629.83	620.83	629.83	1	71	0.4	83.7	588.43	215	93.0	304.0
r207	568.74	563.38	568.74	1	20	0.2	87.9	549.14	13	1110.0	1158.0
r208	490.05*	-	-	-	-	-	-	477.97	1	13557.8	-
r209	603.10	600.28	603.10	1	10	0.1	21.5	581.70	10	11.7	30.0
r210	636.09	620.43	636.09	1	171	1.2	347.7	610.17	50	151.6	243.8
r211	537.98	531.70	537.98	1	40	0.3	249.5	537.16	4	283.3	296.3
rc201	605.96	598.94	605.96	1	15	0.1	3.6	437.68	418	2.3	79.1
rc202	569.44	550.68	569.44	1	28	0.2	18.8	418.53	247	4.9	101.5
rc203	482.32	464.61	482.32	1	25	0.2	47.1	393.15	122	8.0	82.3
rc204	434.06	417.30	-	-	25	0.6	-	271.33	181	43.1	9444.5
rc205	575.01	553.43	575.01	1	41	0.3	13.9	380.10	641	4.2	200.5
rc206	446.07	410.38	446.07	1	40	0.5	32.6	306.49	515	6.7	90.3
rc207	461.03	445.62	461.03	1	23	0.3	49.2	331.26	105	16.7	109.8
rc208	428.00	425.02	428.00	1	21	0.2	557.3	311.94	500	858.0	5937.1

Table XXI Detailed results for the Solomon Type-1 instances with 100 customers in Group-1

		BPC						BB			
Inst.	*UB*	*LP*	*LPC*	*Nodes*	*Cuts*	*SepT*	*TotT*	*LB*	*Nodes*	*LBT*	*TotT*
c101	679.64	663.92	668.42	3	23	4.6	14.0	619.37	48	2.8	134.1
c102	931.81	926.05	931.81	1	41	6.3	28.3	793.99	2732	12.1	619.7
c103	747.02	720.96	746.57	3	168	41.5	317.0	630.92	3294	18.0	4734.5
c104	669.32	666.23	669.32	1	18	7.1	117.0	657.01	11	33.0	73.5
c105	787.59	778.87	786.20	3	47	11.8	25.0	711.55	531	4.7	518.8
c106	839.19	819.17	835.19	7	180	36.3	75.6	684.41	34082	11.8	6882.2
c107	827.43	824.77	827.43	1	11	2.7	11.9	781.92	52	6.1	132.1
c108	695.98	654.66	686.54	13	363	61.0	548.5	552.69	3136	25.6	2106.4
c109	733.77	729.82	733.77	1	11	1.5	21.0	709.68	48	19.3	102.1
r101	1574.33	1562.27	1567.66	3	77	17.5	23.7	1379.14	19279	10.2	3505.9
r102	1397.62	1391.11	1393.80	9	55	15.2	28.9	1339.05	278	7.2	360.4
r103	1186.60	1171.67	1183.93	3	249	40.2	369.1	1085.66	7553	33.2	8074.8
r104	978.83	953.19	976.58	3	226	72.4	7410.8	921.53	1267	611.7	7236.8
r105	1171.56	1159.61	1171.56	1	134	23.0	37.4	1108.77	2521	37.4	1142.5
r106	1130.98	1126.34	1130.98	1	45	10.2	24.9	1032.20	4973	67.6	4618.9
r107	1108.03	1063.60	1100.13	2	264	51.6	-	1005.13	1254	176.6	3512.8
r108	916.49	895.75	914.13	3	270	78.6	5762.5	842.16	1170	3307.5	5885.6
r109	1160.64	1139.43	1160.60	3	285	56.5	169.4	1063.65	2135	76.1	1471.5
r110	1045.75	1015.42	1031.32	17	828	217.8	6412.7	935.46	5123	118.0	4445.4
r111	1018.71	1012.70	1018.38	3	117	19.6	102.8	967.06	989	154.9	1407.1
r112	871.12	870.55	871.12	1	15	5.3	30.3	847.28	164	2655.2	2772.8
rc101	1283.82	1261.37	1283.82	1	37	6.9	13.3	1166.00	587	43.0	558.2
rc102	1410.60	1355.03	1395.85	15	287	46.3	203.8	1286.73	654	97.1	674.7
rc103	1116.95	1100.61	1116.95	1	78	13.4	42.4	1018.33	2055	314.6	1959.1
rc104	1132.02	1105.35	1131.56	3	219	28.9	10500.4	1053.78	644	2880.8	-
rc105	1268.34	1257.07	1263.10	5	133	22.8	43.3	1185.02	5123	25.9	2047.0
rc106	1216.71	1187.65	1209.54	23	423	92.7	387.8	1113.86	5391	148.0	3575.3
rc107	1001.52	977.23	1000.89	3	116	16.7	170.6	924.07	319	227.6	652.7
rc108	1109.49	1060.01	1102.63	7	321	66.3	6030.1	1008.73	1909	3116.7	-

Table XXII Detailed results for the Solomon Type-2 instances with 100 customers in Group-1

		BPC						BB			
Inst.	*UB*	*LP*	*LPC*	*Nodes*	*Cuts*	*SepT*	*TotT*	*LB*	*Nodes*	*LBT*	*TotT*
c201	682.67	682.67	682.67	1	0	0.0	29.4	490.79	5728	19.4	-
c202	829.35*	746.68	-	-	94	5.3	-	591.56	126	30.9	-
c203	666.30*	624.06	-	-	34	1.4	-	521.38	22	197.2	-
c204	604.78*	581.30	-	-	14	0.8	-	512.90	68	3303.9	-
c205	675.30	639.93	675.30	1	89	2.7	604.2	483.59	1307	29.2	-
c206	744.82*	651.38	-	-	185	8.6	-	532.91	915	42.7	-
c207	757.66*	715.89	-	-	75	2.5	-	582.08	71	86.3	-
c208	656.32	635.50	650.31	3	197	6.0	7596.0	453.97	168	133.0	-
r201	1581.66*	1498.16	-	-	157	4.7	-	1105.34	814	101.6	-
r202	1319.94*	1240.96	-	-	40	1.1	-	1009.15	146	251.3	-
r203	1066.99*	-	-	-	-	-	-	871.29	84	757.9	-
r204	963.78*	-	-	-	-	-	-	730.17	7	-	-
r205	1046.88*	996.16	-	-	102	3.5	-	915.12	518	423.0	-
r206	1060.63*	-	-	-	-	-	-	853.90	14	4276.8	-
r207	1384.28*	-	-	-	-	-	-	790.43	1	-	-
r208	886.70*	-	-	-	-	-	-	-	-	-	-
r209	1039.99*	944.02	-	-	5	0.0	-	840.27	4	4127.2	-
r210	996.24*	934.83	-	-	16	0.7	-	861.66	152	3886.5	-
r211	999.28*	-	-	-	-	-	-	740.61	4	-	-
rc201	1367.44*	1311.19	1356.00	2	311	10.2	-	1049.91	958	121.3	-
rc202	1307.11*	1260.02	-	-	54	1.6	-	1003.04	61	166.6	-
rc203	1055.32*	985.42	-	-	25	0.9	-	822.06	706	447.0	-
rc204	911.25*	-	-	-	-	-	-	766.31	13	12414.9	-
rc205	1271.52*	1221.79	-	-	131	3.5	-	931.21	156	116.8	-
rc206	1201.03*	1124.45	-	-	108	2.8	-	997.52	176	159.1	-
rc207	960.95*	922.11	-	-	33	1.2	-	806.80	22	3909.4	-
rc208	931.19*	-	-	-	-	-	-	-	-	-	-

Table XXIII Detailed results for the Solomon Type-1 instances with 100 customers in Group-2

		BPC						BB			
Inst.	*UB*	*LP*	*LPC*	*Nodes*	*Cuts*	*SepT*	*TotT*	*LB*	*Nodes*	*LBT*	*TotT*
c101	639.18	639.18	639.18	1	0	0.0	4.2	619.37	10	2.6	212.5
c102	817.14	817.14	817.14	1	0	0.0	10.4	793.99	66	8.4	115.5
c103	747.02	720.96	746.57	3	168	40.3	314.8	630.92	3470	10.4	3217.2
c104	669.32	666.23	669.32	1	18	7.4	121.2	657.01	11	29.3	79.2
c105	784.63	777.14	784.63	1	43	8.4	19.9	704.93	2070	4.2	1290.2
c106	796.49	789.15	796.49	1	12	2.2	10.4	684.41	10164	5.1	4659.8
c107	798.45	798.45	798.45	1	0	0.0	8.2	759.61	60	5.4	218.3
c108	675.71	648.20	670.63	5	215	34.8	223.0	552.69	1934	9.0	1692.1
c109	733.77	729.82	733.77	1	11	2.0	21.3	709.68	48	15.9	116.4
r101	1524.84	1519.93	1524.84	1	32	4.9	7.8	1379.14	30784	5282.7	1524.8
r102	1392.88	1388.39	1390.41	7	107	21.7	34.5	1330.25	1347	891.4	1392.9
r103	1163.44	1150.77	1163.44	1	182	26.2	66.7	1085.66	5166	3593.2	1163.4
r104	959.20	937.92	957.93	7	484	118.0	11047.6	921.53	1036	3080.1	959.2
r105	1168.84	1159.13	1168.84	1	114	16.9	28.7	1108.77	2349	1069.9	1168.8
r106	1130.98	1118.57	1130.98	1	73	13.5	29.2	1032.20	5947	3848.0	1131.0
r107	1041.79	1020.75	1038.51	11	614	99.9	3973.1	991.58	1217	2465.7	1041.8
r108	905.73	882.80	903.49	4	386	163.2	-	842.16	998	3162.5	905.7
r109	1122.44	1096.98	1115.34	89	1171	258.9	2535.6	1060.18	3668	3340.3	1122.4
r110	1008.52	992.87	1008.52	1	158	26.2	75.1	935.46	2069	1628.0	1008.5
r111	1018.04	1005.69	1017.98	3	130	24.2	104.0	965.52	2174	1528.1	1018.0
r112	871.12	870.55	871.12	1	15	5.5	29.9	847.28	164	829.8	871.1
rc101	1263.36	1243.74	1263.36	1	72	13.9	23.4	1166.00	393	23.2	714.6
rc102	1296.04	1275.90	1296.04	1	115	15.1	31.6	1243.84	162	61.8	565.3
rc103	1116.95	1100.21	1116.95	1	97	12.1	38.0	1018.33	2185	111.0	1480.8
rc104	1082.31	1053.99	1082.31	1	202	26.0	786.0	1042.80	309	394.2	1511.3
rc105	1265.10	1253.87	1258.87	17	52	20.3	50.3	1185.02	5818	17.8	1786.1
rc106	1190.90	1173.63	1190.90	1	78	15.1	33.2	1113.86	2978	121.3	1707.0
rc107	1001.52	976.77	999.87	5	184	27.2	356.0	924.07	322	54.1	441.8
rc108	1050.98	1015.64	1048.77	7	245	29.7	4276.6	1006.27	808	123.1	2246.7

Table XXIV Detailed results for the Solomon Type-2 instances with 100 customers in Group-2

		BPC						BB			
Inst.	*UB*	*LP*	*LPC*	*Nodes*	*Cuts*	*SepT*	*TotT*	*LB*	*Nodes*	*LBT*	*TotT*
c201	591.37	591.37	591.37	1	0	0.0	23.0	490.79	964	18.9	5508.2
c202	591.56	591.56	591.56	1	0	0.0	64.5	591.56	1	30.3	92.4
c203	590.99	590.06	590.99	1	20	0.4	2039.5	521.38	660	166.3	-
c204	565.78	565.78	565.78	1	0	0.0	14251.4	512.90	235	2976.5	14162.9
c205	588.88	588.88	588.88	1	0	0.0	31.7	483.59	1551	28.9	-
c206	588.49	588.49	588.49	1	0	0.0	48.3	532.91	674	42.0	10616.5
c207	588.29	588.29	588.29	1	0	0.0	85.6	582.08	2	84.0	423.3
c208	568.38	568.38	568.38	1	0	0.0	76.7	453.97	1146	131.5	-
r201	1139.81	1134.53	1136.38	7	178	1.0	305.4	1105.34	725	95.4	12587.0
r202	1033.55	1022.75	1031.88	7	136	1.2	3287.5	1009.15	334	220.3	12991.7
r203	874.87	870.96	874.87	1	80	1.8	1718.4	871.29	21	666.5	2678.0
r204	749.40*	729.28	-	-	10	0.5	-	730.17	11	6161.9	-
r205	945.14	935.95	945.14	1	160	3.2	369.3	915.12	1321	384.6	-
r206	896.39*	867.34	-	-	150	3.8	-	853.90	175	3765.7	-
r207	797.99*	794.59	-	-	40	1.2	-	790.43	7	6968.8	-
r208	722.33*	-	-	-	-	-	-	-	-	-	-
r209	859.39*	843.99	856.83	3	190	3.8	-	841.85	52	3875.9	-
r210	927.90*	888.04	-	-	130	3.3	-	851.73	126	9092.0	-
r211	849.01*	740.81	-	-	30	0.9	-	740.61	18	4612.7	-
rc201	1138.46	1137.50	1138.46	1	46	0.5	79.2	1049.91	1920	113.1	13402.2
rc202	1075.29	1061.70	1075.29	1	33	0.4	132.6	1003.04	327	153.5	7628.5
rc203	906.36	891.46	906.36	1	143	2.6	4555.9	822.06	747	384.0	-
rc204	812.29*	-	-	-	-	-	-	766.31	12	11207.1	-
rc205	1087.54	1084.40	1087.52	3	91	1.0	233.7	931.21	3055	106.0	-
rc206	1047.24	1030.80	1047.24	1	194	3.4	377.5	997.52	1268	142.4	-
rc207	889.99	877.89	889.99	1	140	2.9	3023.3	827.59	451	4754.7	10354.6
rc208	817.12*	758.59	-	-	30	0.9	-	-	-	-	-

Appendix C: Detailed Results on the Solomon-100 VRPTW Instances

To evaluate the implementation of our branch-and-price-and-cut (BPC) algorithm, we apply it to solve the 100-customer Solomon VRPTW instances. Table XXV compares the computational results of our BPC algorithm and the two BPC algorithms proposed by Jepsen et al. (2008) and Desaulniers et al. (2008) on these instances. From the table, we can see that our BPC algorithm is comparable to the best BPC algorithms for the VRPTW reported in the literature, especially for the large instances.

Table XXV Results of the Solomon-100 VRPTW instances

		Our			Jepsen et al. (2008)		Desaulniers et al. (2008)		
Inst.	UB	Nodes	Cuts	Time(s)	Nodes	Time(s)	Nodes	Cuts	Time(s)
c101	827.3	1	0	2	1	3	1	0	2
c102	827.3	1	0	7	1	13	1	0	8
c103	826.3	1	0	10	1	34	1	0	28
c104	822.9	1	0	30	1	4113	1	0	86
c105	827.3	1	0	4	1	5	1	0	3
c106	827.3	1	0	4	1	7	1	0	4
c107	827.3	1	0	3	1	7	1	0	4
c108	827.3	1	0	7	1	14	1	0	7
c109	827.3	1	0	10	1	21	1	0	16
r101	1637.7	13	36	54	3	2	15	19	8
r102	1466.6	1	0	6	1	4	1	0	3
r103	1208.7	1	52	25	1	24	1	53	20
r104	971.5	3	291	645	3	32343	11	391	3103
r105	1355.3	3	144	48	5	43	3	144	36
r106	1234.6	3	145	61	1	75	3	144	87
r107	1064.6	5	421	542	3	1310	5	227	416
r108	932.1	1	331	905	1	5912	1	296	891
r109	1146.9	23	385	390	19	1432	65	588	1127
r110	1068.0	3	222	149	3	1068	5	219	426
r111	1048.7	7	312	1146	39	83931	111	736	5738
r112	948.6	11	683	10816	9	202804	19	574	16073
rc101	1619.8	1	69	24	1	12	1	87	19
rc102	1457.4	5	241	153	1	77	3	193	120
rc103	1258.0	13	294	949	3	2706	5	262	541
rc104	1132.3	28	302	5961	7	65807	21	437	11773
rc105	1513.7	1	67	25	1	27	1	79	33
rc106	1372.7	97	885	2958	37	15892	71	755	3916
rc107	1207.8	1	110	58	1	154	1	158	161
rc108	1114.2	1	129	132	1	3365	1	228	635
Average				866		14524			1562
c201	589.1	1	0	16	1	203	1	0	9
c202	589.1	1	0	36	1	3483	1	0	49
c203	588.7	1	0	128	1	13070	1	0	122
c204	588.1	1	0	557	-	-	1	0	6416
c205	586.4	1	0	27	1	417	1	0	15
c206	586.0	1	0	55	1	595	1	0	24
c207	585.8	1	0	79	1	1241	1	0	84
c208	585.8	1	0	106	1	555	1	0	26
r201	1143.2	1	60	114	1	139	1	52	78
r202	1029.6	11	149	1075	13	8282	17	152	1663
r203	870.8	3	60	1367	1	54187	1	78	641
r204	731.3	-	-	-	-	-	-	-	-
r205	949.8	17	942	14662	-	-	9	345	6904
r206	875.9	1	170	9688	-	-	1	171	60608
r207	794.0	1	30	12165	-	-	1	24	11228
r208	701.2*	-	-	-	-	-	-	-	-
r209	854.8	3	230	19999	3	78560	3	248	22514
r210	900.5	-	-	-	-	-	-	-	-
r211	746.7	-	-	-	-	-	-	-	-
rc201	1261.8	3	70	185	3	229	3	55	92
rc202	1092.3	1	40	126	1	313	1	39	89
rc203	923.7	1	50	604	1	14917	1	47	324
rc204	783.5	-	-	-	-	-	-	-	-
rc205	1154.0	1	50	145	1	221	1	32	111
rc206	1051.1	1	50	164	1	340	1	73	344
rc207	962.9	-	-	-	-	-	-	-	-
rc208	776.1	-	-	-	-	-	-	-	-
Average				3065		11047			5567