The robust vehicle routing problem with time windows: compact formulation and branch-price-and-cut method

Pedro Munari, Alfredo Moreno, Jonathan De La Vega,

Production Engineering Department, Federal University of Sao Carlos, Sao Carlos-SP, Brazil,
munari@dep.ufscar.br, alfredmorenoarteaga@gmail.com, jdlvmartinez@gmail.com

Douglas Alem,

Business School, University of Edinburgh, Scotland, UK, douglas.alem@ed.ac.uk,

Jacek Gondzio,

School of Mathematics, University of Edinburgh, Scotland, UK, J.Gondzio@ed.ac.uk,

Reinaldo Morabito

Production Engineering Department, Federal University of Sao Carlos, Sao Carlos-SP, Brazil,
morabito@ufscar.br

Technical Report 002/2018, Operations Research Group, DEP/UFSCar (05 Sep 2018)

Abstract

We address the robust vehicle routing problem with time windows (RVRPTW) under customer demand and travel time uncertainties. As presented thus far in the literature, robust counterparts of standard formulations have challenged general-purpose optimization solvers and specialized branch-and-cut methods. Hence, optimal solutions have been reported for small-scale instances only. Additionally, although the most successful methods for solving many variants of vehicle routing problems are based on the column generation technique, the RVRPTW has never been addressed by this type of method. In this paper, we introduce a novel robust counterpart model based on the well-known budgeted uncertainty set, which has advantageous features in comparison to other formulations and presents better overall performance when solved by commercial solvers. This model results from incorporating dynamic programming recursive equations into a standard deterministic formulation and does not require the classical dualization scheme typically used in robust optimization. In addition, we propose a branch-price-and-cut method based on a set partitioning formulation of the problem, which relies on a robust resource constrained elementary shortest path problem to generate routes that are robust regarding both vehicle capacity and customer time windows. Computational experiments using Solomons's instances show that the proposed approach is effective and able to obtain robust solutions within a reasonable running time. The results of an extensive Monte Carlo simulation indicate the relevance of obtaining robust routes for a more reliable decision-making process in real-life settings.

Keywords: vehicle routing; time windows; robust optimization; polyhedral-interval uncertainty; compact model; branch-price-and-cut.

1 Introduction

Vehicle routing is a cornerstone activity frequently associated with the decision-making process of many complex logistics systems and supply chains. As a result, academics and practitioners have devoted impressive efforts to develop models and powerful solution methods for the deterministic vehicle routing problem (VRP) over the past decades (Braekers et al., 2016; Toth and Vigo, 2014). However, much less attention has been paid to the VRP under uncertainty, mainly because providing an optimal solution for many deterministic variants of the problem is already computationally challenging and often impossible within an acceptable amount of time. In addition, defining a proper uncertainty modeling approach to reflect real-world concerns without unnecessarily increasing the difficulty of solving the problem is not trivial (Jaillet et al., 2016).

In the past few years, there has been a rapid increase in attention to vehicle routing under uncertainty, which has led to the proposition of new models and solution strategies (Oyola et al., 2016a; Jaillet et al., 2016) mainly focused on the classic variants, such as the capacitated vehicle routing problem (CVRP) and the vehicle routing problem with time windows (VRPTW). The vast majority of studies rely on stochastic programming (SP) techniques based on either recourse models or probabilistic/chance constraints to address uncertainty in demands and travel times, among others. Broadly speaking, the recourse models assume that a set of routes is chosen according to a given probability distribution of the stochastic parameters. In scenario-based formulations, the underlying probability is often used to generate a finite set of outcomes or scenarios. The first-stage design variables are thus "forced" to meet all the scenarios simultaneously, whereas the second-stage *recourse* aims at ensuring a feasible completion for each single realization. In chance constraint approaches, the so-called *probabilistic constraints* are supposed to hold with some probability or reliability level instead of guaranteeing feasibility for all realizations. For example, the minimum probability of meeting demands in a VRP should be greater than 0.95. The reader is referred to Oyola et al. (2016a,b) for a recent review on the stochastic VRP, including a comparison of different recourse decisions and specialized solution methods.

Defining reliable probability distributions may be difficult due to the lack of historical data associated with many real-life applications and/or to the excessive theoretical requirements for using the available scenario-generation methods. These drawbacks have been overcome over the past decade by using robust optimization (RO) techniques, the utmost goal of which is to provide solutions "immunized against uncertainty". In this paper, we consider a class of (strict) RO problems with bounded uncertainty in the sense of Ben-Tal and Nemirovski (1999), Bertsimas and Sim (2003) and Klamroth et al. (2017), in which the uncertain data belong to a convex set $\mathscr{U}$ known as the *uncertainty set* in RO terminology. We are interested in finding a *robust solution*, i.e., a solution that is deterministically feasible for every realization of the uncertain parameters that belongs to $\mathscr{U}$. The resulting model is optimized from a minimax or worst-case perspective according to the decision-maker preferences towards uncertainty. Even though it might not be trivial to evaluate the "shape" of the uncertainty set, polyhedral-type sets provide robust counterparts as tractable as their original formulations, which is particularly appealing to address combinatorial problems under uncertainty. The so-called budgeted uncertainty polytope proposed by Bertsimas and Sim (2003) is one of the most popular polyhedral-interval uncertainty sets. Despite all these potential advantages, only a few papers have used RO techniques to handle a VRP under uncertainty.

Especially in vehicle routing activities, RO approaches can play an important operational role in defining the routes for a particular day or even from a mid-term perspective, e.g., evaluating a proper fleet size or staff flexibility, when uncertainty matters. The practical implication of using the RO paradigm over the deterministic is ensuring that the planned routes will remain feasible with high probability even after the occurrence of atypical events. Obviously, if the price for that is overly conservative, one might prefer to adopt a contingency plan after knowing the uncertainty. Consequently, proposing RO techniques that strike the right balance between robustness and performance is of fundamental importance.

1.1 Related work

Sungur et al. (2008) were the first to apply robust optimization concepts in the CVRP to handle demand uncertainty. The authors proposed three robust versions of the two-index CVRP formulation with Miller-Tucker-Zemlin (MTZ) constraints based on convex hull, box, and ellipsoidal uncertainty sets. All the proposed models may be perceived as worst-case instances of the deterministic CVRP in which the nominal demand parameter is replaced by an augmented modified demand from the uncertainty sets. For this reason, the robust CVRP models provided overly conservative solutions in comparison to chance constraint and/or two-stage stochastic approaches. However, the two-stage model was significantly more difficult to solve than the robust versions with merely a small to medium number of scenarios. Following similar ideas, Ordonez (2010) developed some RO formulations based on bounded uncertainty to a class of VRP and discussed real-life applications in large-scale emergencies and courier deliveries.

On the other hand, Lee et al. (2012) used the idea of RO with bounded uncertainty to adjust the degree of conservativeness of the robust solution in a VRP with deadlines and both travel time and demand uncertainties. The paper presented a three-index vehicle flow formulation with MTZ constraints as well and derived the robust version for both uncertainties according to the popular budgeted uncertainty polytope. Due to the computational intractability of the resulting model, the authors proposed a set partitioning formulation and solved it via a branch-and-price method. The uncertainty sources were encapsulated into the shortest-path subproblem, which was solved by dynamic programming. The numerical results revealed a certain benefit of using RO, but even the branch-and-price method was not able to solve several instances with 25 customers within the time limit of one hour. The authors also noted that their algorithm cannot be directly applied to the VRPTW.

Agra et al. (2012) applied the idea of RO with bounded uncertainty to model a heterogeneous fleet VRPTW without capacity constraints under uncertain travel times using a formulation based on the concept of layered graphs. The resulting RO model was then applied to a ship routing and scheduling problem in a maritime transportation context consisting of small instances (10 and 20 request nodes). Computational experiments showed that a commercial optimization package was able to solve most of the instances within a time limit of one hour but failed in returning a feasible solution for a few of those with 20 nodes. Hence, the authors concluded that this approach is suitable to solve instances with up to 20 nodes only. Motivated by these results, Agra et al. (2013) developed two main approaches to handle the heterogeneous fleet VRPTW with uncertain travel times. The first one is based on the resource inequality formulation, in which uncertainty was incorporated by adjustable RO. The second approach uses static RO implicitly via the addition of canonical cuts. The computational results based on realistic instances from a ship routing and scheduling problem showed that the proposed methods successfully solved instances with 50 nodes within a reasonable running time.

Recently, De La Vega et al. (2018) applied RO to model demand uncertainty in the VRPTW with multiple deliverymen, a variant that includes the number of service workers per route as an endogenous decision of the problem (Pureza et al., 2012; Alvarez and Munari, 2017). The authors developed the robust counterpart problem based on a new deterministic formulation, which showed advantages in comparison to previous models in the literature. A construction heuristic was also devised to obtain initial robust feasible solutions, which were further used as starting points within a general-purpose optimization software. Computational experiments based on adapted Solomon's instances revealed that the proposed solution approach is suitable for finding robust feasible solutions for instances with up to 25 customers, but the optimality was effectively proven for only a few instances. The advantages of the RO approach in guiding decision making in the VRPTW with multiple deliverymen were highlighted by a Monte Carlo simulation.

Substantial contributions to modeling the robust CVRP with demand uncertainty were presented by Gounaris et al. (2013, 2016). In both efforts, customer demands were considered random variables within a generic polyhedron. Gounaris et al. (2013) developed exact robust counterparts of the deterministic CVRP based on well-known distinct formulations, namely, two-index vehicle flow, MTZ-based, one-commodity flow, and two-commodity flow. For each one, the authors established necessary and sufficient conditions to induce a robust feasible set of routes. The numerical results confirmed that all the proposed formulations led to equally robust solutions for two different uncertainty sets, but some of them had a clear advantage in terms of running time. Following the theoretical results from the previous study, Gounaris et al. (2016) focused on developing an adaptive memory programming metaheuristic for some robust CVRP under polyhedral demand uncertainty.

The CVRP with deadlines under travel time uncertainty was studied by Adulyasak and Jaillet (2016), who developed branch-and-cut and Benders decomposition algorithms for stochastic programming and RO versions of the problem. The SP model was based on a known probability distribution and its main focus was to minimize the expected number of deadline violations. The RO model was designed to be feasible for a family of probability distributions to minimize the so-called lateness index. The goal was to find a feasible route whose corresponding arrival times provide the minimum lateness index. Other variants of the VRP under uncertainty were also discussed. The computational study analyzed small instances defined on a directed graph only. The overall results revealed that both approaches can be competitive in terms of computational effort and solution quality under mild conditions. In a more recent contribution, Jaillet et al. (2016) proposed distributionally robust optimization approaches for a class of routing problems with uncertain travel times and demands to incorporate some information on the probability distributions of the random variables, thereby reducing the conservativeness of the robust solutions.

1.2 Our contributions

The contributions of this paper are threefold. First, we develop a novel compact RO formulation to address demand and travel time uncertainties. Instead of using the so-called dualization scheme to obtain the corresponding robust counterpart model, our formulation explicitly includes dynamic programming (DP) recursive equations in the two-index vehicle flow formulation of the VRPTW to evaluate the worst-case loads and earliest exact times for starting the service at the customers. The main advantage of this novel modeling paradigm hinges on the fact that the robust counterpart is straightforwardly obtained from the deterministic model without necessitating further, (often) more expensive reformulations, thereby making it more amenable to numerical solution schemes. Most robust VRP variants based on the dualization technique indeed had to introduce additional constraints and variables to encapsulate the uncertainties in such a way to end up with a less conservative formulation, e.g., Ordonez (2010); Gounaris et al. (2013); De La Vega et al. (2018). Importantly, although the results reported in this paper are obtained for the VRPTW, the proposed modeling approach based on DP might be applied not only to other VRP variants but also to a myriad of operations research problems under uncertainty, making them more tractable computationally and encouraging the usage of general-purpose optimization software.

Second, for the first time, we devise a branch-price-and-cut (BPC) method to solve a set partitioning formulation of the robust VRPTW (RVRPTW). We describe how to extend the labeling algorithm to solve the robust resource constrained shortest path problem. It involves redefining the labels and the dominance rule to cope with the uncertainties in demands and travel times simultaneously. As addressed in previous papers (Lee et al., 2012; Pessoa et al., 2015), the extensions regarding the travel times are not trivial because the starting time of time windows may absorb the worst-case deviations through the routes. The proposed BPC method relies on interior points of the feasible set to stabilize column and cut generation. The points are obtained by a primal-dual interior point method (Gondzio, 2012), which guarantees that they are well centered. These central points are also used for branching according to a central branching strategy. Comprehensive computational experiments on the well-known Solomon's instances reveal the effectiveness of the proposed BPC method for solving the instances with up to 100 customers within a plausible running time even when both uncertain parameters vary by up to 50% of their nominal values.

Third, we perform an extensive numerical study based on a Monte Carlo simulation to assess the relevance of the robust approach. For this purpose, we conduct a feasibility analysis to evaluate the empirical probability distribution of the constraint violation based on 10,000 simulation runs. In addition, we evaluate the *price of robustness* (Bertsimas and Sim, 2003) using the solutions provided by the BPC method. The results show that it is possible to obtain robust solutions at the expense of only a minor increase in the optimal values. In addition, it is not necessary to be over-conservative to achieve deterministically feasible solutions with a corresponding rather reasonable *price*, suggesting that in contrast to typical deterministic solutions, the RVRPTW is indeed appropriate to support decision making in an uncertain environment.

1.3 Organization of the paper

The remainder of this paper is structured as follows. Section 2 states the robust VRPTW under uncertain demand and travel time and establishes some conditions for the robustness of feasible routes. Section 3 presents a novel two-index compact formulation to cope with both sources of uncertainty simultaneously. Section 4 describes the set partitioning formulation and develops the BPC algorithm. Section 5 is focused on the computational results of the compact formulation and exact method. In addition, practical insights into the usage of the RO approach are discussed. Finally, concluding remarks and promising future research topics are provided in Section 6.

2 VRPTW via robust optimization

The VRPTW can be defined on a graph $G = (\mathcal{N}, \mathcal{E})$, in which $\mathcal{N} = \mathcal{C} \cup \{0, n+1\}$ is the set of nodes associated with customers in $\mathcal{C} = \{1, \ldots, n\}$ and with the depot nodes 0 and $n+1$. We use two nodes to represent the same single depot and impose that all routes must start at 0 and return to $n+1$. Set $\mathcal{E}$ contains all arcs (i, j) for each pair of nodes $i, j \in \mathcal{C}$, with $i \neq j$, and all arcs $(0, i)$ and $(i, n+1)$ for nodes $i \in \mathcal{C}$. The cost of crossing an arc $(i, j) \in \mathcal{E}$ is denoted by c_{ij}.

We assume an unlimited, homogeneous fleet of vehicles with capacity Q. Each node has a demand q_i, such that $q_i > 0$ for each $i \in \mathcal{C}$ and $q_0 = q_{n+1} = 0$. In addition, there is a time window $[w_i^a, w_i^b]$ for each node $i \in \mathcal{N}$ imposing that the service at this node cannot start earlier than the exact time w_i^a or later than w_i^b. If the vehicle arrives before w_i^a, then it has to wait until this instant to start servicing the node. To each arc $(i, j) \in \mathcal{E}$, we assign a travel time t_{ij}, which respects the triangle inequality. Additionally, each node i has a service time s_i that corresponds to the minimum amount of time that the vehicle remains at a visited node.

The problem consists of determining a set of minimal-cost routes that satisfies the following conditions: (i) Each route must start at the depot, visit a subset of customers and then return to the depot; (ii) the capacity of a vehicle is not violated; (iii) time windows are not violated; (iv) a customer is visited at most once by a route, and its demand is fulfilled by this visit; and (v) all customers are visited exactly once. We use $\mathcal{R}$ to denote the set of all feasible routes, given by the routes that satisfy conditions (i) to (iv). Any feasible solution $R = (r_1, \ldots, r_p)$ of the problem is given by a set of $p > 0$ feasible routes $r_j \in \mathcal{R}$, $j = 1, \ldots, p$, that together also satisfy condition (v).

In this paper, we address the RVRPTW assuming that the demands and travel times are uncorrelated uncertain values modeled as independent random variables that fall within the symmetric and bounded ranges given by $\tilde{q}_i \in [q_i - \hat{q}_i, q_i + \hat{q}_i]$ and $\tilde{t}_{ij} \in [t_{ij} - \hat{t}_{ij}, t_{ij} + \hat{t}_{ij}]$, respectively. Here, $\hat{q}$ and $\hat{t}$ are the vectors of positive deviations of the random variables from their corresponding nominal values q_i and t_{ij}. It is also common to rewrite the random variables as a linear combination of their nominal values and a normalized *scale deviation*, i.e., $\xi_i^q = (\tilde{q}_i - q_i)/\hat{q}_i$ and $\xi_{ij}^t = (\tilde{t}_{ij} - t_{ij})/\hat{t}_{ij}$, in which both scale deviations are themselves random variables in $[-1, 1]$. Consistent with most studies in the literature, the data uncertainty model is represented by the well-known *polyhedral uncertainty set* as follows:

$$\mathscr{U}^q = \left\{ \tilde{q} \in \mathbb{R}_+^{|\mathcal{C}|} \mid \tilde{q}_i = q_i + \hat{q}_i \xi_i^q,\ \sum_{i \in \mathcal{C}} \xi_i^q \leq \Gamma^q,\ 0 \leq \xi_i^q \leq 1,\ \forall i \in \mathcal{C} \right\}, \tag{2.1}$$

$$\mathscr{U}^t = \left\{ \tilde{t} \in \mathbb{R}_+^{|\mathcal{E}|} \mid \tilde{t}_{ij} = t_{ij} + \hat{t}_{ij} \xi_{ij}^t,\ \sum_{(i,j) \in \mathcal{E}} \xi_{ij}^t \leq \Gamma^t,\ 0 \leq \xi_{ij}^t \leq 1,\ \forall (i,j) \in \mathcal{E} \right\}, \tag{2.2}$$

where the cumulative uncertainty of each random variable is bounded by its corresponding *budget of* *uncertainty* Γ^q or Γ^t. If $\Gamma^q = \Gamma^t = 0$, then the uncertainties are not taken into account. On the other hand, larger budgets express more conservative/robust solutions. The precise value for the budgets can be evaluated via the probability of constraint violation or according to the decision-maker preferences towards uncertainty/risk. Both uncertainty sets assume that the decision variables are non-negative; thus, it follows that the worst case will always be achieved at the right-hand side of the range. For this reason, we assume ξ_i^q and ξ_{ij}^t in $[0,1]$ in (2.1) and (2.2) without loss of generality. Moreover, notice that further knowledge of the left-hand side of the range is not necessary under such an assumption and that the confidence interval does not have to be symmetric.

Let $r \in \mathcal{R}$ be a route that sequentially visits $k+1$ nodes, denoted by $(v_0, v_1, \cdots, v_k)$, where $v_0 = 0$ and $v_k = n+1$. Following the RO paradigm, this route is *robust feasible* if it satisfies conditions $(i)-(iv)$ for all possible demand realizations $\tilde{q} \in \mathscr{U}^q$ and travel time realizations $\tilde{t} \in \mathscr{U}^t$. We express the set of robust feasible routes as $\widetilde{\mathcal{R}}$. In fact, we are interested in the feasible routes that are simultaneously robust regarding demands and travel times. We say that a given solution $R = (r_1, \ldots, r_p)$ is robust feasible if all routes r_j, $j = 1, \ldots, p$, are robust feasible and if they together satisfy condition (v).

Due to the structure of sets (2.1) and (2.2), the robustness of a route can be defined explicitly using recursive equations (Lee et al., 2012; Agra et al., 2013). Let $u_{v_j\gamma}$ be the largest vehicle load at node v_j of the route, $j = 1, \ldots, k$, when up to $\gamma \leq \Gamma^q$ demand values attain their worst case. This value can be computed by the recursion:

$$u_{v_j\gamma} = \begin{cases} q_{v_0}, & \text{if } j = 0; \\ u_{v_{j-1}\gamma} + q_{v_j}, & \text{if } \gamma = 0; \\ \max\{u_{v_{j-1}\gamma} + q_{v_j},\ u_{v_{j-1},\gamma-1} + q_{v_j} + \hat{q}_{v_j}\}, & \text{otherwise.} \end{cases} \tag{2.3}$$

To be robust feasible, the route must satisfy $u_{v_j\gamma} \leq Q$ for all $\gamma = 0, 1, \ldots, \Gamma^q$ and $j = 1, \ldots, k$. Since $u_{v_j\gamma}$ is nondecreasing through the nodes in the path, we can simply check $u_{v_k\Gamma^q} \leq Q$. In addition, note that $u_{v_j\gamma}$ can be calculated alternatively using only the Γ^q largest demand deviations of the customers visited up to $u_{v_j\gamma}$, that is:

$$u_{v_j\gamma} = \sum_{i=1}^{j} q_{v_i} + \max_{\substack{\mathcal{I} \subseteq \{1,\ldots,j\} \\ |\mathcal{I}| \leq \gamma}} \sum_{i \in \mathcal{I}} \hat{q}_{v_i}. \tag{2.4}$$

Hence, the worst-case value is always achieved by taking the largest demand deviations of the visited customers.

Regarding the travel times, let $w_{v_j\gamma}$ be the earliest exact time from which the service can start at node v_j when up to $\gamma \leq \Gamma^t$ travel times reach their worst-case values. We can compute this value according to the following recursion:

$$w_{v_j\gamma} = \begin{cases} w_{v_0}^a, & \text{if } j = 0; \\ \max\{w_{v_j}^a,\ w_{v_{j-1}\gamma} + s_{v_{j-1}} + t_{v_{j-1}v_j}\}, & \text{if } \gamma = 0; \\ \max\{w_{v_j}^a,\ w_{v_{j-1}\gamma} + s_{v_{j-1}} + t_{v_{j-1}v_j}, & \\ \qquad w_{v_{j-1},\gamma-1} + s_{v_{j-1}} + t_{v_{j-1}v_j} + \hat{t}_{v_{j-1}v_j}\}, & \text{otherwise.} \end{cases} \tag{2.5}$$

To be robust feasible, the route must satisfy $w_{v_j\gamma} \leq w_{v_j}^b$ for all $\gamma = 0, 1, \ldots, \Gamma^t$ and $j = 1, \ldots, k$. It is enough to check $w_{v_j\Gamma^t} \leq w^b_{v_j}$, for $j = 1, \ldots, k$.

In contrast to the worst-case load, the value $w_{v_j\gamma}$ is not necessarily obtained by taking the Γ^t largest travel time deviations. Since the service can start only after the customer time window opens, the largest deviations can be absorbed by the waiting times. Indeed, Fig. 1 illustrates this situation for route $(v_0, v_1, v_2, v_3, v_4)$, with $v_0 = 0$ and $v_4 = n + 1$. The time windows are presented above their respective nodes, whereas travel times are shown below each arc, with their respective deviations inside parentheses. We assume that the service times of customers v_1, v_2 and v_3 are equal to 1. This route is deterministic feasible, as the time windows are not violated if we take the nominal values of travel times. For $\Gamma^t = 1$, we have that $w_{v_1 1} = 5$, which is achieved either because of the window opening time or because the travel time between v_0 and v_1 goes to its worst case. At time instant 6, the vehicle is ready to go from v_1 to v_2, and hence, we have $w_{v_2 1} = 12$, as the worst case now is attained when the travel time between v_1 and v_2 reaches its maximum value. Notice that in this example, the value of $w_{v_2 1}$ does not depend on the largest travel time deviation (i.e., between v_0 and v_1) but rather on the smallest one. For the next nodes, we have $w_{v_3 1} = 17$ and $w_{v_4 1} = 21$, as the worst case now is attained when the travel time between v_2 and v_3 reaches its maximum value. Again, these values are not obtained by taking the largest travel time deviations. This route is robust feasible for $\Gamma^t = 1$, but notice that if the time window of customer v_3 is replaced by $[11, 15]$, the route no longer remains robust feasible, although it is still deterministic feasible.

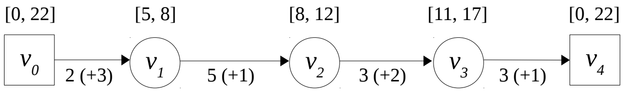


Fig. 1: Example of a route in which the worst-case earliest time is not determined by the largest deviations.

3 A two-index compact robust optimization formulation

In this section, we present a novel two-index vehicle flow formulation with extended MTZ constraints to cope with demand and travel time uncertainties simultaneously. As presented in other papers that addressed robust formulations of VRP variants thus far in the literature, the MTZ-based constraints make incorporating polyhedral uncertainty sets into the models difficult. Hence, several authors have proposed reformulations for the deterministic models to obtain more tractable robust counterpart formulations (Ordonez, 2010; Agra et al., 2012, 2013; Gounaris et al., 2013; De La Vega et al., 2018).

The formulation proposed below is a robust counterpart of the standard two-index vehicle flow formulation of the VRPTW (Irnich et al., 2014; Desaulniers et al., 2014) with the advantage of not requiring any reformulation of the original model. In addition, it provides a compact formulation without the need for defining an adversarial (auxiliary) problem, which is commonly used in the RO literature to encapsulate the uncertain parameter. Consequently, it is not necessary to apply the typical dualization technique to obtain the robust counterpart, which would result in a prohibitive number of additional decision variables and constraints, as noted by Lee et al. (2012) and Agra et al. (2013).

Different from previous studies in the field, this novel robust formulation is based on the linearization of the set of recursive equations defined in (2.3) and (2.5). Let x_{ij} be a usual binary decision variable that assumes the value of 1 if and only if a vehicle traverses the arc $(i,j) \in \mathcal{E}$ in the solution. For each $i \in \mathcal{N}$, let $u_{i\gamma}$ and $w_{i\gamma}$ be continuous variables having the same meaning as in (2.3) and (2.5) for $\gamma = 0, \dots, \Gamma^q$ and $\gamma = 0, \dots, \Gamma^t$, respectively. They are similar to the continuous variables used in the original formulation, but here they have the additional index γ to take uncertainty into account. For example, $u_{i\gamma}$ is the load in the vehicle that services node i when up to γ demand values attain their worst case. We can extend the MTZ-based constraints for including this worst-case load as follows:

$$u_{j\gamma} \geq u_{i\gamma} + q_j x_{ij} - Q(1 - x_{ij}),\ \forall (i,j) \in \mathcal{E},\ \gamma = 0, \dots, \Gamma^q, \tag{3.1}$$

$$u_{j\gamma} \geq u_{i\gamma-1} + (q_j + \hat{q}_j) x_{ij} - Q(1 - x_{ij}),\ \forall (i,j) \in \mathcal{E},\ \gamma = 1, \dots, \Gamma^q, \tag{3.2}$$

$$q_i \leq u_{i\gamma} \leq Q,\ \forall i \in \mathcal{C},\ \gamma = 0, \dots, \Gamma^q. \tag{3.3}$$

These constraints evaluate the cumulative vehicle load at customer $j \in \mathcal{C}$ if the vehicle is coming directly from customer i, according to γ demand values attaining their worst case. Constraints (3.1) and (3.3) are similar to the standard MTZ-based constraints used in the formulation of the deterministic VRPTW but are defined here for all index values $\gamma = 0, \cdots, \Gamma^q$. Constraints (3.2) are included to cope with the demand deviations, and they work together with (3.1) in a disjunctive way: (3.1) models the case in which the number of demand values attaining their worst case in node j is the same as that in node i, causing the load to increase by the nominal value q_j only, whereas (3.2) models the case in which the demand value at node j achieves its worst case $(q_j + \hat{q}_j)$ and then the number of demand values attaining their worst case at node j (γ) is one unit larger than that in node i $(\gamma - 1)$.

Following the same reasoning, we can obtain the robust counterpart of the MTZ-based constraints regarding the timing variables $w_{i\gamma}$, leading to the following robust two-index vehicle flow formulation of the RVRPTW:

$$\min \sum_{(i,j) \in \mathcal{E}} c_{ij} x_{ij} \tag{3.4}$$

$$\text{s.t.} \sum_{(i,j) \in \mathcal{E}} x_{ij} = 1,\ \forall j \in \mathcal{C} \tag{3.5}$$

$$\sum_{(i,j) \in \mathcal{E}} x_{ij} - \sum_{(j,i) \in \mathcal{E}} x_{ji} = 0,\ \forall i \in \mathcal{C} \tag{3.6}$$

$$\sum_{j:(0,j) \in \mathcal{E}} x_{0j} - \sum_{i:(i,n+1) \in \mathcal{E}} x_{i,n+1} = 0, \tag{3.7}$$

$$u_{j\gamma} \geq u_{i\gamma} + q_j x_{ij} - Q(1 - x_{ij}),\ \forall (i,j) \in \mathcal{E},\ \gamma = 0, \dots, \Gamma^q \tag{3.8}$$

$$u_{j\gamma} \geq u_{i\gamma-1} + (q_j + \hat{q}_j) x_{ij} - Q(1 - x_{ij}),\ \forall (i,j) \in \mathcal{E},\ \gamma = 1, \dots, \Gamma^q \tag{3.9}$$

$$q_i \leq u_{i\gamma} \leq Q,\ \forall i \in \mathcal{C},\ \gamma = 0, \dots, \Gamma^q \tag{3.10}$$

$$w_{j\gamma} \geq w_{i\gamma} + (s_i + t_{ij}) x_{ij} - M_{ij}(1 - x_{ij}),\ \forall (i,j) \in \mathcal{E},\ \gamma = 0, \dots, \Gamma^t \tag{3.11}$$

$$w_{j\gamma} \geq w_{i\gamma-1} + (s_i + t_{ij} + \hat{t}_{ij}) x_{ij} - M_{ij}(1 - x_{ij}),\ \forall (i,j) \in \mathcal{E},\ \gamma = 1, \dots, \Gamma^t \tag{3.12}$$

$$w_i^a \leq w_{i\gamma} \leq w_i^b,\ \forall i \in \mathcal{N},\ \gamma = 0, \dots, \Gamma^t \tag{3.13}$$

$$x_{ij} \in \{0,1\}, \ \forall (i,j) \in \mathcal{E}. \tag{3.14}$$

The objective function (3.4) consists of minimizing the overall transportation costs. Flow constraints (3.5)−(3.7) ensure that each customer $i \in \mathcal{C}$ is visited by exactly one vehicle and that all vehicles depart from and return to the depot. Constraints (3.8)–(3.13) are the robust MTZ-based constraints described previously based on the linearization of the recursive equations (2.3) and (2.5). Finally, constraints (3.14) impose the domain of the x_{ij} variables. Notice that, in comparison with the standard two-index vehicle flow formulation of the deterministic VRPTW (Irnich et al., 2014), the proposed RO formulation has only $(\Gamma^q + \Gamma^t) \times |\mathcal{N}|$ additional variables and two additional sets of constraints, (3.9) and (3.12), which are similar to those already available in the deterministic model.

We believe that the proposed formulation has a practical appeal, as it is a compact model and an intuitive extension of the deterministic formulation, thereby encouraging its usage by non-expert practitioners, especially those relying on general-purpose optimization software. In addition, as the results of computational experiments presented in Section 5 indicate, it is possible to solve most instances with 25 customers to optimality and find feasible solutions for almost all instances with 100 customers using this formulation within one hour of running time. Notably, the other compact models proposed thus far in the literature for the RVRPTW and related variants are not effective for instances with more than 10 or 20 customers, as reported by Lee et al. (2012) and Agra et al. (2012) and supported by our computational experiments presented in Section 5.1.

4 Set partitioning formulation and branch-price-and-cut method

Currently, the most effective exact methods for solving VRP variants are based on the set partitioning formulation (Pecin et al., 2016, 2017). The variables in this formulation correspond to feasible routes of the problem. As previously defined in Section 2, let $\widetilde{\mathcal{R}}$ be the set of all robust feasible routes regarding vehicle capacity and customer time windows. Let λ_r be a binary decision variable that is equal to 1 if and only if the route $r \in \widetilde{\mathcal{R}}$ is selected in the optimal solution. The set partitioning formulation for the RVRPTW is as follows:

$$\min \quad \sum_{r \in \widetilde{\mathcal{R}}} c_r \lambda_r \tag{4.1}$$

$$\text{s.t.} \quad \sum_{r \in \widetilde{\mathcal{R}}} a_{ri} \lambda_r = 1, \quad i \in \mathcal{C}, \tag{4.2}$$

$$\lambda_r \in \{0,1\}, \qquad r \in \widetilde{\mathcal{R}}. \tag{4.3}$$

Objective function (4.1) consists of minimizing the total cost of the selected routes. The cost of a route $r \in \widetilde{\mathcal{R}}$, denoted by c_r, is computed straightforwardly using the arc costs c_{ij} of each arc (i,j) in the path. Constraints (4.2) impose exactly one visit to each customer node. We assume that all routes are elementary (i.e., without any cycle), so that each column $a_r = (a_{r1}, \ldots, a_{rn})^T$ is a binary vector in which $a_{ri} = 1$ if and only if the corresponding route r visits customer i for each $i \in \mathcal{C}$. This is the standard set partitioning formulation, which can be used to model different VRP variants (Poggi and Uchoa, 2014; Desaulniers et al., 2014) using an appropriate definition of the set of feasible routes.

Generating all feasible routes to define the set partitioning formulation explicitly is impractical in general, as a huge number of routes is typically available. Hence, we recur to the column generation (CG) technique for solving the linear relaxation of model (4.1)–(4.3) (Lübbecke and Desrosiers, 2005; Vanderbeck and Wolsey, 2010). To obtain optimal integer solutions, CG is combined with a branch-and-bound tree, leading to a branch-and-price method. Since valid inequalities are also employed, we obtain a branch-price-and-cut (BPC) method (Desrosiers et al., 2010). In the remainder of this section, we develop a BPC method for the RVRPTW and describe the changes required in the CG subproblem to generate robust feasible solutions.

4.1 Column generation

In the CG technique, we start with a relatively small subset of routes $\overline{\mathcal{R}} \subset \widetilde{\mathcal{R}}$ that leads to the initial columns of the so-called restricted master problem (RMP), which corresponds to replacing $\widetilde{\mathcal{R}}$ with $\overline{\mathcal{R}}$ in the linear relaxation of the set partitioning formulation (4.1)–(4.3). In this context, the original set partitioning formulation is called the master problem (MP). Since the RMP has only a subset of the columns of the MP formulation, its optimal solution is also feasible for the linear relaxation of the MP but is not necessarily optimal. To check its optimality for the linear relaxation of the MP, we recur to the pricing subproblem (oracle). Let $u = (u_1, \ldots, u_n)$ be the vector of dual variables associated with constraints (4.2) in the RMP. At each iteration of the CG algorithm, we solve the RMP to obtain a dual solution $\overline{u}$, where $\overline{u}_0 := 0$, which is then checked in the following subproblem:

$$rc(\overline{u}) := \min \left\{ \sum_{(i,j)\in\mathcal{E}} (c_{ij} - \overline{u}_i) x_{rij} \mid r \in \widetilde{\mathcal{R}} \right\}, \tag{4.4}$$

where $x_r = (x_{rij})_{(i,j)\in\mathcal{E}}$ is a binary vector such that $x_{rij} = 1$ if and only if route $r \in \widetilde{\mathcal{R}}$ traverses arc (i, j); i.e., it visits node i and goes directly to node j. The optimal value of $rc(\overline{u})$ gives the minimum reduced cost among all columns in the MP, including those that are not in the RMP. Hence, if $rc(\overline{u})$ is negative, we add (at least) one new column to the RMP and repeat the process. Otherwise, the current optimal solution of the RMP is also optimal for the linear relaxation of the MP, and the CG algorithm terminates successfully.

Recall that the columns in the set partitioning formulation correspond to routes. Therefore, the subproblem (4.4) is a *resource constrained elementary shortest path problem* (RCESPP) (Irnich and Desaulniers, 2005). Any solution $\bar{x}_r$ of the subproblem describes a route, which is converted to a column as follows:

$$c_r := \sum_{(i,j)\in\mathcal{E}} c_{ij} \bar{x}_{rij},$$

$$a_{ri} := \sum_{(i,j)\in\mathcal{E}} \bar{x}_{rij}, \; i \in \mathcal{C}.$$

There are different strategies to solve the RCESPP, and one of most effective currently is the labeling algorithm (Righini and Salani, 2008; Pugliese and Guerriero, 2013; Pecin et al., 2017). It is a DP algorithm based on the gradual extension of feasible paths through the network while keeping a list of labels to represent each non-dominated partial path. Dominance rules play an important role in avoiding full enumeration of all possible routes. In addition, a few other clever techniques have been proposed to improve the computational performance of the implementations (Feillet et al., 2004; Righini and Salani, 2008; Chabrier, 2006; Desaulniers et al., 2008; Baldacci et al., 2011; Contardo et al., 2015). In its standard definition, the labeling algorithm can generate routes for the deterministic RCESPP only. To guarantee that only robust feasible routes are generated, we have to change the algorithm to take into account the uncertainty sets of demands and travel times simultaneously. This consideration is addressed in the following section, based on previous developments for related VRP variants (Lee et al., 2012; Pessoa et al., 2015; Errico et al., 2016).

4.1.1 Labeling algorithm for the robust RCESPP

In the standard labeling algorithm, each partial path from the depot to a given node $i \in \mathcal{N}$ is represented by a label $L_i = (C_i, V_i, R_i^q, R_i^t)$, where C_i is the total reduced cost of the path, V_i is the set containing all customers visited by the path, R_i^q is the total vehicle load up to node i and R_i^t is the earliest time at which service can start at node i. Starting from node 0, a label is initially defined with all its components equal to zero except when the time window at the depot opens at $w_0^a > 0$, as $R_0^t = w_0^a$. Then, the algorithm iteratively extends each label L_i in a given node $i \in \mathcal{C} \cup \{0\}$ to each non-visited node j such that $(i,j) \in \mathcal{E}$, generating a new label L_j with values:

$$\begin{aligned} C_j &= C_i + c_{ij} - \overline{u}_i; \\ V_j &= V_i \cup \{j\}; \\ R_j^q &= R_i^q + q_j; \\ R_j^t &= \max\{w_j^a, R_i^t + s_i + t_{ij}\}. \end{aligned}$$

The extension is carried out only if the path corresponding to the resulting label L_j is feasible, that is, $j \notin V_i$, $R_j^q \leq Q$ and $R_j^t \leq w_j^b$.

The definition presented above does not take into account the uncertainty in demands and travel times. Hence, for the robust RCESPP, we redefine the label at node i as $\widetilde{L}_i = (C_i, V_i, R_{i0}^q, \ldots, R_{i\Gamma^q}^q, R_{i0}^t, \ldots, R_{i\Gamma^t}^t)$. Components C_i and V_i are defined as in the deterministic case, whereas the total vehicle load and earliest time to start servicing the node are now defined for each value $\gamma = 0, \ldots, \Gamma^q$ and $\gamma = 0, \ldots, \Gamma^t$, respectively. For a given γ, $R_{i\gamma}^q$ is the maximum total vehicle load at node i when the demands of up to γ customers attain their worst case. Similarly, $R_{i\gamma}^t$ is the worst-case earliest time for the service start time at node i, considering that up to γ travel times attain their maximum values. Notice that $R_{i0}^q = R_i^q$ and $R_{i0}^t = R_i^t$. With the recursive equations defined in (2.3) and (2.5), an extension of label L_i to a non-visited node j, such that $(i,j) \in \mathcal{E}$, leads to a label L_j computed as:

$$\begin{aligned} C_j &= C_i + c_{ij} - \bar{u}_i; && (4.5) \\ V_j &= V_i \cup \{j\}; && (4.6) \\ R_{j\gamma}^q &= \max\{R_{i\gamma}^q + q_j,\ R_{i\gamma-1}^q + q_j + \hat{q}_j\}; && (4.7) \\ R_{j\gamma}^t &= \max\{w_j^a,\ R_{i\gamma}^t + s_i + t_{ij},\ R_{i\gamma-1}^t + s_i + t_{ij} + \hat{t}_{ij}\}. && (4.8) \end{aligned}$$

The resulting label is accepted as a feasible extension only if $j \notin V_i$, $R^q_{j\gamma} \leq Q$, for $\gamma = 0, \ldots, \Gamma^q$; and $R^t_{j\gamma} \leq w^b_j$, for $\gamma = 0, \ldots, \Gamma^t$. We can define an extension in a more general way as a sequence of single-arc extensions through a subset of nodes. We use $e = (e_1, \ldots, e_k)$ to denote an extension with k nodes of a given label L_i, which means that we apply (4.5)–(4.8) sequentially for each arc (e_j, e_{j+1}), where $j = 0, \ldots, k-1$ and $e_0 := i$. We use $E(L_i)$ to denote the set of all possible feasible extensions of a given label L_i.

As already mentioned, dominance rules play a crucial role in the labeling algorithm, as they may significantly reduce the number of extensions required to find an optimal solution. Given two labels, e.g., L^1_i and L^2_i, corresponding to paths that end at node i, we say that L^1_i dominates L^2_i if any feasible extension e of L^2_i ending at a given node j is also feasible for L^1_i and satisfies $C^1_j \leq C^2_j$. In such a case, L^2_i can be eliminated, as it cannot lead to a path with a better reduced cost than L^1_i. In the deterministic RCESPP, given a feasible extension $e \in E(L^2_i)$, we can verify if e also belongs to $E(L^1_i)$ by checking the following conditions: $V^1_i \subset V^2_i$, $R^{q1}_i \leq R^{q2}_i$ and $R^{t1}_i \leq R^{t2}_i$. Proposition 4.1 shows that similar properties can be used for the robust RCESPP to verify dominance between two given labels.

Proposition 4.1: (Dominance rule for the robust RCESPP) Let L^1_i and L^2_i be two labels associated with paths ending at node i. L^1_i dominates L^2_i if the following conditions hold:

(i) $C^1_i \leq C^2_i$;

(ii) $V^1_i \subset V^2_i$;

(iii) $R^{q1}_{i\gamma} \leq R^{q2}_{i\gamma}$, $\forall \gamma = 0, 1, \ldots, \Gamma^q$;

(iv) $R^{t1}_{i\gamma} \leq R^{t2}_{i\gamma}$, $\forall \gamma = 0, 1, \ldots, \Gamma^t$.

Proof. We need to show that, under hypotheses (i)–(iv), any feasible extension of L^2_i is also feasible for L^1_i and leads to a path with smaller or equal reduced cost. Let $e \in E(L^2_i)$ be a robust feasible extension of the label L^2_i that visits $k > 0$ nodes and ends at node j, resulting in label L^2_j. Let $e_1, \ldots, e_k$ denote the nodes in the extension e, where $e_k = j$. In addition, we define $e_0 := i$. Since e is a robust feasible extension of L^2_i, for each $h = 1, \ldots, k$, we have $e_h \notin V^2_{e_{h-1}}$; $R^{q2}_{e_h\gamma} \leq Q$, for all $\gamma = 0, 1, \ldots, \Gamma^q$; and $R^{t2}_{e_h\gamma} \leq w^b_{e_h}$, for all $\gamma = 0, 1, \ldots, \Gamma^t$. Let L^1_j be the label resulting from applying extension e to label L^1_i. Since $V^1_i \subset V^2_i$, we also have $e_h \notin V^1_{e_{h-1}}$, for each $h = 1, \ldots, k$, so the path associated with L^1_j is also elementary. Now, notice that due to condition (iii) and (4.7), we have that for all $\gamma = 1, \ldots, \Gamma^q$:

$$\begin{aligned} R^{q1}_{e_1\gamma} &= \max\{R^{q1}_{e_0\gamma} + q_{e_1},\ R^{q1}_{e_0\gamma-1} + q_{e_1} + \hat{q}_{e_1}\} \\ &\leq \max\{R^{q2}_{e_0\gamma} + q_{e_1},\ R^{q2}_{e_0\gamma-1} + q_{e_1} + \hat{q}_{e_1}\} \\ &= R^{q2}_{e_1\gamma}. \end{aligned}$$

By sequentially applying the same reasoning for any two nodes e_{h-1} and e_h, we obtain that $R^{q1}_{e_h\gamma} \leq R^{q2}_{e_h\gamma}$ for all $h = 1, \ldots, k$, and hence, $R^{q1}_{e_h\gamma} \leq Q$ for all $\gamma = 0, 1, \ldots, \Gamma^q$. In a similar way, we obtain that $R^{t1}_{e_h\gamma} \leq w^b_{e_h}$ for all $h = 1, \ldots, k$ and $\gamma = 0, 1, \ldots, \Gamma^t$. Therefore, e is also a robust feasible extension for label L_i^1. By analyzing the reduced costs of the resulting labels, we obtain:

$$\begin{aligned} C_j^1 &= C_{e_0}^1 + \sum_{h=1}^{k} (c_{e_{h-1}e_h} - \bar{u}_{e_{h-1}}) \\ &\leq C_{e_0}^2 + \sum_{h=1}^{k} (c_{e_{h-1}e_h} - \bar{u}_{e_{h-1}}) \\ &= C_j^2, \end{aligned}$$

which completes the proof. □

Several strategies have been used to improve the performance of the labeling algorithm. Feillet et al. (2004) propose identifying a set of unreachable nodes regarding vehicle capacity and customer time windows. Customers that are unreachable can be considered visited in the dominance checking, increasing the number of dominated labels. The same reasoning can be applied to the robust RCESPP, and in particular, the worst-case deviations regarding demands and travel times can be used to check for unreachable nodes. Bi-directional extension is another significant improvement in the labeling algorithm (Righini and Salani, 2008). In addition to the forward extension described above, a backward extension is also carried out, and both extensions stop at the middle point of the depot time window. A backward extension of the robust RCESPP can be defined very similarly to the forward extension described above, so we omit the details from this paper. We have incorporated all these techniques in our BPC method, including all the heuristic strategies proposed by Desaulniers et al. (2008) to speed up the labeling algorithm.

4.1.2 Interior point stabilization

An important issue in the CG algorithm concerns the stability of the dual solutions of the RMPs. It is widely known in the literature that optimal dual solutions obtained by the simplex method typically oscillate unstably between consecutive iterations of the CG algorithm (Vanderbeck, 2005; Lübbecke and Desrosiers, 2005; Munari and Gondzio, 2015). This leads to relatively slow progress of the algorithm, especially in the initial and in the final iterations (heading-in and tailing-off effects). In addition, degeneracy in the RMP may negatively affect the CG algorithm if we rely on the simplex method.

Stabilization techniques try to overcome these drawbacks by controlling the oscillation of dual solutions, which has been shown to be successful for a variety of problems (Briant et al., 2008; Ben Amor et al., 2009; Gondzio et al., 2013). We rely on the technique known as the primal-dual column generation method (Gondzio et al., 2013; Munari and Gondzio, 2013; Gondzio et al., 2016), which uses a primal-dual interior point algorithm to naturally control the distance between dual solutions. Each RMP is solved by the interior point algorithm within a given optimality tolerance $\varepsilon > 0$ that is dynamically adjusted at each CG iteration according to the relative gap in the CG method. More specifically, let $gap = (\mathrm{UB} - \mathrm{LB})/(1 + |\mathrm{UB}|)$ be the relative gap at a given iteration of the CG method, where UB is the smallest objective function value of the RMP in previous iterations and LB is the largest Lagrangian bound obtained from previous dual solutions. Then, we solve the current RMP by the interior point method using the optimality tolerance $\varepsilon = gap/D$, where $D > 1$ is a fixed scaling parameter. Hence, the dual solutions sent to the subproblem are suboptimal solutions that are well centered in the feasible set, as the primal-dual algorithm keeps them inside a neighborhood of the central path (Gondzio, 2012). The dynamic adjustment of the tolerance ε guarantees the convergence of the method and brings about significant gains in relation to the standard CG method, in particular for VRPs. As presented in the literature, in comparison to the use of optimal dual solutions, the use of well-centered, suboptimal dual solutions obtained by an interior point method has delivered significant reductions in computational times, particularly for the VRPTW (Munari and Gondzio, 2013). For further details on this, please see Gondzio et al. (2013); Munari and Gondzio (2015).

4.2 Valid inequalities

Subset row (SR) inequalities (Jepsen et al., 2008) are currently the most powerful valid inequalities for the set partitioning formulation. Although they can adversely affect the performance of the labeling algorithm, the lower bound provided by the linear relaxation of the MP becomes stronger, typically leading to a significant reduction in the number of nodes in the BPC method. Therefore, the SR inequalities are included in the state-of-the-art implementations of BPC algorithms for many VRP variants (Desaulniers et al., 2014; Pecin et al., 2016). In our method, we use the SR inequalities defined by subsets of three customers and stated as follows. Consider a subset $\mathcal{S} = \{i_1, i_2, i_3\} \subset \mathcal{C}$. The corresponding SR inequality must ensure that the number of routes that visit at least two nodes in $\mathcal{S}$ is at most equal to one. This is imposed by adding the following constraint to the RMP:

$$\sum_{r \in \mathcal{I}_{\mathcal{S}}} \lambda_r \leq 1,$$

where $\mathcal{I}_{\mathcal{S}} \subset \overline{\mathcal{R}}$ is the set of all columns that are associated with routes that visit at least two nodes in $\mathcal{S}$.

The separation of SR inequalities is carried out by a straightforward enumeration that checks all the columns in the RMP for each possible subset $\mathcal{S}$ of three nodes. We call this separation procedure inside the CG process as soon as the CG relative gap falls below a fixed tolerance ε_c (set to 0.1 in our experiments). Since we are using an interior point algorithm to solve the RMPs (see Section 4.1.2), the separation uses central primal (feasible) solutions to generate the valid inequalities, which can lead to deeper cuts and hence reduce the total number of generated inequalities (Munari and Gondzio, 2013). To reduce the negative impact of SR inequalities on the performance of the labeling algorithm, the separation procedure is called every three iterations of the CG method, and we add up to three cuts at each call, given by the most violated ones. In addition, as proposed by Desaulniers et al. (2008), a customer can be in a set S at most five times, and we add only the valid inequalities with violations greater than 0.1.

4.3 Branching

Once the CG procedure terminates with a solution $\overline{\lambda}$, we compute the corresponding arc solution $\overline{x} = (\overline{x}_{ij})_{(i,j)\in\mathcal{E}}$ given by $\overline{x}_{ij} = \sum_{r\in\overline{\mathcal{R}}} x_{rij}\overline{\lambda}_r$, where x_r is the binary vector corresponding to the rth route obtained from the subproblem, as defined in Section 4.1. If the components $\overline{x}_{ij}$ are integers for all $(i,j) \in \mathcal{E}$, then we prune the node with an integer solution of the problem. Otherwise, we apply the following branching strategy:

- If $\alpha := \sum_{r\in\overline{\mathcal{R}}} \overline{\lambda}_r$ is not an integer, then we create two child nodes, imposing the constraint $\sum_{r\in\overline{\mathcal{R}}} \lambda_r \leq \lfloor\alpha\rfloor$ to the master problem of one node and the constraint $\sum_{r\in\overline{\mathcal{R}}} \lambda_r \geq \lceil\alpha\rceil$ to the master problem of the other;
- Otherwise, we select the component $\overline{x}_{ij}$ with most fractional value (ties are broken randomly), and then we create two child nodes: in one node, we impose $x_{ij} = 0$ by removing the arc (i, j) from the network and dropping from the master problem all columns corresponding to routes that do not satisfy this requirement; on the other node, we impose $x_{ij} = 1$ by removing all arcs (i, j') and (i', j) such that $i' \neq i$ and $j' \neq j$, whereas in the master problem, we also drop all the columns that do not satisfy this constraint.

Additionally, we rely on a central branching strategy based on the suboptimal, well-centered solutions provided by the interior point algorithm used to solve the RMPs. It consists of terminating the CG procedure as soon as the relative gap falls below a threshold value ε_b (equal to 10^{-3} in our implementation) and then applying the branching strategy described before. If the solution suggests that the node should be pruned, then we resume the CG procedure with the standard optimality tolerance (10^{-6}) and apply the branching rules to the new solution. This strategy has been used for other VRP variants and leads to reductions in the total number of nodes in the BPC tree, as well as reducing the time spent at each node (Munari and Gondzio, 2013; Munari and Morabito, 2016; Alvarez and Munari, 2017).

4.4 MIP heuristic

To quickly identify integer feasible solutions of the problem, we rely on a mixed-integer programming (MIP) heuristic based on the columns of the MP. At the end of each node, right before branching, we impose integrality to all variables in the RMP of the last CG iteration. Then, to solve the resulting MIP problem, we run a general-purpose optimization software up to a given time limit (set to 30 seconds in our experiments). Despite its simplicity, this heuristic works well on VRPs, as it typically identifies feasible solutions much earlier than approaches relying only on the BPC tree (Alvarez and Munari, 2017; Munari and Gondzio, 2013; Joncour et al., 2010).

5 Computational results

In this section, we report on the computational performance of the proposed RVRPTW approaches. Our specific goals are threefold. The first goal is studying the behavior of the novel compact two-index formulation based on recursive equations via a general-purpose optimization software. The second goal relies on evaluating the efficiency of the BPC method in providing optimal or good-quality solutions within a plausible running time. The last goal is investigating the quality of the robust solutions in terms of feasibility and price of robustness. We summarize the results for each goal in the following subsections, while detailed results can be found in the supplementary material available at `http://www.dep.ufscar.br/docentes/munari/rvrptw`.

We use benchmark instances proposed by Solomon (1987). These instances include 100 customers and are classified according to the spatial distribution of customers: classes C1 and C2 follow a clustered distribution, R1 and R2 follow a random distribution, and RC1 and RC2 follow a mixture of both distributions. In addition, instances in classes C2, R2 and RC2 have wider time windows and larger vehicle capacities than the other classes, so they are more challenging for BPC methods. The demands and travel times of the instances were used as nominal values of the corresponding parameters in the definition of the uncertainty sets (2.1) and (2.2).

To incorporate uncertainty into Solomon's instances, we defined $\hat{q}_i = trunc(\alpha^q \times q_i)$ and $\hat{t}_{ij} = 0.1 \times trunc(\alpha^t \times 10 \times t_{ij})$, in which α^q and α^t belong to $\{0.1, 0.25, 0.5\}$. We truncate the resulting deviations in order to avoid numerical errors. Travel times are truncated after the first decimal place to avoid perturbations equal to zero and to follow a conventional practice in the VRPTW literature. Preliminary experiments suggest selecting both budgets of uncertainty as $\Gamma^q(\Gamma^t) = 0, 1, 5, 10$.

The two-index formulation was implemented using the Concert library of the IBM CPLEX Optimization Studio 12.7 and solved by the standard optimization solver of this package. The BPC method was coded in C++ using the interior point BPC framework proposed by Munari and Gondzio (2013), which relies on the interior point code HOPDM for solving the RMPs. The optimization solver of IBM CPLEX 12.7 was also used within the BPC method but only for the MIP heuristic. All the experiments were run on a Linux PC with an Intel Xeon E5-2680 2.7 GHz processor and 192 GB of memory.

5.1 Computational performance of the robust two-index formulation

This section presents the results of the proposed compact robust two-index vehicle flow formulation using the general-purpose optimization software IBM CPLEX version 12.7. We have run the model using combinations of values for the parameters Γ^q and Γ^t in the set $\{0, 1, 5, 10\}$ and α^q and α^t in the set $\{0, 0.1, 0.25, 0.5\}$. Hence, each instance was solved 28 times: using $\Gamma^q = \Gamma^t = 0$ and $\alpha^q = \alpha^t = 0$ (deterministic case); using $\Gamma^q = 1, 5, 10$ and $\Gamma^t = 0$ for each $\alpha^q = 0.1, 0.25, 0.5$ and $\alpha^t = 0$; using $\Gamma^q = 0$ and $\Gamma^t = 1, 5, 10$ for each $\alpha^q = 0$ $\alpha^t = 0.1, 0.25, 0.5$; and using $\Gamma^q = \Gamma^t = 1, 5, 10$ for each $\alpha^q = \alpha^t = 0.1, 0.25, 0.5$. A maximum running time limit of 3,600 seconds was imposed in the experiments.

In the first experiment with the compact formulation, we consider only the 25 first customers of the instances (experiments with 50 and 100 customers are described later in this section). This allows the optimization software to solve most of these instances to optimality and hence leads to more interesting analyses. Additionally, previous studies on compact models for the robust VRP and related variants have used instances of a similar scale. Table 1 shows the average results grouped according to the different values of the budgets of uncertainty Γ^q and Γ^t for instances in classes C1, R1 and RC1 with 25 customers. For each pair of values for Γ^q and Γ^t (referred as a group), we present the average results over all instances solved using the corresponding choice of values for these parameters, as specified in the first two columns of the table. Then, for each instance class (C1, R1 and RC1), the table shows the average objective value of the solutions (*Obj*), the average relative gap (*Gap*) in percentage, the average elapsed time (*Time*), the total number of instances in the group (*Ins*), the total number of instances solved to optimality (*Opt*) within the time limit of 3,600 seconds, and the number of infeasible instances (*Inf*). The last row of each table gives the average (arithmetic mean) over the results of all groups. The gap of a solution was taken from the output given by CPLEX. When all instances are solved to optimality in a group, we use the symbol '$-$' to express that the gap is zero. The number of instances solved in groups with $\Gamma^q > 0$ and/or $\Gamma^t > 0$ is three times the number of instances solved in the deterministic group

($\Gamma^q = \Gamma^t = 0$) due to the three different positive deviation values 0.1, 0.25 and 0.5. Table 2 shows the results using instances in classes C2, R2 and RC2, and its columns follow the same structure.

In Tables 1 and 2, we observe that as the budgets of uncertainty and the deviations increase, the objective values of the solutions also increase. This is expected, as these solutions are now immunized against the uncertainties, according to these parameters. The increase in the objective values is justified if the risk of the solution becoming infeasible is reduced. An extensive analysis of this behavior is carried out in Section 5.3. Notice that the running time for solving the robust model can be significantly larger than the time required for solving the deterministic model, especially for larger budgets and deviations, which illustrates the difficulty in solving RO models in comparison to their deterministic counterparts. Nevertheless, the results presented in the tables indicate the effectiveness of the proposed model for instances with 25 customers.

Tab. 1: Average results of the robust two-index model with different values of budgets of uncertainty using instances of classes C1, R1 and RC1 with 25 customers.

		C1						R1						RC1					
Γ^q	Γ^t	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Inf*	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Inf*	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Inf*
0	0	190.59	–	0.89	9	9	0	463.37	–	64.17	12	12	0	350.24	–	4.75	8	8	0
1	0	202.68	–	2.70	27	27	0	463.37	–	94.67	36	36	0	360.14	0.014	619.17	24	20	0
5	0	238.50	0.011	551.00	27	24	0	463.37	–	200.61	36	36	0	432.36	0.121	2,457.13	24	9	0
10	0	239.11	0.020	659.30	27	23	0	463.37	0.003	435.06	36	33	0	438.27	0.132	2,687.21	24	8	0
0	1	192.34	–	0.93	27	27	0	470.33	–	66.81	36	32	4	356.87	–	10.13	24	23	1
0	5	195.83	–	2.63	27	27	0	478.16	0.002	317.28	36	31	4	382.91	0.023	526.65	24	20	1
0	10	195.83	–	5.89	27	27	0	478.49	0.005	422.63	36	30	4	392.57	0.037	744.22	24	19	1
1	1	202.86	–	3.93	27	27	0	470.33	–	139.97	36	32	4	368.53	0.014	583.65	24	20	1
5	5	238.93	0.013	598.52	27	23	0	478.19	0.006	395.44	36	30	4	444.25	0.136	2,492.00	24	8	1
10	10	239.69	0.018	716.67	27	23	0	478.48	0.011	652.59	36	28	4	446.52	0.150	2,650.48	24	8	1
Average		213.64	0.006	254.24				470.75	0.003	278.92				397.26	0.063	1,277.54			

Tab. 2: Average results of the robust two-index model with different values of budgets of uncertainty using instances in classes C2, R2 and RC2 with 25 customers.

		C2						R2						RC2					
Γ^q	Γ^t	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Inf*	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Inf*	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Inf*
0	0	214.45	–	3.50	8	8	0	382.15	–	17.36	11	11	0	319.28	0.110	1,464.25	8	5	0
1	0	214.45	–	7.46	24	24	0	382.15	–	27.82	33	33	0	319.28	0.116	1,700.25	24	15	0
5	0	214.45	–	22.29	24	24	0	382.15	–	72.91	33	33	0	319.28	0.128	1,838.00	24	12	0
10	0	214.45	–	28.75	24	24	0	382.15	–	147.06	33	33	0	319.28	0.141	1,859.96	24	12	0
0	1	214.51	–	9.04	24	24	0	383.86	–	26.42	33	33	0	319.60	0.115	1,595.58	24	15	0
0	5	214.57	–	17.83	24	24	0	384.80	–	77.82	33	33	0	319.78	0.129	1,823.13	24	12	0
0	10	214.57	–	50.38	24	24	0	384.89	–	160.52	33	33	0	319.90	0.141	1,835.96	24	12	0
1	1	214.51	–	10.88	24	24	0	383.86	–	39.03	33	33	0	319.58	0.119	1,736.96	24	14	0
5	5	214.57	–	45.00	24	24	0	384.80	–	176.00	33	33	0	319.77	0.140	1,856.67	24	12	0
10	10	214.57	–	95.83	24	24	0	384.89	0.001	430.33	33	32	0	320.08	0.152	1,900.54	24	12	0
Average		214.51	–	29.10				383.57	0.000	117.53				319.58	0.129	1,761.13			

Considering all the sets, 86.86% of the instances were solved to optimality or proven to be infeasible using the proposed model. The average gap in classes with instances not solved to optimality was smaller than 0.2%. For instances RC1 in the group with $\Gamma^q = 10$ and $\Gamma^t = 10$, for example, even when only 8 of 24 instances were solved to optimality, the average gap in the remaining instances was 0.15%. The best performance of the model is observed on instances with clustered customers, as 94.04% and 100% of the instances in C1 and C2 were solved to optimality with average running times equal to 272.34 and 30.92 seconds, respectively. On the other hand, instances with mixed geographic distribution of customers were the most challenging for the model, as 63.83% and 54.02% of the instances in RC1 and RC2 were solved to optimality with average running times of 1,373.97 and 1,782.33 seconds, respectively.

The instances of all classes, except RC2, were solved to optimality in the deterministic case ($\Gamma^q = \Gamma^t = 0$). As the values of the budgets of uncertainty increase, the number of instances solved to optimality decreases, and the average elapsed time increases. Similar behavior was also observed when increasing the values of the deviations, although this finding is not evidenced in Tables 1 and 2 (see the supplementary material for further details). Therefore, as expected, the difficulty in solving the model increases as the budgets of uncertainty and/or deviations also increase. This behavior was observed even when the increase in these parameters does not change the deterministic solution. For example, for class R1, the objective values and solutions in the group with $\Gamma^q = \Gamma^t = 0$ are the same as those in the group with $\Gamma^q = 10$ and $\Gamma^t = 0$, although in the latter case, we observe a higher average elapsed time, and not all instances were solved to optimality. Infeasible instances arise only in classes R1 and RC1 and for groups with the deviation value $\alpha^t = 0.5$. Thus, when the travel time increases by 50%, no robust solution can be found for some instances in classes R1 and RC1, whereas all instances of other classes remain robust feasible. We observe more infeasible instances in class R1 than in class RC1, as they are more sensitive to increases in the travel time. Specifically, the infeasible instances were R101, R102, R103, R104 and RC105.

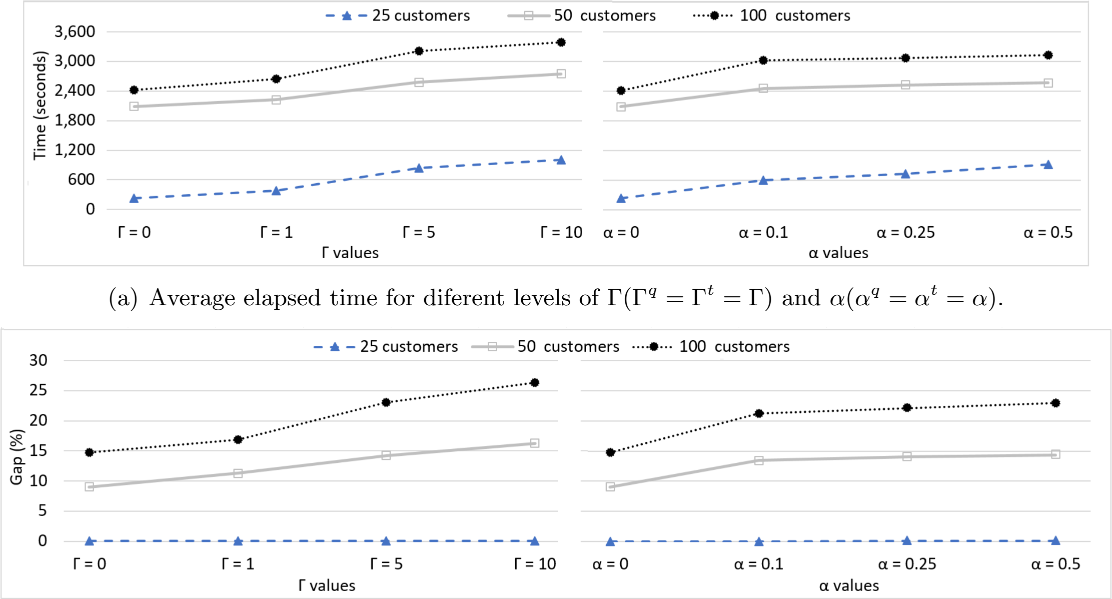
The average results for instances with 25, 50 and 100 customers are summarized in Table 3. Regarding the instances with 50 and 100 customers, CPLEX was able to find feasible solutions for most of them using the proposed model within the time limit of 3,600 seconds. Notice that 34.76% (21.36%) of the instances with 50 (100) customers were solved to optimality. For instances not solved to optimality within the time limit, the average gap in classes C1, R1 and RC1 was 5.38% (11.55%), 12.73% (27.03%), and 24.58% (33.66%), respectively, for instances with 50 (100) customers. The average gap in classes C2, R2 and RC2 was 0.75% (1.73%), 8.09% (20.27%), and 26.94% (24.95%), respectively, for instances with 50 (100) customers. Evidently, instances in group RC were more challenging, as only 13.39% (8.93%) of the instances with 50 (100) customers were solved to optimality. Similar to the results with 25 customers, instances with larger values of the budgets of uncertainty and deviations were more challenging.

Tab. 3: Average results of the robust two-index model for instances with 25, 50 and 100 customers.

	25 customers						50 customers						100 customers					
Class	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Inf*	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Inf*	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Inf*
C1	215.28	0.01	272.34	252	237	0	415.67	5.38	1,207.57	252	175	0	976.72	11.55	2,619.04	252	73	0
R1	470.93	0.01	291.32	336	300	24	780.84	12.73	2,917.41	336	63	42	1,294.60	27.03	3,273.87	336	30	42
RC1	400.68	0.07	1,373.97	224	143	6	849.60	24.58	3,147.61	224	28	6	1,555.78	33.66	3,335.55	224	19	12
C2	214.51	–	30.92	224	224	0	360.60	0.73	804.88	224	186	0	603.98	1.75	1,284.29	224	164	0
R2	383.67	0.00	124.68	308	307	0	623.92	8.09	3,017.97	308	61	0	962.54	20.27	3,319.03	308	28	0
RC2	319.60	0.13	1,782.33	224	121	0	578.66	26.94	3,131.37	224	32	0	1,099.05	24.95	3,361.90	224	21	0
Avg.	334.11	0.03	645.93	261	222	5	601.55	13.08	2,371.14	261	91	8	1,082.11	19.87	2,865.61	261	56	9

Figures 2(a) and 2(b) show the average elapsed time and average gap of instances with 25, 50 and 100 customers for different values of the budgets of uncertainty ($\Gamma^q = \Gamma^t = \Gamma$) and deviations ($\alpha^q = \alpha^t = \alpha$). Notice that the elapsed times and gaps increase when the budgets of uncertainty and deviation increase. Additionally, as expected, the instances are more challenging when the number of customers grows. Detailed results using the instances with 50 and 100 costumers are reported in the supplementary material.

To verify the performance of the proposed model in relation to those of other RO (compact) formulations reported in the literature, we extended the robust layered formulation of Agra et al. (2012) to include demand uncertainty by adding a robust capacity constraint and dualizing it



(a) Average elapsed time for diferent levels of $\Gamma(\Gamma^q = \Gamma^t = \Gamma)$ and $\alpha(\alpha^q = \alpha^t = \alpha)$.

(b) Average gap for diferent levels of $\Gamma(\Gamma^q = \Gamma^t = \Gamma)$ and $\alpha(\alpha^q = \alpha^t = \alpha)$.

Fig. 2: Average elapsed time and gap for different values of the budgets of uncertainty and deviations.

according to the standard dualization scheme used in the RO literature. This formulation is based on a layered representation of the graph and as suggested in Agra et al. (2012), it is important to recur to a preprocessing strategy to reduce the number of layers needed to represent the problem. At first, we attempted to perform computational experiments without applying this preprocessing strategy (i.e., setting the number of layers equal to the number of nodes), but it resulted in a very poor performance of the model. For example, when $\Gamma^t > 0$, the layered formulation without preprocessing was not able to generate feasible solutions for any instance in class R1 with 25 customers within a time limit of 3,600 seconds. For the same instances, when $\Gamma^t = 0$ was used, the gap was 65.68% on average. For classes C2, R2 and RC2, the reduction in layers was not effective, as these instances have wider time windows and larger vehicle capacities than the other classes, which weakens the layer reduction.

Table 4 shows the average results of the robust layered formulation (with layer reduction) for instances in classes C1, R1 and RC1 with 25 customers. The column "*Ratio*" shows the proportion of the average objective value of this formulation with respect to our robust two-index formulation. The results reveal that our model is superior to the layered formulation. Indeed, the layered formulation was not able to prove the optimality of any of the instances, while the two-index formulation proved optimality for 86.86% of them. Moreover, within the time limit of 3,600 seconds, the two-index formulation found feasible solutions or proved infeasibility for all the instances, while the robust layered formulation found feasible solutions or proved infeasibility for 59.48% of them. The analysis of the objective values and ratios demonstrates that considering instances in classes C1, R1 and RC1, the robust counterpart of the layered formulation leads to upper bounds approximately 2.71, 1.56 and 2.39 times worse than those obtained with our robust model, respectively. Even the best performance of the robust layered model in class R1 for $\Gamma^t = 5$ yields an objective value 1.29 times worse than the one obtained by our model.

Tab. 4: Average results of the robust layered formulation with different values of budgets of uncertainty using instances in classes C1, R1 and RC1 with 25 customers.

		C1						R1						RC1					
Γ^q	Γ^t	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Ratio*[1]	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Ratio*[1]	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Ratio*[1]
0	0	352.99	41.10	3,600	9	0	1.85	771.90	51.74	3,600	12	0	1.67	702.46	48.43	3,600	8	0	2.01
1	0	347.96	46.83	3,600	27	0	1.72	766.40	51.43	3,600	36	0	1.65	643.94	58.89	3,600	24	0	1.79
5	0	420.93	60.05	3,600	27	0	1.76	774.25	52.10	3,600	36	0	1.67	638.94	60.09	3,600	24	0	1.48
10	0	462.15	61.40	3,600	27	0	1.93	755.66	51.22	3,600	36	0	1.63	629.29	59.29	3,600	24	0	1.44
0	1	815.81	75.12	3,600	27	0	4.24	732.33	42.56	3,600	36	0	1.56	812.34	56.91	3,600	24	0	2.28
0	5	NA	NA	3,600	27	0	NA	617.54	37.14	3,600	36	0	1.29	638.11	61.54	3,600	24	0	1.67
0	10	NA	NA	3,600	27	0	NA	668.94	40.50	3,600	36	0	1.40	616.10	52.22	3,600	24	0	1.57
1	1	969.49	84.01	3,600	27	0	4.78	765.69	44.18	3,600	36	0	1.63	1,829.47	86.65	3,600	24	0	4.96
5	5	NA	NA	3,600	27	0	NA	743.64	41.62	3,600	36	0	1.56	1,502.72	79.10	3,600	24	0	3.38
10	10	NA	NA	3,600	27	0	NA	723.81	41.40	3,600	36	0	1.51	1,501.24	77.89	3,600	24	0	3.36
Average		561.56	61.42	3,600			2.71	732.02	45.39	3,600			1.56	951.46	64.10	3,600			2.39

NA: Value is not available.

[1] $Ratio = \frac{Obj \text{ layered}}{Obj \text{ two-index}}$.

We are not aware of any other RO compact model that leads to a reasonable performance in practice for solving instances with 25 or more customers using a general-purpose optimization software. In fact, as mentioned in previous papers and confirmed by our results, the other compact models for the RVRPTW and related variants are not effective for instances with more than 20 customers (Agra et al., 2012; Lee et al., 2012). For example, for the three-index vehicle flow formulation proposed by Lee et al. (2012) and applied to the VRP with deadlines, a general-purpose optimization software was reported to fail to prove optimality for an instance with 10 customers and a budget of uncertainty equal to 1 after 10 hours of computations. They obtained an out-of-memory error from the software in an experiment with the budget of uncertainty equal to 2 due to the huge number of constraints in their model. The compact formulation proposed by Agra et al. (2012) to the heterogeneous fleet RVRPTW with travel time uncertainty was suitable for solving instances with up to 20 nodes only. For larger instances, they recommend alternative RO techniques.

5.2 Computational performance of the branch-price-and-cut method

This section presents the results of the proposed BPC method using the RVRPTW instances with 100 customers and a time limit of 3,600 seconds. All the instances with 25 customers used in the previous section were solved to optimality or proven to be infeasible by the BPC within a few seconds. Similarly, 91.13% of the instances with 50 customers were solved to optimality or proven to be infeasible by the BPC. Detailed results obtained using the BPC to solve instances with 25 and 50 costumers are reported in the supplementary material, and thus, we omit them from this section. We solved the instances using the same combinations of values described in Section 5.1 for the parameters Γ^q, Γ^t, α^q and α^t. Following the same structure of the previous tables, Tables 5 and 6 show the results grouped by the values chosen for the budgets of uncertainty Γ^q and Γ^t. The gap of an instance not solved to optimality within 3,600 seconds is computed as 100*(UB - LB)/UB, where UB is the objective value of the solution obtained by the BPC and LB is the optimal value of the linear relaxation at the root node.

The proposed BPC method solved 58% of the instances to optimality. The highest average gap was 3.17% for instances in class C1, while the smallest one was found for instances in class RC2 with 0.29%. However, the smallest percentage of optimal solutions was achieved for class R2, in

Tab. 5: Average results of the BPC method with different values of budgets of uncertainty using instances in classes C1, R1 and RC1 with 100 customers.

Γ^q	Γ^t	C1						R1						RC1					
		Obj	*Gap*	*Time*	*Ins*	*Opt*	*Inf*	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Inf*	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Inf*
0	0	826.70	–	21.56	9	9	0	1,173.61	0.19	519.75	12	11	0	1,334.49	0.34	739.50	8	7	0
1	0	924.02	2.98	1,517.81	27	18	0	1,174.70	0.40	912.39	36	30	0	1,347.63	1.07	1,251.46	24	18	0
5	0	1,043.62	6.49	2,944.78	27	9	0	1,181.48	0.68	1,239.03	36	27	0	1,388.11	1.92	1,791.79	24	14	0
10	0	1,057.23	5.69	2,803.70	27	10	0	1,187.21	0.96	1,371.17	36	24	0	1,401.77	1.95	1,929.21	24	13	0
0	1	847.99	0.25	682.44	27	24	0	1,154.24	0.44	900.33	36	24	7	1,399.36	0.88	1,024.04	24	17	2
0	5	863.05	0.22	696.59	27	24	0	1,195.23	0.59	1,083.42	36	21	7	1,472.40	1.03	1,059.21	24	18	2
0	10	863.53	0.28	660.93	27	24	0	1,203.29	0.49	1,251.53	36	22	7	1,481.05	0.91	1,107.58	24	17	2
1	1	927.19	2.92	1,605.93	27	18	0	1,154.42	0.70	1,061.44	36	20	7	1,411.58	1.52	1,235.67	24	15	2
5	5	1,046.98	7.05	3,070.30	27	7	0	1,197.28	0.52	1,277.47	36	22	7	1,492.56	1.81	1,559.38	24	13	2
10	10	1,060.66	5.86	2,867.89	27	9	0	1,207.75	1.08	1,457.61	36	17	7	1,506.19	1.60	1,598.50	24	14	2
Average		946.10	3.17	1,687.19				1,182.92	0.61	1,107.41				1,423.51	1.30	1,329.63			

Tab. 6: Average results of the BPC method with different values of budgets of uncertainty using instances in classes C2, R2, RC2 with 100 customers.

Γ^q	Γ^t	C2						R2						RC2					
		Obj	*Gap*	*Time*	*Ins*	*Opt*	*Inf*	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Inf*	*Obj*	*Gap*	*Time*	*Ins*	*Opt*	*Inf*
0	0	587.38	0.00	570.25	8	7	0	872.56	0.83	2,732.82	11	3	0	1,000.68	0.27	1,537.88	8	5	0
1	0	587.38	0.00	600.63	24	21	0	885.46	0.54	2,635.82	33	12	0	1,001.11	0.34	1,540.88	24	15	0
5	0	594.45	0.56	1,312.17	24	17	0	875.58	0.65	2,735.15	33	11	0	1,001.07	0.28	1,559.67	24	15	0
10	0	619.95	2.94	2,063.42	24	12	0	885.13	0.64	2,732.06	33	11	0	1,001.65	0.27	1,580.67	24	15	0
0	1	593.43	0.22	1,261.17	24	18	0	886.10	1.47	2,523.00	33	13	0	1,006.06	0.27	1,735.67	24	14	0
0	5	599.81	0.28	1,714.29	24	17	0	893.60	1.26	2,618.21	33	11	0	1,010.81	0.20	1,677.38	24	15	0
0	10	599.81	0.38	1,895.96	24	16	0	895.52	1.44	2,610.15	33	13	0	1,012.25	0.34	1,707.29	24	14	0
1	1	593.43	0.31	1,315.79	24	17	0	893.50	1.41	2,516.15	33	14	0	1,006.04	0.32	1,796.54	24	13	0
5	5	606.33	1.18	2,439.96	24	12	0	897.33	1.35	2,694.42	33	13	0	1,010.89	0.26	1,722.83	24	14	0
10	10	632.26	4.35	3,066.33	24	6	0	897.15	0.93	2,779.24	33	13	0	1,012.93	0.32	1,877.33	24	13	0
Average		601.42	1.02	1,624.00				888.19	1.05	2,657.70				1,006.35	0.29	1,673.61			

which only 37% of the instances were solved to optimality. Thus, for the BPC, the most challenging instances were those in class R2. On the other hand, the method showed good performance on instances in class R1, as it proved the optimality for 65% of these instances with an average elapsed time of 1,062 seconds. Although the geographic distribution of customers is random in both classes, the instances in R2 have wider time windows and larger vehicle capacities, and hence, they become more challenging for the BPC method, as expected.

Instances with clustered customers (C1 and C2) are more sensitive to uncertainties in the customer demands than in the travel times, as indicated by the differences between the groups with either $\Gamma^q = 0$ or $\Gamma^t = 0$ in objective values, running times and number of instances solved to optimality. For other classes, differences in the impact of uncertainties in the customer demands or in the travel times are not so significant. We observe instances that became infeasible due to deviation values $\alpha^t = 0.25$ and $\alpha^t = 0.5$ only in classes R1 and RC1. This means that no robust solution can be found when the travel times increase by 25% and 50% for some instances in these classes. The infeasible instances were R101, R102, R103, R104, and RC105, as in the experiments with 25 customers.

Similar to the results presented in the previous section for the proposed compact model, the running times of the BPC method increase significantly for some values of the budgets of uncertainty and the deviations in comparison to those of the deterministic formulation. However, an interesting difference is observed: as the values of the budgets of uncertainty or the deviations increase, the instances do not necessarily become more challenging. For example, we observe similar average running times for instances in class C1 when $\Gamma^q = 0$ and Γ^t varies (see Table 5). The BPC method was faster and solved more instances to optimality for $\Gamma^q = \Gamma^t = 10$ than for $\Gamma^q = \Gamma^t = 5$. This behavior can be explained by a trade-off in the labeling algorithm for the robust RCESPP. On the one hand, for larger budgets of uncertainty, the amount of information in a label increases, and more comparisons are needed to check the dominance rule in the algorithm. On the other hand, larger budgets of uncertainty and deviations lead to fewer extensions, as these extensions are more likely to result in (robust) infeasible paths than in the deterministic case.

5.3 Robustness analysis

We designed a robustness analysis based on a Monte Carlo simulation to evaluate the quality of solutions resulting from our RO approaches. The simulation was performed by generating 10,000 random uniform realizations for demands and travel times in the half-interval $[q_i, q_i + \hat{q}_i]$ for all $i \in \mathcal{C}$ and $[t_{ij}, t_{ij} + \hat{t}_{ij}]$ for all $(i, j) \in \mathcal{E}$, respectively. Since the goal of the RVRPTW is to find solutions that are immunized against uncertainty, one might expect that the corresponding routes are more likely to be feasible than those from traditional deterministic approaches when demands and/or travel times increase, which is what commonly occurs in real-life situations. Moreover, the extra cost incurred by such a robust solution should be compensated by the gain in terms of robustness/feasibility. To analyze the aforementioned trade-off, we pay particular attention to the following performance measures:

- The *price of robustness* (PoR), which is defined as $\frac{z(x^{\boldsymbol{\alpha},\boldsymbol{\Gamma}})-z}{z} \cdot 100\%$, in which $x^{\boldsymbol{\alpha},\boldsymbol{\Gamma}}$ is the (sub)optimal solution for a given pair of deviations (α^q, α^t) and budgets of uncertainty (Γ^q, Γ^t); $z(x^{\boldsymbol{\alpha},\boldsymbol{\Gamma}})$ is its corresponding objective value; and z is the (sub)optimal value of the equivalent deterministic (nominal) problem. All these solutions were provided by the BPC method within a running time of 3,600 seconds (thus, the solutions obtained for some of the 100 instances may not be optimal).

- The *probability of constraint violation* (Risk), which is empirically evaluated as the number of times a given optimal solution $x^{\boldsymbol{\alpha},\boldsymbol{\Gamma}}$ is infeasible out of the 10,000 generated random realizations given by the Monte Carlo simulation. An estimate of this probability is obtained via the relative frequency of infeasible solutions. The simulation procedure is summarized as follows:

  1. **Input.** Solution $x^{\boldsymbol{\alpha},\boldsymbol{\Gamma}}$.
  2. For $\omega = 1$ to 10,000, do:
     - $\tilde{q}_i^\omega \sim U[q_i, q_i + trunc(\alpha^q \times q_i)]$, $\forall i \in \mathcal{C}$;
     - $\tilde{t}_{ij}^\omega \sim U[t_{ij}, t_{ij} + 0.1 \times trunc(\alpha^t \times 10 \times st_{ij})]$, $\forall (i, j) \in \mathcal{E}$;
     - Evaluate the feasibility of $x^{\boldsymbol{\alpha},\boldsymbol{\Gamma}}$ using $\tilde{q}^\omega$ and $\tilde{t}_{ij}^\omega$ according to conditions (i)–(v) defined in Section 2.
  3. **Output.** The empirical probability function for the constraint violation of a given solution $\mathscr{P}[X(\alpha^q, \alpha^t, \Gamma^q, \Gamma^t)]$.

The robustness analysis takes into account the instances with 25 and 100 customers in classes C1, R1 and RC1. We have run experiments using all integer values ranging from 0 to 10 for both budgets of uncertainty and each value in the set $\{0, 0.1, 0.25, 0.5\}$ for both deviations. The results for the instances with 25 and 100 customers are summarized in Tables 7 and 8, respectively. For each class of instances and combination of budget of uncertainty and deviation, the tables show the average price of robustness and the average probability of constraints violation, both of which are expressed in percentage scale, over the total number of instances. Table 7 presents only the results for Γ^q and Γ^t up to 6 because the probability of constraint violation is equal to 0 from this value onwards in all instances. For the same reason, Table 8 presents only the results for Γ^q and Γ^t up to 7. Average values equal to zero are represented in the tables using the symbol '$-$'. Only instances that are feasible for all the values of the budget of uncertainty and deviation were considered in the analysis. Thus, instances R101, R102, R103, R104, and RC105 were discarded.

For the results in the columns corresponding to α% deviation, with $\alpha = 10, 25, 50$, the BPC method and the Monte Carlo simulation were run using α^q and/or α^t equal to $0.01 \times \alpha$. For the BPC method, if $\Gamma^q > 0$ and $\Gamma^t > 0$, then $\alpha^q = \alpha^t = \alpha$; otherwise, $\Gamma^q = 0$ ($\Gamma^t = 0$) implies $\alpha^q = 0$ ($\alpha^t = 0$). This is different from the Monte Carlo simulation, as we also need to verify the risk of the deterministic solutions regarding different deviation values. Hence, in the first set of results, with $\Gamma^q = 0, 1, \ldots, 6$ and $\Gamma^t = 0$, the risks were obtained from a Monte Carlo simulation using $\alpha^q = \alpha$ and $\alpha^t = 0$; in the second set of results, with $\Gamma^q = 0$ and $\Gamma^t = 0, 1, \ldots, 6$, the risks were obtained from a simulation using $\alpha^q = 0$ and $\alpha^t = \alpha$; and finally, the risks for $\Gamma^q = \Gamma^t = 0, 1, \ldots, 6$ were obtained from a simulation using $\alpha^q = \alpha^t = \alpha$.

Unsurprisingly, the deterministic approach fails in protecting against uncertainty for all classes of instances and deviations. The average risk metric indeed reveals that under uncertainty, most routes will be infeasible if demands/travel times increase. Even though the average risk is at most 30.5% for the deterministic solutions with 25 customers and deviations of 10%, it significantly increases for larger deviations, reaching at least 71.90% in RC1 instances. This behavior is more pronounced when 100 customers are considered in the analysis. In such cases, only the instances in class R1 present a relatively low probability of constraint violation, 0.1198, when $\alpha^q/\alpha^t = 0.1$.

A careful analysis of the non-robust solutions from instances in class C1 also reveals that they are even less likely to hedge against demand uncertainty than travel time uncertainty. When only travel time uncertainty is considered, for example, instances in class C1 with 25 customers exhibit robustness for 10% and 25% deviations. The opposite situation is observed for instances in R1 and RC1, which are more affected by travel time uncertainty. In particular, the deterministic solutions of instances in R1 with 25 customers are robust for all demand deviations considering no travel time uncertainty: in these cases, *risk* and *PoR* are zero for all budgets. The higher impact of either demand or travel time uncertainty is analogous for the robust solutions, but as the deviations increase, both uncertainty sources become equally important.

It appears that the robust solutions strike a good trade-off between decreasing the probability of constraint violation and not being too conservative. This main finding is illustrated by the plots of *risk* versus *PoR* depicted in Figures 3 and 4. Each ordered pair highlighted in the plots (triangles for C1, squares for R1 and circles for RC1) is associated with a different budget of uncertainty value. The values of deviation are indicated in the figures as "Deviation: $[\alpha^q, \alpha^t]$". As the trade-off curves become more convex and nearer to the vertical axis, the robust solutions improve. In this sense, we see only a few examples in which the curves exhibit a poor trade-off, and most of them are

Tab. 7: Price of robustness (PoR) and probability of constraint violation (Risk) in percentage for instances with 25 customers.

		10% deviation						25% deviation						50% deviation					
		C1		R1		RC1		C1		R1		RC1		C1		R1		RC1	
Γ^q	Γ^t	PoR	Risk	PoR	Risk	PoR	Risk	PoR	Risk	PoR	Risk	PoR	Risk	PoR	Risk	PoR	Risk	PoR	Risk
0	0	–	11.30	–	–	–	–	–	95.38	–	–	–	82.91	0.00	99.98	–	–	–	97.21
1	0	0.20	0.30	–	–	–	–	0.20	94.91	–	–	–	82.91	18.63	99.21	–	–	9.94	97.05
2	0	0.20	0.30	–	–	–	–	18.63	26.80	–	–	9.94	42.98	23.22	40.13	–	–	32.22	11.16
3	0	18.62	–	–	–	–	–	19.80	8.06	–	–	21.66	0.05	24.60	29.39	–	–	32.45	1.44
4	0	18.63	–	–	–	9.94	–	22.90	0.01	–	–	21.81	0.03	33.01	0.07	–	–	33.08	0.36
5	0	18.63	–	–	–	9.94	–	23.22	–	–	–	32.22	–	33.57	0.20	–	–	33.65	–
6	0	18.63	–	–	–	9.94	–	23.22	–	–	–	32.22	–	34.41	–	–	–	39.58	–
0	0	–	–	–	30.50	–	21.98	–	–	–	71.90	–	73.74	0.00	4.53	–	97.98	–	98.08
0	1	–	–	0.95	12.15	0.23	7.82	–	–	2.61	25.45	0.74	30.45	2.76	0.10	5.40	60.60	7.81	48.97
0	2	–	–	1.34	11.66	0.47	0.07	–	–	3.11	19.98	3.02	5.31	3.34	0.10	7.51	7.19	17.09	16.61
0	3	–	–	1.55	–	0.47	0.07	–	–	3.61	0.03	5.31	0.65	3.34	0.10	9.21	1.87	20.25	3.82
0	4	–	–	1.55	–	0.62	–	–	–	3.88	–	5.40	0.09	8.24	–	9.77	1.84	21.32	0.38
0	5	–	–	1.55	–	0.62	–	–	–	3.98	–	8.59	–	8.24	–	10.53	0.06	24.24	0.03
0	6	–	–	1.55	–	0.62	–	–	–	3.98	–	8.59	–	8.24	–	10.78	–	29.70	–
0	0	–	11.32	–	30.36	–	21.98	–	95.40	–	71.90	–	99.14	0.00	99.98	–	98.01	–	100.00
1	1	0.20	0.25	0.95	12.39	0.23	7.85	0.20	94.79	2.61	25.36	0.74	84.27	18.92	97.71	5.40	60.47	19.03	85.50
2	2	0.20	0.25	1.34	11.68	0.47	0.10	18.92	22.05	3.11	19.76	15.16	27.65	24.01	38.89	7.51	7.29	38.80	19.77
3	3	18.71	–	1.55	–	0.47	0.10	20.40	6.59	3.61	0.03	24.64	0.05	25.77	26.66	9.21	1.74	40.36	2.90
4	4	18.75	–	1.55	–	10.75	–	23.05	0.02	3.88	–	24.71	0.04	33.30	0.08	9.77	1.71	41.54	0.01
5	5	18.75	–	1.55	–	11.69	–	23.43	–	3.98	–	34.90	0.01	33.91	0.14	10.53	0.07	42.25	–
6	6	18.75	–	1.55	–	11.69	–	23.43	–	3.98	–	34.92	–	34.22	–	10.78	–	42.78	–

Tab. 8: Price of robustness (PoR) and probability of constraint violation (Risk) in percentage for instances with 100 customers.

		10% deviation						25% deviation						50% deviation					
		C1		R1		RC1		C1		R1		RC1		C1		R1		RC1	
Γ^q	Γ^t	PoR	Risk	PoR	Risk	PoR	Risk	PoR	Risk	PoR	Risk	PoR	Risk	PoR	Risk	PoR	Risk	PoR	Risk
0	0	–	100.00	–	11.98	–	85.62	–	100.00	–	49.08	–	87.98	–	100.00	–	88.38	–	99.94
1	0	8.11	0.39	0.03	–	0.78	9.92	8.70	99.95	0.11	35.02	0.94	87.62	18.71	100.00	0.20	80.80	1.72	99.84
2	0	8.11	0.39	0.03	–	0.86	1.85	18.72	67.86	0.12	17.53	1.66	43.07	22.56	99.67	0.53	60.08	3.33	84.27
3	0	12.85	–	0.12	–	0.92	–	19.80	19.28	0.18	6.53	1.99	16.09	26.08	81.27	0.66	21.27	4.67	36.73
4	0	16.71	–	0.11	–	1.36	–	20.90	0.07	0.49	0.65	2.43	0.30	29.65	17.79	1.43	1.77	5.59	4.93
5	0	17.84	–	0.12	–	1.54	–	21.92	0.01	0.73	0.03	3.03	0.01	38.95	7.87	1.78	0.12	8.85	0.19
6	0	18.06	–	0.29	–	1.69	–	24.35	–	0.71	–	3.32	–	39.74	0.07	2.56	0.01	10.04	0.01
7	0	18.03	–	0.11	–	1.71	–	24.37	–	0.74	–	3.87	–	39.92	–	3.32	–	10.34	–
0	0	–	99.82	–	100.00	–	99.97	–	100.00	–	100.00	–	100.00	–	100.00	–	100.00	–	100.00
0	1	2.12	–	0.76	89.23	1.26	83.97	2.21	3.69	2.33	96.87	4.65	97.89	3.39	9.66	4.61	100.00	13.37	97.63
0	2	2.21	–	1.50	45.80	2.44	32.59	2.19	0.10	3.53	81.40	6.77	55.34	5.88	1.38	8.89	87.64	19.80	69.69
0	3	2.10	–	1.77	23.55	3.02	8.24	2.28	0.12	4.59	32.24	8.09	19.26	6.04	0.66	11.13	51.66	22.29	19.36
0	4	2.03	–	1.97	5.76	3.12	0.24	2.22	–	5.27	12.11	8.57	2.80	7.82	–	12.13	19.54	23.56	1.49
0	5	2.21	–	2.15	0.02	3.10	–	2.35	–	5.47	2.44	9.01	0.04	8.63	–	12.68	0.81	24.38	0.06
0	6	2.21	–	2.19	0.01	3.32	–	2.35	–	5.68	0.26	9.16	–	8.63	–	13.54	0.03	24.89	–
0	7	2.21	–	2.22	–	3.41	–	2.35	–	5.88	0.00	9.37	–	8.80	–	14.00	–	25.05	–
0	0	–	100.00	–	100.00	–	100.00	–	100.00	–	100.00	–	100.00	–	100.00	–	100.00	–	100.00
1	1	8.45	0.83	0.78	94.82	2.15	89.43	9.04	99.96	2.26	97.11	5.23	98.42	19.22	100.00	4.71	100.00	14.76	99.71
2	2	8.45	0.86	1.53	43.42	3.65	35.98	19.20	60.57	3.68	82.55	8.07	61.27	25.29	99.67	8.98	89.15	20.87	72.12
3	3	13.54	–	1.79	21.52	3.94	4.09	20.58	18.48	4.94	29.00	9.59	10.13	28.29	76.00	11.38	54.79	23.24	33.13
4	4	17.27	–	1.97	5.68	4.32	0.12	21.57	0.08	5.29	7.57	10.70	4.41	32.11	6.98	12.42	10.34	24.77	2.72
5	5	18.41	–	2.19	0.03	4.39	0.02	22.65	0.01	5.53	2.14	11.09	0.02	39.12	0.51	13.27	2.19	25.82	0.15
6	6	18.26	–	2.24	–	4.85	–	25.18	–	5.85	0.15	11.06	–	39.81	0.06	14.08	0.01	26.84	0.03
7	7	18.53	–	2.29	–	4.59	–	24.56	–	6.26	–	11.50	–	40.38	–	14.48	–	27.53	–

Note. The cases in which PoR (Risk) does not present a monotonic increase (decrease) are justified by non-optimal solutions.

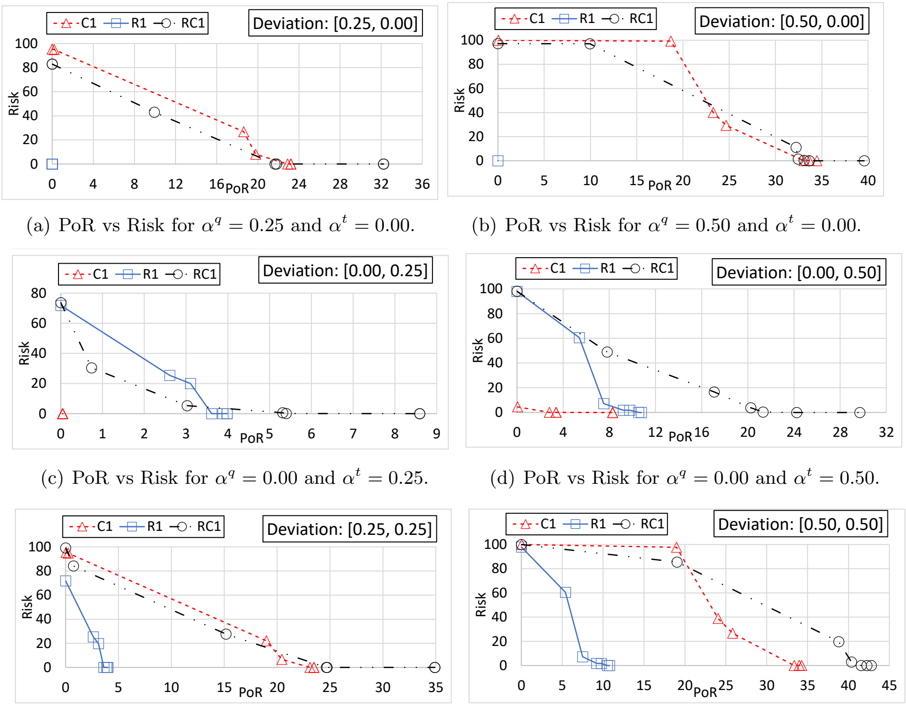
related to instances in C1. In particular, we can see in Figures 3(b) and 4(f) quasi-horizontal dashed lines, indeed showing that risk is not improved but PoR substantially increases. This behavior is especially true for less conservative budgets. When robustness is further enforced, risk sharply decreases accordingly, e.g., an almost zero-risk robust solution with a PoR equal to 39.81% (C1 instance with 100 customers and a 50% deviation).

On the other hand, we found many cases where the solutions are *much* less risky at the expense of a negligible deterioration in average costs. For example, we have to pay a 0.20% (0.62%) higher cost to reduce the risk from 11.30% (21.98%) to 0.30% (0.00%) for instances in C1 (RC1) with 25 customers and a 10% deviation. In some of those cases, mitigating the risk completely is not worthwhile because PoR may increase dramatically as more robustness is enforced; see that PoR in C1 varies from 0.20% to 18.62% to eliminate the corresponding risk. Remarkably, instances in class RC1 with 100 customers and a 10% deviation yield a zero-risk solution with a negligible PoR of 0.92%. Rich robust solutions that can mitigate risk while incurring a minor price are more frequently seen for instances in class R1. This result is confirmed in Figures 3(e), 3(f), 4(a), 4(b), 4(e), and 4(f), in which the trade-off curves clearly dominate those from instances in C1 and RC1. Interestingly, when both parameters vary, the risk is only marginally increased, suggesting that robust strategies that encapsulate only the most significant uncertainty source end up protecting against the remaining uncertain parameter. In those cases, decision makers might avoid testing additional budget of uncertainty and deviation values for various uncertain parameters simultaneously. A similar discussion is introduced in Alem and Morabito (2012); Righetto et al. (2016).

Finally, it is worth noting that the robust solutions do not suffer from the "curse of overprotection", as noted in several papers that address demand uncertainty in practical applications (Alem et al., 2018; Richter and Stiller, 2018). In such cases, even a small demand deviation may lead to a dramatic PoR, compromising the effectiveness of the robust approach. In our case, zero-risk solutions are attained by paying a PoR of 42.25% in the worst-case situation (RC1 instances with 25 customers and a 50% deviation), whereas the average PoR over all the 54 average values is less than 15%. Therefore, it is almost always beneficial to adopt robust solutions to solve real-life problems.

5.4 Design of an optimal robust route

An interesting insight could be obtained if we examine the changes in the structures of both robust and non-robust solutions. To this end, we depicted in Figures 5 to 7 the optimal solution of the (arbitrarily chosen) instance RC102 with 25 customers and a 25% deviation when $\Gamma^q = 0$ and $\Gamma^t = 0, 1, 2$. The schemes present the sequence of nodes visited in each route, where nodes 0 and 26 correspond to the depot. Customer time windows are shown right above each customer node, and the depot time window is $[0, 240]$. The service times in this instance are the same for all customers and are equal to 10. Nominal travel times are given below the arrows connecting the nodes. Under each customer node, the figures present the earliest exact time at which the vehicle can start service at the node. Below these numbers, in Figures 6 and 7, we also have the worst-case earliest exact time from which the service can start at the node, when up to Γ^t travel times attain their worst case, as computed using (2.5).



(a) PoR vs Risk for $\alpha^q = 0.25$ and $\alpha^t = 0.00$.

(b) PoR vs Risk for $\alpha^q = 0.50$ and $\alpha^t = 0.00$.

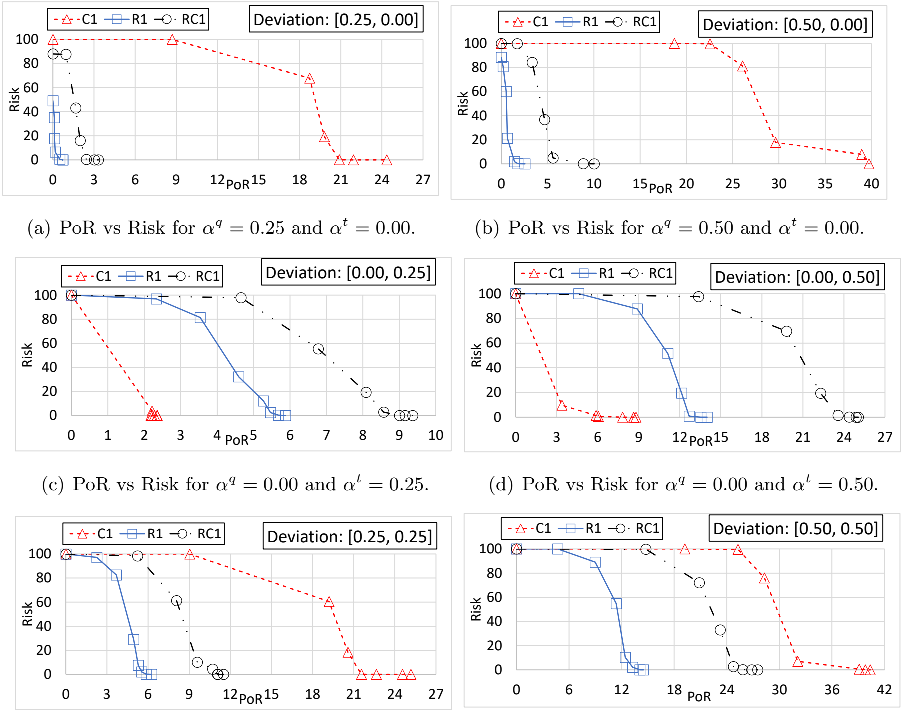
(c) PoR vs Risk for $\alpha^q = 0.00$ and $\alpha^t = 0.25$.

(d) PoR vs Risk for $\alpha^q = 0.00$ and $\alpha^t = 0.50$.

(e) PoR vs Risk for $\alpha^q = 0.25$ and $\alpha^t = 0.25$.

(f) PoR vs Risk for $\alpha^q = 0.50$ and $\alpha^t = 0.50$.

Fig. 3: Trade-off between price of robustness (PoR) and probability of constraint violation (Risk) for 100-series set instances with 25 customers.



(a) PoR vs Risk for $\alpha^q = 0.25$ and $\alpha^t = 0.00$.

(b) PoR vs Risk for $\alpha^q = 0.50$ and $\alpha^t = 0.00$.

(c) PoR vs Risk for $\alpha^q = 0.00$ and $\alpha^t = 0.25$.

(d) PoR vs Risk for $\alpha^q = 0.00$ and $\alpha^t = 0.50$.

(e) PoR vs Risk for $\alpha^q = 0.25$ and $\alpha^t = 0.25$.

(f) PoR vs Risk for $\alpha^q = 0.50$ and $\alpha^t = 0.50$.

Fig. 4: Trade-off between price of robustness (PoR) and probability of constraint violation (Risk) for 100-series set instances with 100 customers.

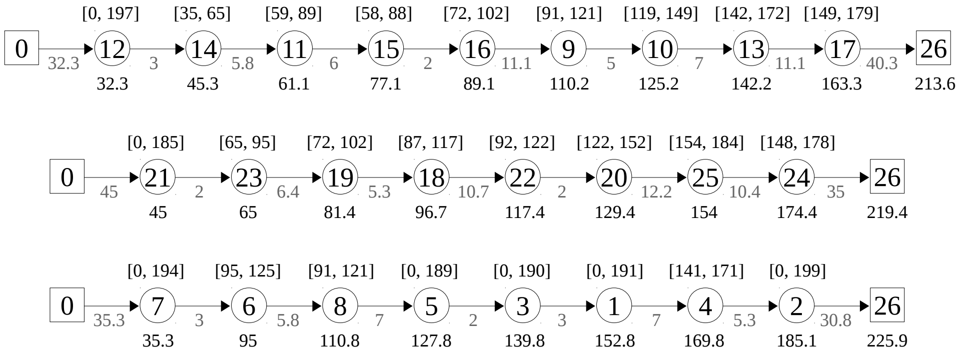


Fig. 5: Optimal solution of instance RC102 for $\Gamma^t = 0$ (deterministic solution) with an optimal value of 351.8 and a risk of 98.25%.

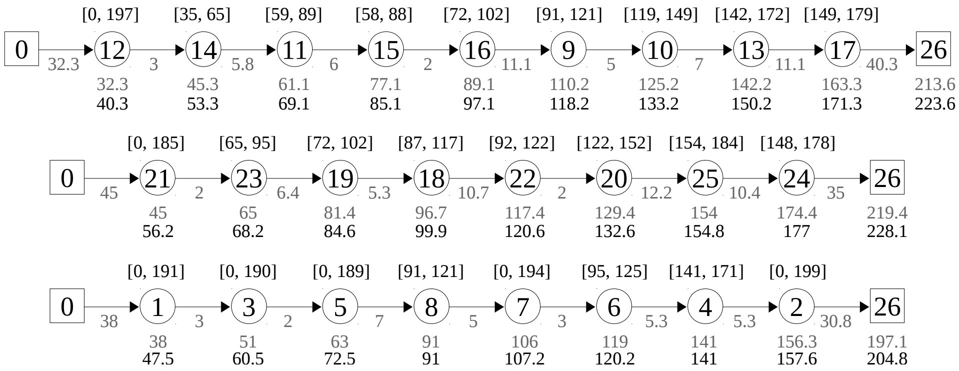


Fig. 6: Optimal solution of instance RC102 for $\Gamma^t = 1$ with an optimal value of 352.0 (PoR of 0.06%) and a risk of 50.47%.

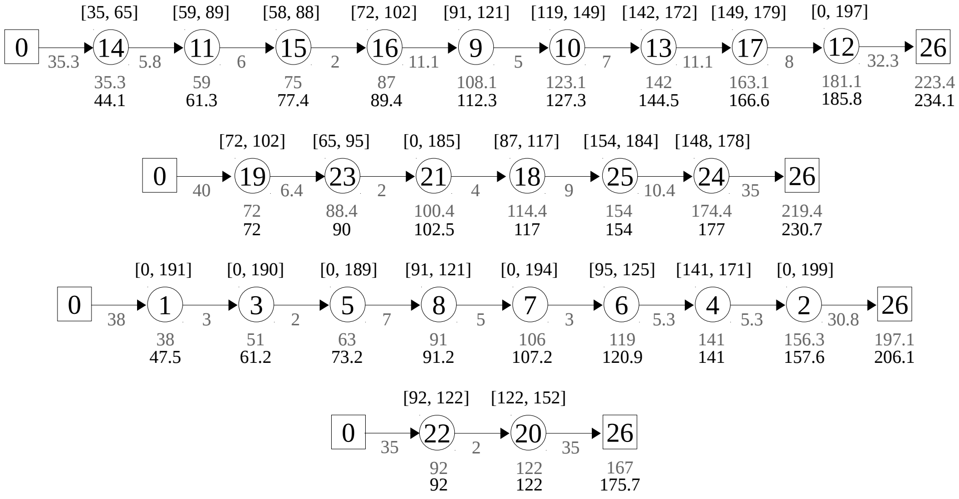


Fig. 7: Optimal solution of instance RC102 for $\Gamma^t = 2$ with an optimal value of 401.8 (PoR of 14.21%) and a risk of 0.01%.

The solution presented in Figure 5 is optimal for $\Gamma^t = 0$ and has an objective value of 351.8 with a corresponding risk equal to 98.25% (evaluated according to the Monte Carlo simulation), showing that the non-robust solution is totally unprotected against uncertainties from a practical point of view. In fact, if a single travel time in the third route assumes its worst-case value, the solution will become infeasible. This may happen, e.g., for the travel time between nodes 8 and 5 in the third route. In the worst-case scenario, this parameter increases from 7 to 8.7 for a 25% deviation (recall that we truncate the number after the first decimal place). With the time increased by 1.7, the vehicle reaches node 5 at the exact time 129.5. As a consequence, the exact time of arrival at node 4 becomes 171.5, 0.5 after the time window of this node has closed.

The optimal solution for $\Gamma^t = 1$ in Figure 6 exhibits an objective value of 352, a PoR equal to 0.06% and a risk of 50.47%. This solution is able to absorb the deviation of up to one travel time per route, achieving its worst case rapidly. The difference between the actual and the nominal solutions originates from the sequence in which the nodes are visited in the third route. By visiting the first six nodes in a different order, the third route leads to an increase in the objective value, but the worst-case deviation of the travel times is absorbed by the start time of the time window of node 4, as the vehicle arrives earlier than 141 at this node. The figure indeed shows that the exact time that the vehicle can start the service at node 6 is either 120.2 (if exactly one travel time assumes its worst-case value) or 119 (if all the travel times assume their nominal values). Then, in the worst case, the vehicle arrives in node 4 at 135.6 $(119+10+5.3+1.3$ using (2.5)) but can start the service only at the exact time 141.

The third route depicted in Figure 6 can actually absorb up to two travel times assuming their worst-case values, as shown in Figure 7 for the case with $\Gamma^t = 2$. For this budget of uncertainty, when Γ^t increases from 1 to 2, not only do the previous two routes change, but also a new route is created in an attempt to reduce the risk. This solution has an optimal value of 401.8 with a PoR of 14.21% and a risk of 0.01%, and it is almost totally immunized against uncertainties. The trade-off curve presented in Fig. 8 clearly emphasizes the feasibility gain against the price.

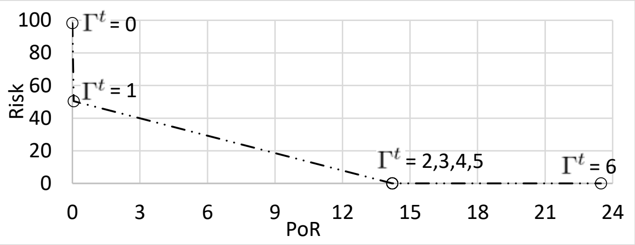


Fig. 8: Trade-off between PoR and Risk for instance RC102.

From a decision maker's standpoint, it would be problematic if the results from the robust solutions differed significantly among each other or in comparison to non-robust solutions because it would be necessary to re-schedule all routes and/or use a completely different number of vehicles to serve the customers. Although there is a reasonable deterioration in the objective value in this case, many logistics systems clearly may benefit from the robust solution for two main reasons: first, because customers could be guaranteed higher service levels, and second, because robust routes are not much more expensive and can be easily implemented by experts. For example, in the second route of Figure 7, we see that the vehicle reaches node 18 at the exact time 114.4 if the actual travel times follow their nominal values. On the other hand, if two of the actual travel times assume their worst-case values, then the vehicle will arrive at the exact time 117, which is tight with the closure of the time window but is enough to service the customer. This worst-case analysis can also be helpful in guaranteeing that the total duration of a route satisfies the legal restrictions of the driver's shift regarding working hours.

6 Conclusion

We proposed a compact formulation and a branch-price-and-cut (BPC) algorithm for the robust vehicle routing problem with time windows (RVRPTW) when both demands and travel times are subjected to uncertainty. The compact formulation is based on integrating dynamic programming (DP) recursive equations into the standard two-index vehicle flow formulation with the Miller-Tucker-Zemlin (MTZ) constraints. It is a robust counterpart model that does not require the commonly used dualization of constraints involving the uncertain parameters. The proposed formulation has also a practical appeal, as it is an intuitive extension of the deterministic formulation and thus encourages its usage by non-expert practitioners, especially those relying on general-purpose optimization software. The results of computational experiments using benchmark instances from the literature show that the model is effective in solving small-scale instances and can be used to obtain robust feasible solutions for instances with 50 and 100 customers within a reasonable running time. As the values of the budgets of uncertainty and deviations increase, the number of instances solved to optimality decreases, and the average elapsed time increases, which is a typical behavior in robust optimization (RO) models. We have also conducted additional numerical experiments with an existing RO formulation for the VRP based on the dualization technique. The results demonstrated the superiority of our model based on DP for solving the benchmark instances. More importantly, we are not aware of any other RO (compact) model that leads to a reasonable performance in practice for solving instances with 25 or more customers using a general-purpose optimization software.

The proposed BPC method encapsulates the uncertainties in the subproblem of the column generation (CG) procedure, which is a robust resource constrained elementary shortest path problem. We presented how to extend the labeling algorithm to solve this subproblem considering the uncertainties in demands and travel times simultaneously. Computational experiments with Solomon's instances show that the proposed BPC method solves most of the instances with 100 customers to optimality even for large values of budgets of uncertainty and deviations. As the results indicate, the instances become more challenging to solve after incorporating uncertainties, but the performance of the BPC remains stable as the values of the budgets of uncertainty and deviations increase. This is an important feature that distinguishes the proposed method from other approaches in the literature, which seem to degrade in performance when robust constraints are taken into account. A robustness analysis based on the solutions provided by the BPC method and a Monte Carlo simulation showed that the increase in the objective values of the robust routes is small compared to the reduction in the risk of these routes being infeasible in practice. Moreover, the robust solutions do not suffer from the "curse of overprotection", as noted in several papers that address demand uncertainty in practical applications. Therefore, the robust solutions may be more suitable for supporting decision making in logistic systems.

Our findings also bring up interesting questions for future research. We would like to adapt the methodologies proposed in this paper to other variants of the vehicle routing problem, such as the capacitated vehicle routing problem (Pecin et al., 2016) and the pickup and delivery problem with time windows (Furtado et al., 2017), in order to verify their performance and develop strategies that may be required by these variants. Another interesting research topic is to develop a specialized branch-and-cut method for the proposed compact formulation based on valid inequalities that can improve its linear relaxation bound. The resulting solution strategy has the potential to be more effective than the standalone model and to allow larger instances to be solved. Finally, we would like to explore practical applications of the RVRPTW in different logistics contexts, particularly in the last mile relief distribution, which requires efficient and effective solutions. Most papers on this subject use stochastic programs within two-/three-index formulations to handle uncertainty, which commonly results in intractable problems even for small-size instances. Therefore, it would be interesting to apply the approaches presented in this paper to effectively solving large-scale instances, thereby encouraging the usage of analytical models in practical disaster response operations.

7 Acknowledgments

The authors thank the anonymous reviewers for their valuable comments. This work was supported by the Sao Paulo Research Foundation (FAPESP) [grant numbers 13/07375-0, 14/50228-0, 15/26453-7, 16/23366-9 and 16/01860-1].

References

Adulyasak, Y. and Jaillet, P. (2016). Models and algorithms for stochastic and robust vehicle routing with deadlines. *Transportation Science*, 50(2):608–626.

Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L. M., Poss, M., and Requejo, C. (2013). The robust vehicle routing problem with time windows. *Computers & Operations Research*, 40(3):856 – 866. Transport Scheduling.

Agra, A., Christiansen, M., Figueiredo, R., Magnus Hvattum, L., Poss, M., and Requejo, C. (2012). Layered formulation for the robust vehicle routing problem with time windows. In Mahjoub, A. R., Markakis, V., Milis, I., and Paschos, V. T., editors, *Combinatorial Optimization: Second International Symposium, ISCO 2012, Athens, Greece, April 19-21, 2012, Revised Selected Papers*, pages 249–260. Springer Berlin Heidelberg, Berlin, Heidelberg.

Alem, D., Curcio, E., Amorim, P., and Almada-Lobo, B. (2018). A computational study of the general lot-sizing and scheduling model under demand uncertainty via robust and stochastic approaches. *Computers & Operations Research*, 90:125 – 141.

Alem, D. J. and Morabito, R. (2012). Production planning in furniture settings via robust optimization. *Computers & Operations Research*, 39(2):139 – 150.

Alvarez, A. and Munari, P. (2017). An exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen. *Computers & Operations Research*, 83:1–12.

Baldacci, R., Bartolini, E., and Mingozzi, A. (2011). An exact algorithm for the pickup and delivery problem with time windows. *Operations research*, 59(2):414–426.

Ben Amor, H. M., Desrosiers, J., and Frangioni, A. (2009). On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, 157(6):1167 – 1184.

Ben-Tal, A. and Nemirovski, A. (1999). Robust solutions of uncertain linear programming. *Operations Research Letters*, 25:1–13.

Bertsimas, D. and Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming*, 98(1):49–71.

Braekers, K., Ramaekers, K., and Nieuwenhuyse, I. V. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300 – 313.

Briant, O., Lemarechal, C., Meurdesoif, P., Michel, S., Perrot, N., and Vanderbeck, F. (2008). Comparison of bundle and classical column generation. *Mathematical programming*, 113(2):299–344.

Chabrier, A. (2006). Vehicle routing problem with elementary shortest path based column generation. *Computers and Operations Research*, 33(10):2972 – 2990.

Contardo, C., Desaulniers, G., and Lessard, F. (2015). Reaching the elementary lower bound in the vehicle routing problem with time windows. *Networks*, 65(1):88–99.

De La Vega, J., Munari, P., and Morabito, R. (2018). Robust optimization for the vehicle routing problem with multiple deliverymen. *Central European Journal of Operations Research*. Online First.

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

Desaulniers, G., Madsen, O. B., and Ropke, S. (2014). The vehicle routing problem with time windows. In Toth, P. and Vigo, D., editors, *Vehicle routing: Problems, methods, and applications*, MOS/SIAM Ser Optim, pages 119–159.

Desrosiers, J., Lübbecke, M. E., Cochran, J. J., Cox, L. A., Keskinocak, P., Kharoufeh, J. P., and Smith, J. C. (2010). *Branch-Price-and-Cut Algorithms.* John Wiley & Sons, Inc.

Errico, F., Desaulniers, G., Gendreau, M., Rei, W., and Rousseau, L.-M. (2016). A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times. *European Journal of Operational Research*, 249(1):55–66.

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

Furtado, M. G. S., Munari, P., and Morabito, R. (2017). Pickup and delivery problem with time windows: a new compact two-index formulation. *Operations Research Letters*, 45(4):334–341.

Gondzio, J. (2012). Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601.

Gondzio, J., Gonzalez-Brevis, P., and Munari, P. (2013). New developments in the primal-dual column generation technique. *European Journal of Operational Research*, 224(1):41–51.

Gondzio, J., González-Brevis, P., and Munari, P. (2016). Large-scale optimization with the primal-dual column generation method. *Mathematical Programming Computation*, 8(1):47–82.

Gounaris, C. E., Repoussis, P. P., Tarantilis, C. D., Wiesemann, W., and Floudas, C. A. (2016). An adaptive memory programming framework for the robust capacitated vehicle routing problem. *Transportation Science*, 50(4):1239–1260.

Gounaris, C. E., Wiesemann, W., and Floudas, C. A. (2013). The robust capacitated vehicle routing problem under demand uncertainty. *Operations Research*, 61(3):677–693.

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

Irnich, S., Toth, P., and Vigo, D. (2014). The family of vehicle routing problems. In Toth, P. and Vigo, D., editors, *Vehicle routing: Problems, methods, and applications*, MOS/SIAM Ser Optim, pages 1–33.

Jaillet, P., Qi, J., and Sim, M. (2016). Routing optimization under uncertainty. *Operations Research*, 64(1):186–200.

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

Joncour, C., Michel, S., Sadykov, R., Sverdlov, D., and Vanderbeck, F. (2010). Column generation based primal heuristics. *Electronic Notes in Discrete Mathematics*, 36:695–702.

Klamroth, K., Köbis, E., Schöbel, A., and Tammer, C. (2017). A unified approach to uncertain optimization. *European Journal of Operational Research*, 260(2):403–420.

Lee, C., Lee, K., and Park, S. (2012). Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society*, 63:1294–1306.

Lübbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, 53(6):1007–1023.

Munari, P. and Gondzio, J. (2013). Using the primal-dual interior point algorithm within the branch-price-and-cut method. *Computers & Operations Research*, 40(8):2026 – 2036.

Munari, P. and Gondzio, J. (2015). Column generation and branch-and-price with interior point methods. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 3(1).

Munari, P. and Morabito, R. (2016). A branch-price-and-cut for the vehicle routing problem with time windows and multiple deliverymen. Technical report, Industrial Engineering Department, Federal University of São Carlos.

Ordonez, F. (2010). Robust vehicle routing. In *Risk and Optimization in an Uncertain World*, chapter 7, pages 153–178.

Oyola, J., Arntzen, H., and Woodruff, D. L. (2016a). The stochastic vehicle routing problem, a literature review, part I: models. *EURO Journal on Transportation and Logistics*, pages 1–29.

Oyola, J., Arntzen, H., and Woodruff, D. L. (2016b). The stochastic vehicle routing problem, a literature review, part II: solution methods. *EURO Journal on Transportation and Logistics*, pages 1–40.

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

Pecin, D., Pessoa, A., Poggi, M., and Uchoa, E. (2016). Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, pages 1–40.

Pessoa, A. A., Di Puglia Pugliese, L., Guerriero, F., and Poss, M. (2015). Robust constrained shortest path problems under budgeted uncertainty. *Networks*, 66(2):98–111.

Poggi, M. and Uchoa, E. (2014). New exact algorithms for the capacitated vehicle routing problem. In Toth, P. and Vigo, D., editors, *Vehicle routing: Problems, methods, and applications*, MOS/SIAM Ser Optim, pages 59–86.

Pugliese, L. D. P. and Guerriero, F. (2013). A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*, 62(3):183–200.

Pureza, V., Morabito, R., and Reimann, M. (2012). Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the VRPTW. *European Journal of Operational Research*, 218(3):636 – 647.

Richter, A. T. and Stiller, S. (2018). Robust strategic route planning in logistics. *Transportation Science*. Online First.

Righetto, G. M., Morabito, R., and Alem, D. (2016). A robust optimization approach for cash flow management in stationery companies. *Computers & Industrial Engineering*, 99:137 – 152.

Righini, G. and Salani, M. (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.

Sungur, I., Ordonez, F., and Dessouky, M. (2008). A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions (Institute of Industrial Engineers)*, 40(5):509–523.

Toth, P. and Vigo, D. (2014). *Vehicle Routing: Problems, Methods and Applications*. MOS-SIAM Series in Optimization, Second edition.

Vanderbeck, F. (2005). Implementing mixed integer column generation. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column Generation*, pages 331–358. Springer US.

Vanderbeck, F. and Wolsey, L. A. (2010). Reformulation and decomposition of integer programs. In Jünger, M., Liebling, T. M., Naddef, D., Nemhauser, G. L., Pulleyblank, W. R., Reinelt, G., Rinaldi, G., and Wolsey, L. A., editors, *50 Years of Integer Programming 1958-2008*, pages 431–502. Springer Berlin Heidelberg.