Exact approaches to the robust vehicle routing problem with time windows and multiple deliverymen

Jonathan De La Vegaa, Pedro Munari, Reinaldo Morabito

aProduction Engineering Department, Federal University of São Carlos, Rodovia Washington Luís - Km 235, São Carlos, SP 13565-905, Brazil

Abstract
This paper addresses the vehicle routing problem with time windows and multiple deliverymen (VRPTWMD) under uncertain demand as well as uncertain travel and service times. This variant is faced by logistics companies that deliver products to retailers located in congested urban areas, where service times are relatively long compared to travel times, and depend on the number of deliverymen assigned to each route. Differently from traditional variants, these service times show high variability, requiring an appropriate way of handling the related uncertainty. We extend two mathematical formulations to represent the VRPTWMD under uncertainty, using the robust optimization paradigm with budgeted uncertainty sets, and developed effective exact solution methods for solving each of them. The first formulation is a robust vehicle flow model solved by a tailored branch-and-cut algorithm that resorts to 1- and 2-path inequalities that we show how to effectively separate. The second formulation is a set partitioning model, for which we propose a branch-price-and-cut algorithm that relies on a robust resource-constrained elementary shortest path problem. The results of computational experiments using instances from the literature and risk analysis via a Monte Carlo simulation show the importance of incorporating uncertainties in the VRPTWMD, and indicate the sensitivity of decisions as well as cost and risk to the level of uncertainty in the input data.

Keywords: Routing, Multiple Deliverymen, Uncertainty, Robust Optimization, Branch-price-and-cut.

1. Introduction

In this paper, we study a variant of the vehicle routing problem (VRP) in which customer demand and travel and service times are uncertain. This variant generalizes the VRP with time windows and multiple deliverymen (VRPTWMD) and represents real-life distribution or collection of goods in highly congested urban areas, with limited parking lots at the customer’s locations (Pureza et al., 2012). As a consequence, service times may be long with respect to travel times, as the deliverymen visit customers on foot from the vehicle parking location, and depend on the number of deliverymen assigned to the route. The more deliverymen we assign to a route, the shorter the service times are, but the more expensive the route becomes. Indeed, the route cost consists of three components related to the number of vehicles, the number of deliverymen and the total distance traveled. Minimizing the overall cost must take into account the trade-off between these components (Munari and Morabito, 2018).
In practical settings, the actual values of some parameters of the problem, such as customer demand and travel and service times, are unknown at the time decisions need to be made. A common practice in these situations is to use the parameters’ corresponding nominal or expected value realizations (Gendreau et al., 2010). However, variations in these parameters around their nominal values can make the obtained solution infeasible. For this reason, in this paper we resort to a static robust optimization (RO) approach that leads to a solution immune to possible variations of uncertain parameters.

RO is a mathematical programming technique used to model and solve optimization problems with uncertain parameters, which has emerged as an approach to overcoming the difficulties and/or disadvantages involved in stochastic programming (SP) (Ben-Tal et al., 2015; Bertsimas et al., 2011). In SP, two possible difficulties can be noticed. The first entails the fact that probability distributions of uncertain data must be known a priori. In many real cases, it may be very difficult, if not impossible, to estimate sufficiently accurate probability distributions of data. Second, solution approaches based on SP are affected by the curse of dimensionality that impacts their computational tractability (Gounaris et al., 2016).

RO does not require probability distributions of uncertain parameters. Instead, these parameters are represented as bounded and independent random variables with possible realizations contained in a set called the uncertainty set (Bertsimas and Sim, 2003; Sniedovich, 2012). Thus, RO seeks a solution among those insensitive to uncertain data, i.e., candidate solutions that are feasible for all possible data realizations within the uncertainty set. For our robust approach, we choose the budgeted uncertainty set introduced in Bertsimas and Sim (2004) that maintains the advantages of the linearity of its deterministic counterpart and also allows analyzing the trade-off between the routing costs and the risks associated with their degrees of infeasibility (e.g., violating vehicle capacity and customer time windows).

To the best of our knowledge, the VRPTWMD under uncertainty has been addressed only by De La Vega et al. (2019) and without considering uncertainty of travel and service times. The authors proposed a robust optimization model for the VRPTWMD with demand uncertainty based on the dualization scheme commonly used in the RO literature. This introduces a large number of variables and constraints to the model, making it more difficult to solve by general-purpose optimization software. Even with the aid of initial solutions obtained by a heuristic that provides robust feasible solutions, the software terminated with large optimality gaps after 1 hour of execution for most problem instances with 25 customers considered in computational experiments. In addition, it is worth mentioning that no tailored exact solution approaches were proposed by the authors.

Given that travel and service times are important sources of uncertainty in practice, especially in the context in which the VRPTWMD appears, in this paper we develop formulations and solution methods that incorporate the uncertainty related to these sources. We propose two mathematical formulations to represent the robust VRPTWMD that are essentially different from the RO formulation proposed by De La Vega et al. (2019) and lead to more efficient solution approaches. The first formulation is a robust vehicle flow model obtained by linearizing the recursive equations that model the possible realizations of uncertain parameters, as introduced by Munari et al. (2019). This formulation is a robust counterpart that does not rely on the dualization scheme and hence has a reduced number of variables and constraints. To solve the robust vehicle flow model, we use a tailored branch-and-cut (BC) algorithm that relies on 1- and 2-path valid inequalities that are dynamically added to the problem, when violated. In particular, the 2-path inequalities separation routine uses a dynamic programming labeling algorithm to solve to optimality the robust traveling salesman problem with time windows (RTSPTW). To the best of our knowledge, this is the first tailored BC algorithm for the robust VRPTWMD. Additionally, we extend the heuristic proposed by De La Vega et al. (2019) to
obtain robust feasible solutions used as starting points in the proposed approaches.

The second formulation proposed in this paper is a set-partitioning model. We develop a branch-price-and-cut (BPC) algorithm to solve this model, in which the subproblem is a robust resource-constrained elementary shortest path problem. We include the recursive equations that model the possible realizations of uncertain parameters into the bidirectional labeling algorithm used to solve this subproblem. These recursive equations extend those described by Munari et al. (2019) to consider uncertain service times separately from uncertain travel times. Such separate treatment is important in the context of the VRPTWMD, as service times depend on the number of deliverymen assigned to routes and may be relatively large compared to travel times. To the best of our knowledge, this is the first specialized exact approach based on the column-generation technique proposed for the VRPTWMD under uncertainty.

The remainder of this paper is organized as follows. In Section 2, we review the use of RO to incorporate uncertainty into vehicle routing problems. In Section 3, we introduce the deterministic VRPTWMD and present the vehicle flow and set partitioning formulations for this problem variant. The robust version of the problem is considered in Section 4. The proposed BPC algorithm is presented in Section 5, and computational experiments are described in Section 6. Finally, conclusions and directions for future research are outlined in Section 7.

2. Related literature

The aim of this section is to highlight the most important features of several studies in the robust VRP literature. More complete reviews of VRP in uncertain contexts are provided by Oyola et al. (2016a,b); Gendreau et al. (2016). Sungur et al. (2008) were the first to introduce the term robust vehicle routing. In the cited paper, the two-index flow formulation with Miller-Tucker-Zemlin (MTZ) subtour elimination constraints was used to model the capacitated VRP (CVRP). Demand was considered to be uncertain and was represented by a finite number of realizations called scenarios. The latter were used to construct various uncertainty sets: convex hull, box and ellipsoid. Using these sets, three different robust formulations were obtained that were observed to be identical to one instance of the deterministic CVRP, in which customer demand values were modified by their respective worst-case realizations within each set. Thus, the resulting robust formulations provide fully conservative CVRP solutions. However, those robust formulations were more computationally treatable than SP formulations with only a few scenarios. Ordóñez (2010) discussed several mathematical formulations for the VRP with time windows (VRPTW) as well as the respective most appropriate uncertainty sets to develop strategies to adequately address the risk-cost trade-off.

In Agra et al. (2012) and Agra et al. (2013), the VRP with no capacity constraints yet with time windows was studied. In both studies, the problem was extended to the robust version where the travel times’ realizations were within the budgeted uncertainty set. Agra et al. (2012) presented a new formulation of the problem based on the concept of layered graphs. The underlying reason was that the duality technique commonly used in the RO approach could then be used directly. The new formulation led to a significant increase in the number of discrete variables as well as in that of problem constraints. For this reason, optimal solutions could only be obtained for small instances (with 10 and 20 customers) using commercial optimization solvers. Motivated by the results reported in Agra et al. (2012), Agra et al. (2013) proposed new formulations of the problem based on two RO approaches: static and adjustable. In the static approach, we consider only here-and-now decisions. On the other hand, in the adjustable one, in addition to the here-and-now decisions, wait-and-see decisions (recourse decisions) are also considered (Ben-Tal et al. 2009).
Depending on the robust approach used, specific exact methods were developed that successfully solved more realistic instances (with 50 customers) with computations taking a reasonable time.

Lee et al. (2012) discussed the VRP with deadlines. The authors considered customer demand and travel time uncertainty. Once again, the budgeted uncertainty set was used to represent the possible realizations of demand and travel times. To model the problem mathematically, a flow-based formulation was proposed and extended to a robust version to explicitly incorporate all combinations of worst-case realizations into the MTZ constraints for vehicle load and travel time. The resulting formulation is intractable by commercial optimization solvers even for instances with 10 customers. Therefore, a formulation based on set partitioning was also proposed, and the branch-and-price (BP) algorithm was used to solve it. The uncertainty sources were encapsulated in the shortest path problem with resource constraints that was solved by a bidirectional dynamic programming labeling algorithm. Numerical results have shown some benefit of using the static RO approach. However, the BP algorithm was unable to obtain optimal solutions to many instances with 25 customers within the time limit of 1 hour. The authors also emphasized that their algorithm could not be applied directly to the VRPTW.

Contributions to robust CVRP (RCVRP) modeling under uncertain demand were also presented by Gounaris et al. (2013, 2016). In both studies, customer demand was considered to be randomly varying within generic but polyhedral uncertainty sets. Gounaris et al. (2013) discussed various formulations of the problem, and for each of them, necessary and sufficient conditions were introduced to ensure that robust solutions were obtained. Considering the computational experiments, some formulations showed a clear advantage in terms of running time. Following the theoretical results of the previous study, the focus of Gounaris et al. (2016) was on the development of a solution methodology based on the meta-heuristic adaptive memory programming to solve the robust CVRP under uncertain demand belonging to a set of polyhedral uncertainty. Recently, Subramanyam et al. (2020) generalized these two works incorporating several realistic aspects, including the heterogeneity of the fleet. In addition to the uncertainty sets previously used by Gounaris et al. (2013, 2016), the authors proposed other three uncertainty sets and showed how these sets can be constructed using historical data. The resulting problem was solved via heuristic algorithms that use efficient strategies to assess the robustness of a given set of routes in each of the five uncertainty sets used.

Pessoa et al. (2018) addressed the RCVRP with uncertain demands belonging to the well-known budgeted polytope of Bertsimas and Sim (2003) and the partitioned budgeted polytope introduced by Gounaris et al. (2013). The authors reformulated the robust feasibility sets as the union of a limited number of 0-1 knapsack polytopes. The main advantage of this reformulation is that the robust pricing problem can be solved through a sequence of nominal pricing problems, which reduces the RCVRP to a nominal CVRP with heterogeneous fleet. The resulting problem was solved via an exact method based on BPC that incorporates capacity cuts adapted to the reformulated uncertainty sets. Their algorithm was able to solve to optimality all but one instance for the partitioned budgeted uncertainty set considered previously by Gounaris et al. (2013).

Lu and Gzara (2019) addressed the robust VRPTW with uncertainty of the demand only. The researchers adopted a budgeted uncertainty set and presented two robust formulations based on the deterministic flow and set partitioning formulations. Then, a BPC method was developed to solve the set partitioning formulation, in which the subproblem was the robust elementary shortest path problem with resource constraints. This subproblem was equivalently replaced by a series of its respective deterministic counterparts that were subsequently solved by the standard labeling algorithm. Moreover, several types of inequalities were discussed and extended to the robust case. In particular, the researchers proposed a new separation procedure for enhanced rounded capacity inequality. Computational experiments showed that the BPC algorithm was
successful in obtaining optimal solutions of instances with 25 and 50 customers within a reasonable time. Munari et al. (2019) also addressed the robust VRPTW, but considered uncertain demand and travel times. The authors proposed a compact formulation that, for the first time, used a linearization of recursive equations to incorporate robustness into a flow-based formulation, avoiding the need for the dualization scheme that is commonly used in the RO literature (Agra et al. 2012; De La Vega et al. 2019). The authors also proposed a set partitioning formulation and a branch-price-and-cut (BPC) method, in which the labeling algorithm was adapted and extended appropriately to introduce the robust resource consumption of load and time. Computational experiments showed that a commercial optimization solver could effectively solve the robust flow formulation with instances of 25 and 50 customers, whereas the BPC method solved instances with up to 100 customers.

As mentioned before, the VRPTWMD under uncertainty was first addressed by De La Vega et al. (2019), but only demand uncertainty was considered. The authors proposed a new deterministic formulation of the problem and obtained its robust counterpart using the dualization scheme. To improve the performance of a general-purpose optimization solver used to solve the RO formulation, the authors adapted the I1 heuristic of Solomon (1987) to consider possible realizations of demand, and used it to provide an initial robust feasible solution to the optimization solver. The results of computational experiments using instances with 25 customers showed that the heuristic led to significant reductions in the optimality gap of solutions obtained by the solver, although optimal solutions were not obtained for most instances. To the best of our knowledge, no other paper has addressed the VRPTWMD under uncertainty.

3. Definition of the problem

In the VRPTWMD, we assume that all n customers require either delivery or collection services but not both. Additionally, there is only one depot available that has enough goods to meet the demand of all customers, as well as a fleet of K identical vehicles of capacity Q and a number of deliverymen E that can be used to form the crews of vehicles. Following the common practice in the VRPTW literature that represents the problem using a network, we represent the depot by locations 0 and n + 1, where 0 defines the initial location and n + 1 denotes the final location. Thus, we have a total of n + 2 locations, counting n customers and two locations 0 and n + 1 representing the depot. We assume that each vehicle allows a maximum number L of deliverymen on board, which is small and no more than three due to the limited size of vehicle cabs in practice. For simplicity, the word mode is used to indicate the number of deliverymen assigned to a given vehicle.

Each of n + 2 locations has a nonnegative nominal demand \( \bar{q}_i \), where \( i = 0, \ldots, n + 1 \), and a nonnegative nominal service time \( \bar{s}_i^\ell \) that depends on the number of deliverymen or mode \( \ell \). Thus, time spent at a location is influenced by the number of deliverymen used to service it. Moreover, the time windows of each location are represented by \([w_a^i, w_b^i]\). In particular, for the initial and final depot locations, \( \bar{q}_0 = \bar{q}_{n+1} = 0 \), \( \bar{s}_0 = \bar{s}_{n+1} = 0 \) for all \( \ell \), \( w_a^0 = w_a^{n+1} = 0 \) and \( w_b^0 = w_b^{n+1} \) corresponds to a non-negative value that describes the maximum duration of a route or the working day of operators (drivers and deliverymen). Given the nonnegative nominal values of travel times \( (\bar{t}_{ij}) \) and the Euclidean distances \( (d_{ij}) \) between each pair of locations, which satisfy the triangle inequality, the VRPTWMD consists of designing routes and the visiting schedule with the additional information related to the number of deliverymen on board each of the vehicles, with the objective of minimizing the operational costs that depend on the number of routes generated, the number of deliverymen used and the total distance traveled.
Next, we present two formulations to mathematically represent the deterministic VRPTWMD. These formulations assume that the nominal or expected values of parameters such as demand, service and travel times are known in advance. The first formulation corresponds to the three-index flow formulation proposed in Pureza et al. (2012), while the second corresponds to the set partitioning formulation described in Morabito (2018), Alvarez and Munari (2017). Both are described using a complete and directed graph $G = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{0, 1, \ldots, n, n + 1\}$ is the set of nodes that correspond to customers’ premises (indexed by set $\mathcal{N}$) and to depot locations 0 and $n + 1$. In addition, $\mathcal{A} = \{(0, j) : j \in \mathcal{N}\} \cup \{(i, j) : i, j \in \mathcal{N} \text{ and } i \neq j\} \cup \{(i, n + 1) : i \in \mathcal{N}\}$ defines the set of arcs of the graph.

### 3.1 Three-index flow formulation

This subsection presents the mathematical formulation that we refer to as the three-index flow formulation (TIFF) because the variables that define the flow in the arcs of the graph have the additional index $\ell$ that indicates the mode of the operation of the vehicle. To present the formulation, it is necessary to define the following decision variables: 1) $x_{ij}^\ell$ is a binary variable that assumes the value of 1 if and only if there is a route operating in mode $\ell$ visiting customer $j$ immediately after visiting customer $i$; 2) $u_i^\ell$ is a continuous variable used to account for the vehicle load during operation in mode $\ell$ when leaving customer $i$, and 3) $w_i^\ell$ is a continuous variable that denotes the exact time the vehicle operating in mode $\ell$ starts its service of customer $i$. The TIFF can be written as follows:

$$
\begin{align*}
\min \quad & p_1 \sum_{\ell=1}^{\mathcal{L}} \sum_{j \in \mathcal{N}} x_{0j}^\ell + p_2 \sum_{\ell=1}^{\mathcal{L}} \sum_{j \in \mathcal{N}} \ell x_{0j}^\ell + p_3 \sum_{\ell=1}^{\mathcal{L}} \sum_{(i,j) \in \mathcal{A}} d_{ij} x_{ij}^\ell, \\
\text{s.t.} \quad & \sum_{(i,j) \in \mathcal{A}} x_{ij}^\ell = 1, \quad \forall j \in \mathcal{N}, \quad \forall \ell = 1, \ldots, \mathcal{L}, \\
& \sum_{(i,j) \in \mathcal{A}} x_{ij}^\ell = 1, \quad \forall i \in \mathcal{N}, \quad \forall \ell = 1, \ldots, \mathcal{L}, \\
& \sum_{(j,i) \in \mathcal{A}} x_{ij}^\ell = \sum_{(j,i) \in \mathcal{A}} x_{ji}^\ell, \quad \forall i \in \mathcal{N}, \quad \forall \ell = 1, \ldots, \mathcal{L}, \\
& u_i^\ell \geq u_i^{\ell-1} + q_j x_{ij}^\ell - Q(1 - x_{ij}^\ell), \quad \forall (i,j) \in \mathcal{A}, \quad \forall \ell = 1, \ldots, \mathcal{L}, \\
& q_i \leq u_i^\ell \leq Q, \quad \forall i \in \mathcal{N}, \quad \forall \ell = 1, \ldots, \mathcal{L}, \\
& w_i^\ell \geq w_i^{\ell-1} + (t_{ij} + s_i^\ell) x_{ij}^\ell - M_i(1 - x_{ij}^\ell), \quad \forall (i,j) \in \mathcal{A}, \quad \forall \ell = 1, \ldots, \mathcal{L}, \\
& w_i^\ell \leq w_i^{\ell-1}, \quad \forall i \in \mathcal{V}, \quad \forall \ell = 1, \ldots, \mathcal{L}, \\
& \sum_{j \in \mathcal{N}} \sum_{\ell=1}^{\mathcal{L}} \ell x_{0j}^\ell \leq \mathcal{E}, \\
& \sum_{j \in \mathcal{N}} \sum_{\ell=1}^{\mathcal{L}} x_{0j}^\ell \leq \mathcal{K}, \\
& \sum_{j \in \mathcal{N}} \sum_{\ell=1}^{\mathcal{L}} x_{0j}^\ell \geq \left\lfloor \frac{\sum_{i \in \mathcal{N}} q_i}{Q} \right\rfloor, \\
& \sum_{i \in \mathcal{N}} q_i \sum_{j \in \mathcal{V} \setminus \{0\}} x_{ij}^\ell \leq Q \sum_{j \in \mathcal{N}} x_{0j}^\ell, \quad \forall \ell = 1, \ldots, \mathcal{L}.
\end{align*}
$$
\[ \sum_{\ell=1}^{\mathcal{L}} \sum_{(i,j) \in \delta(S)} x_{ij}^{\ell} \geq k(S), \; S \subset \mathcal{N}, \; 2 \leq |S| \leq |\mathcal{N}|-1, \quad (3.13) \]

\[ x_{ij}^{\ell} \in \{0, 1\}, \; \forall (i,j) \in \mathcal{A}, \; \ell = 1, \ldots, \mathcal{L}. \quad (3.14) \]

Objective function (3.1) consists of minimizing the operational costs related to the number of vehicles and deliverymen in the solution, as well as the transportation costs in terms of the total traveled distance. Parameters \( p_1, p_2, \) and \( p_3 \) define the unit costs of vehicles, deliverymen, and distance, respectively, or a prioritization of these costs in the objective function if they are difficult to estimate (De La Vega et al., 2019). Constraints (3.2)-(3.4) correspond to flow constraints. They specify that all customers are to be visited and that the visit is made in a single mode.

If there is flow in arc \((i, j), i.e., if \( x_{ij}^{\ell} = 1 \) for a given \( \ell \), constraints (3.5) are responsible for accumulating the load collected at customer \( j \) together with the vehicle load from customer \( i \). Otherwise, constraints (3.5) become redundant. Because the vehicles have load capacity \( Q \), constraints (3.6) ensure that it is not exceeded, as well as that demand of all customers is met. Timing constraints can be observed in the set of constraints (3.7) and (3.8). If \( x_{ij}^{\ell} = 1 \), for a given \( \ell \), the exact time of arrival of the vehicle to customer \( j \) is determined by constraints (3.7), while constraints (3.8) ensure that service of the customer starts within the respective time window. It is important to note that the time determined by constraints (3.7) simply indicates the earliest time of arrival at the location of customer \( j \) but not necessarily the start time of service. It is possible that the vehicle arrives at the customer’s premises before the opening time of the customer’s time window, implying that the vehicle must necessarily wait until the customer is available to receive the service.

Constraints (3.9) ensure that no more deliverymen are used in a solution than those available at the depot. Constraints (3.10) and (3.11) impose upper and lower bounds, respectively, on the number of generated routes or used vehicles. The upper bound is the number of vehicles available in the depot, while the lower bound corresponds to the trivial bound given by the minimum number of vehicles required to satisfy the demand of all customers with respect to vehicle capacity only. In particular, constraints (3.12) correspond to valid inequalities for the VRPTWMD. They simply add all customers from various routes operating in the same mode and specify that the sum of the demand of these customers does not exceed the increased capacity multiplied by a factor related to the number of routes operating in the same mode.

Constraints (3.13) are the \( k \)-path inequalities proposed by Kohl et al. (1999) and extended here for the VRPTWMD. In this set of constraints, \( \delta \) is a subset of customers, \( k(\delta) \) is the minimum number of vehicles needed to service all customers in \( \delta \) without violating the capacity of the vehicles and time windows, and \( \delta(\delta) \) is the set of arcs entering the set \( \delta \). Note that the expression on the left-hand side of this set of constraints determines the flow entering in the set \( \delta \). These constraints are dynamically incorporated into the formulation via cuts. In this paper, we developed separation procedures for the \( k \)-path inequalities with the \( k(\delta) \) parameter set to 1 and 2. These separation procedures will be described later for the robust version of the problem. A special case of these inequalities is the subtour elimination constraints that is obtained for \( k(\delta) = 1 \). Finally, constraints (3.14) specify the type and domain of flow variables.

Another formulation used to represent the VRPTWMD corresponds to the set partitioning formulation (SPF). In this formulation, the decision variables correspond to routes operating in mode \( \ell \). The SPF determines the combination of routes with their respective numbers of deliverymen needed to serve all customers at the minimum operational cost. More details on the SPF for the VRPTWMD are available in Munari and Morabito (2018); Alvarez and Munari (2017).
4. Robust VRPTWMD

The robust VRPTWMD (RVRPTWMD) arises when the concepts of the robust optimization approach are used to address the random nature of some of its parameters. We now present in detail the modeling of uncertain parameters and discuss the budgeted uncertainty sets used to encapsulate the possible realizations of such parameters. Additionally, we define recursive equations that exploit a certain particular structure of these sets to evaluate the robust feasibility of solutions. Furthermore, the deterministic formulations described in Section 3 are extended to the robust version of the problem.

4.1. Uncertain parameters and sets

As mentioned earlier, in contrast to De La Vega et al. (2019) that assumes uncertainty of customer demands only, we also consider the uncertainties in parameters related to the service and travel times. In vehicle routing contexts, uncertainties in time-related parameters are typically incorporated into travel times. However, the customers’ service time in the VRPTWMD require special attention due to their high variability. In fact, service times are not just greatly influenced by customer demands, but also by the location and the traffic conditions close to their locations. Thus, ignoring service time uncertainties increases the number of deliverymen assigned to the routes, the distance from the vehicle parking location to the customer’s location, and the traffic conditions close to their locations. Thus, ignoring service time uncertainties increases the chance of violation of the customer’s time windows, which is highly undesirable in the RVRPTWMD. For this reason, we consider the uncertainties in service times separated from the uncertainties in travel times.

The random parameters \( \tilde{q}_i, \hat{t}_{ij} \), and \( \bar{s}_i^j \) are modeled as bounded and independent random variables with variation intervals corresponding to \( [\tilde{q}_i - \hat{q}_i, \hat{q}_i + \hat{q}_i], [\hat{t}_{ij} - \bar{t}_{ij}, \bar{t}_{ij} + \hat{t}_{ij}] \) and \( [\bar{s}_i^j - \bar{s}_i^j, \bar{s}_i^j + \bar{s}_i^j] \), respectively. Nonnegative parameters \( \hat{q}_i, \hat{t}_{ij} \) and \( \bar{s}_i^j \) are the maximum deviations from the corresponding parameters’ nominal values. We associate other random variables \( \xi_i, \phi_{ij} \) and \( \zeta_i^j \) (called primitive random variables) with each of the random variables assuming values between -1 and 1. Accordingly, the original random variables can be written in terms of primitive ones as follows: \( \tilde{q}_i = \hat{q}_i + \bar{q}_i \xi_i, \hat{t}_{ij} = \bar{t}_{ij} + \hat{t}_{ij} \phi_{ij} \) and \( \bar{s}_i^j = \bar{s}_i^j + \hat{s}_i^j \zeta_i^j \). We use \( \xi, \phi \) and \( \zeta \) to denote vectors with components that correspond to primitive uncertain variables \( \xi_i, \phi_{ij} \) and \( \zeta_i^j \), respectively.

The uncertainty sets correspond to budgeted uncertainty sets (Bertsimas and Sim, 2004; Bertsimas et al., 2011). They are constructed for the primitive uncertain variables and simply restrict the number of these variables allowed to deviate from their nominal values. The budgeted uncertainty sets of primitive uncertain variables \( \xi_i, \phi_{ij} \) and \( \zeta_i^j \) can be written as follows:

\[
\mathcal{U}^q = \left\{ \xi \in \mathbb{R}^{|\mathcal{V}|} : -1 \leq \xi_i \leq 1, i \in \mathcal{N}: \sum_{i \in \mathcal{N}} |\xi_i| \leq \Gamma^q \right\}, \tag{4.1}
\]

\[
\mathcal{U}^{(t,s)} = \left\{ \phi, \zeta \in \mathbb{R}^{|\mathcal{A}|} \times \mathbb{R}^{|\mathcal{V}| \times \mathcal{L}} : -1 \leq \phi_{ij} \leq 1, \forall (i,j) \in \mathcal{A}; -1 \leq \zeta_i^j \leq 1, i \in \mathcal{N}, \ell = 1, \ldots, \mathcal{L}; \sum_{(i,j) \in \mathcal{A}} |\phi_{ij}| + \sum_{\ell=1}^{\mathcal{L}} \sum_{i \in \mathcal{N}} |\zeta_i^\ell| \leq \Gamma^{(t,s)} \right\}. \tag{4.2}
\]

Sets \( \mathcal{U}^q \) and \( \mathcal{U}^{(t,s)} \) depend on parameters \( \Gamma^q \) and \( \Gamma^{(t,s)} \), respectively, which define the uncertainty budget for the load and time resources. For simplicity, we assume that these parameters are nonnegative integers. The larger the values assigned to these parameters are, the greater the number of primitive uncertain variables
represented in the components of vectors $\xi$, $\eta$ and $\zeta$ that are allowed to vary and, therefore, the larger the conservatism level of the RO approach. We highlight two cases:

1) If the minimum values (equal to 0) are assigned to both parameters $\Gamma^q$ and $\Gamma^{(t,s)}$, we obtain the nominal problem in which the decisions are optimally defined under the premise that the realization of the uncertain parameters will be exactly equal to their nominal/expected values, leading to solutions with high levels of risk.

2) If the maximum values are assigned to both parameters $\Gamma^q$ and $\Gamma^{(t,s)}$ (i.e., as in the method of Soyster (1973)), the decisions correspond to solving the deterministic version of an instance of the problem, assuming that the uncertain parameters reach their worst-case values. Thus, fully conservative solutions with zero risk levels are obtained.

Note that in both cases, the RVRPTWMD results in an instance of the same problem, though the problem’s deterministic version arises. However, it is unlikely that practical environments where the two abovementioned cases occur will be found. For this reason, appropriate values must be assigned to parameters $\Gamma^q$ and $\Gamma^{(t,s)}$ to obtain solutions that have costs worse than the optimal cost of the nominal/deterministic problem yet have more acceptable levels of risk. Since the random variables related to service and travel times compete for the same time resource, we opted to encapsulate the realizations of their corresponding primitive uncertain variables into a single budgeted uncertainty set, denoted by $U^{(t,s)}$. Hence, a single uncertainty budget ($\Gamma^{(t,s)}$) was used to control the robustness of routes relative to the time resource’s consumption.

4.2. Evaluating robust feasibility

Let $\mathcal{R}$ be the set of routes with their respective modes of operation in a given solution. Each route $r \in \mathcal{R}$, which visits $k + 1$ nodes and operates in mode $\ell$, is represented by vector $(v_1, v_2, \ldots, v_k, v_{k+1})$ with $v_1 = 0$ and $v_{k+1} = n + 1$. We follow the idea used by Agra et al. (2013); Alves Pessoa et al. (2015); Munari et al. (2019) to check robust feasibility recursively on each route $r \in \mathcal{R}$ with respect to the capacity and time window constraints. Defining $u_{v_i, \gamma}$ as the maximum load in the vehicle leaving customer $v_i$, given that the demand of $\gamma$ customers with $\gamma = 0, 1, \ldots, \Gamma^q$ in partial route $r_{v_i} = (v_0, v_1, \ldots, v_i)$ assumes the worst-case value, the recursion equations for the load can be written as:

$$u_{v_{i+1}, \gamma} = \begin{cases} u_{v_i, \gamma} + \bar{q}_{v_{i+1}}, & \text{if } \gamma = 0, \\ \max\{u_{v_i, \gamma} + \bar{q}_{v_{i+1}}, u_{v_i, \gamma-1} + \bar{q}_{v_{i+1}} + \hat{q}_{v_{i+1}}\} & \text{otherwise}, \end{cases}$$

(4.3)

where $u_{v_i, \gamma} = 0$ for $\gamma = 0, 1, \ldots, \Gamma^q$. Note that for $\gamma = 0$, we have the deterministic consumption of the vehicle load leaving customer $v_{i+1}$. In another case, equation (4.3) determines the value that assumes customer demand $v_{i+1}$ in the worst-case scenario.

Recursive equations can also be constructed to quantify the time resource’s consumption. However, in contrast to load consumption, four cases should be considered: 1) waiting at location $v_{i+1}$ absorbs the two possible realizations of the travel and service times, 2) both travel and service times assume their nominal values, 3) exactly one of the two parameters reaches its worst-case value, and 4) both travel and service times assume their respective worst-case values. In case 1), it is observed that travel and service times with the greatest deviations do not necessarily appear in the worst-case scenario, as it is possible that the waiting time absorbs them (see a numerical example in Munari et al. (2019)). Let $w_{v_i, \gamma}$ be the earliest possible instant of time of beginning the service at location $v_i$ by a route operating in mode $\ell$, given that the travel and
service times of \( \gamma \) arcs or nodes in the route before customer \( v_i \) reach their worst-case values. The recursion equations for this time can be written as follows:

\[
 w_{v_i+1} = \begin{cases} 
 \max \left\{ w_{v_i+1}^a, w_{v_i} + \bar{s}_{v_i} + \bar{t}_{v_i,v_{i+1}} \right\} & \text{if } \gamma = 0, \\
 \max \left\{ w_{v_i+1}^a, w_{v_i} + \bar{s}_{v_i} + \bar{t}_{v_i,v_{i+1}}, w_{v_i} - 1 + \bar{s}_{v_i} + \bar{t}_{v_i,v_{i+1}}, w_{v_i} + \bar{s}_{v_i} + \bar{t}_{v_i,v_{i+1}}, w_{v_i} - 1 + \bar{s}_{v_i} + \bar{t}_{v_i,v_{i+1}} \right\} & \text{if } \gamma = 1, \\
 \max \left\{ w_{v_i+1}^a, w_{v_i} + \bar{s}_{v_i} + \bar{t}_{v_i,v_{i+1}}, w_{v_i} - 1 + \bar{s}_{v_i} + \bar{t}_{v_i,v_{i+1}}, w_{v_i} + \bar{s}_{v_i} + \bar{t}_{v_i,v_{i+1}} \right\} & \text{otherwise}, 
\end{cases}
\]

where \( w_{v_i} = 0 \) for \( \gamma = 0, 1, \ldots, \Gamma^q \). Similarly, for \( \gamma = 0 \) we have the deterministic consumption; however, in this case the consumption is that of the time resource. In the other cases, expression (4.4) also determines the value that assumes the service time of customer \( v_i \) and the travel time between locations of customers \( v_i \) and \( v_{i+1} \) in the worst-case scenario. For route \( r \) to be robust, it must satisfy \( w_{v_i} \leq Q \) for all \( \gamma = 0, \ldots, \Gamma^q \) and \( i = 1, \ldots, k \), and \( w_{v_i} \leq w_{v_i}^b \) for all \( \gamma = 0, \ldots, \Gamma^q \) and \( i = 1, \ldots, k \). Note that expression (4.4) extends the definition in [Munari et al. 2019] by considering the uncertain service times separately from the uncertain travel times. This approach is important for the RVRPTWMD, as service times are dependent on the mode of routes unlike the case of the robust VRPTW.

### 4.3. Robust flow formulation

In this subsection, robustness is incorporated into the TIFF (3.1)-(3.14) without needing to use the duality technique that is quite common in the RO literature. The robust TIFF (RTIFF) is based on using a linearization of the recursion Equations (4.3) and (4.4) for the load and time resources, respectively. The continuous variables for the load and time of formulation (3.1)-(3.14) are appropriately redefined as follows: 1) \( u_{r}^\ell \) is a continuous variable that defines the load collected/delivered by the vehicle operating in mode \( \ell \) after leaving customer \( i \), given that \( \gamma \) customer demands, including that of customer \( i \), can deviate from their nominal values, and 2) \( w_{r}^\ell \) is a continuous variable defining the worst-case earliest time of starting the service of customer \( i \) by a route operating in mode \( \ell \), given that the travel and service times of \( \gamma \) arcs or nodes formed by the sequence of visits to customers before customer \( i \) can deviate from their nominal values. Therefore, the mathematical model that incorporates robustness in the flow formulation and that determines the set of minimum-cost routes can be written as follows:

\[
\begin{align*}
\min \quad & p_1 \sum_{\ell=1}^L \sum_{j \in X} x_{ij}^\ell + p_2 \sum_{\ell=1}^L \sum_{j \in X} \ell x_{ij}^\ell + p_3 \sum_{\ell=1}^L \sum_{(i,j) \in a} d_{ij} x_{ij}^\ell. \\
\text{s.t.} \quad & \text{constraints (3.2)-(3.4) and (3.9)-(3.14) hold,} \\
& \quad u_{r}^\ell \geq \bar{u}_{r}^\ell + \bar{q}_{ij} x_{ij}^\ell - Q(1 - x_{ij}^\ell), \quad \forall (i,j) \in \mathcal{A}, \quad \ell = 1, \ldots, \mathcal{L}, \quad \gamma = 0, \ldots, \Gamma^q. \\
& \quad u_{r}^\ell \geq u_{r}^{\ell-1} + (q_{ij} + \bar{q}_{ij}) x_{ij}^\ell - Q(1 - x_{ij}^\ell), \quad \forall (i,j) \in \mathcal{A}, \quad \ell = 1, \ldots, \mathcal{L}, \quad \gamma = 1, \ldots, \Gamma^q. \\
& \quad q_{ij} \leq \bar{q}_{ij} \leq Q, \quad \forall (i,j) \in \mathcal{V}, \quad \ell = 1, \ldots, \mathcal{L}, \quad \gamma = 0, \ldots, \Gamma^q. \\
& \quad w_{r}^\ell \geq w_{r}^\ell + (\bar{s}_{ij} + \bar{t}_{ij}) x_{ij}^\ell - M_{ij}(1 - x_{ij}^\ell), \quad \forall (i,j) \in \mathcal{A}, \quad \ell = 1, \ldots, \mathcal{L}, \quad \gamma = 0, \ldots, \Gamma^{t,s} \quad (4.9). \\
& \quad w_{r}^\ell \geq w_{r}^{\ell-1} + (\bar{s}_{ij} + \bar{t}_{ij} + \bar{u}_{ij}) x_{ij}^\ell - M_{ij}(1 - x_{ij}^\ell), \quad \forall (i,j) \in \mathcal{A}, \quad \ell = 1, \ldots, \mathcal{L}, \quad \gamma = 1, \ldots, \Gamma^{t,s}. \\
& \quad w_{r}^\ell \geq w_{r}^{\ell-1} + (\bar{s}_{ij} + \bar{t}_{ij} + \bar{u}_{ij}) x_{ij}^\ell - M_{ij}(1 - x_{ij}^\ell), \quad \forall (i,j) \in \mathcal{A}, \quad \ell = 1, \ldots, \mathcal{L}, \quad \gamma = 2, \ldots, \Gamma^{t,s}. \\
& \quad w_{r}^\ell \leq w_{r}^b, \quad \forall i \in \mathcal{V}, \quad \gamma = 0, \ldots, \Gamma^{t,s}. \\
\end{align*}
\]
The objective function is the same as in the TIFF and consists of minimizing the costs related to the number of routes in the solution, the number of deliverymen and the total distance traveled. Constraints (4.6) are the same as in the deterministic counterpart of the flow formulation and were explained in Subsection 3.1. The set of constraints (4.7)-(4.9) considers the customer demand uncertainties, while set (4.10)-(4.14) takes into consideration the uncertainties of travel and service times. They correspond to linearizations of Equations (4.3) and (4.4), respectively. Note that the robust flow formulation is reduced to the deterministic flow formulation if the value of 0 is assigned to both parameters $\Gamma^q$ and $\Gamma^{(t,s)}$. In this case, constraints (4.8) and (4.11)-(4.13) are unneeded in the formulation.

Unlike the robust assignment formulation (RAF) proposed in De La Vega et al. (2019), the RTIFF does not involve the vectors associated with the dual protection variables, as it is unnecessary to resort to the duality technique to take robustness into account. This brings advantages over the deterministic flow formulation, as there is no increase in the number of binary variables, which could lead to better performance of general-purpose optimization software than that reported in De La Vega et al. (2019) for the RAF.

4.4. Separation of 1- and 2-path inequalities

We now present the separation procedures for the 1- and 2-path inequalities, given a flow solution $x^\star$. As mentioned before, 1-path inequalities are a special case of the subtour elimination constraints and, therefore, their separation uses the min-cut problem. Let $f$ be the optimal value of the min-cut problem. If $f < 1$, then the 1-path inequality is violated and, therefore, we impose it via a cut in the RTIFF. To separate 2-path inequalities, we search for sets $\delta$ in the flow solution $x^\star$ such that $x(\delta) < 2$, using the greedy heuristic proposed by Kohl et al. (1999). For each set $\delta$, we determine the minimum number of vehicles $k(\delta)$ required to service all customers in $\delta$ without violating vehicle capacity and time windows, and then we check if $x(\delta) < k(\delta)$. The parameter $k(\delta)$ for the RVPTWMD under the budgeted uncertainty set is determined as follows:

$$k(\delta) = \left\lceil \frac{\sum_{i \in \delta} q_i + T_{\text{max}}}{Q} \right\rceil$$

(4.15)

where $T_{\text{max}}$ is the sum of the $\min\{\|\delta\|, \Gamma^q\}$ largest deviations from the customer demands in $\delta$. If $x(\delta) < k(\delta)$, we conclude that the 2-path inequality is violated and then proceed by adding the corresponding cut in the RTIFF. Otherwise, we define an instance of the robust traveling salesman problem with time windows (RTSPTW) using only the customers in $\delta$ and arcs $(i,j)$ in $\delta(\delta) = \{(0,j) : j \in \delta\} \cup \{(i,j) : i,j \in \delta \text{ and } i \neq j\} \cup \{(i,n+1) : i \in \delta\}$. All arc costs are fixed at -1 and we assume that the customers are served with the maximum number of deliverymen allowed per route ($L$). By modeling the RTSPTW this way, it can be effectively solved by the same labeling algorithm described in Subsection 5.2. Let $f$ be the RTSPTW optimal value. If $-f < |\delta| + 1$, the RTSPTW is infeasible, implying that a minimum of $k(\delta) = 2$ vehicles are required to serve all customers in $\delta$. Thus, we proceed by adding the corresponding cut in the RTIFF.

4.5. Robust heuristic I1

Based on De La Vega et al. (2019), we extend the heuristic I1 (HI1) proposed by Solomon to take into account robustness to vehicle capacity and time window constraints. The purpose of the resulting method, named RHI1, is to find good-quality robust solutions that are used as initial solutions of a general-purpose optimization solver in an attempt to accelerate their convergence.

In the RHI1 method, the routes are initialized according to the criterion of the farthest customer, and it is assumed that the journeys along the routes begin with a single deliveryman. For a given iteration of
the heuristic, placement of any non-routed customer \( j \) is tested at all positions of the current partial route, and the customer is inserted into a certain position such that the resulting insertion is robust feasible and minimizes a given indicator involving the increase of the distance and duration of the generated partial route. An important issue here concerns the way in which the robustness of insertion is evaluated. Given a partial route \( r_v = (v_1, v_2, \ldots, v_i) \), a non-routed customer \( j \) and a value of budget \( \Gamma^q \), robustness to vehicle capacity constraints is evaluated as \( D_{v_i}(j) \leq Q \), where \( D_{v_i}(j) \) corresponds to the sum of \( \min\{i + 1, \Gamma^q\} \) largest deviations of customer demand along partial route \( r_v \) and customer \( j \). To evaluate the robustness of time window constraints regarding this insertion, we simply apply the recursion given in Equation (4.4).

If robust insertions are no longer possible, we verify if at least one non-routed customer can be inserted by increasing the number of deliverymen by 1 for the current partial route. If it can, then we increase this number and repeat the procedure. This procedure is performed until the maximum number of deliverymen permitted per route (\( L \)) is reached. If no more feasible robust insertions are possible (even by increasing the number of deliverymen on the partial route), then a new route is initialized. In this study, the maximum number of deliverymen allowed per route is three, i.e., \( L = 3 \).

### 4.6. Robust set partitioning formulation

The SPF for the RVRPTWMD involves decision variables that describe routes operating in some mode \( \ell \). Thus, we define a binary decision variable \( \lambda^\ell_r \) that has the value of 1 if and only if route \( r \in R^\ell \) in mode \( \ell \) entails visiting at least one customer, where \( R^\ell \) corresponds to the set of robust feasible routes in mode \( \ell \). Given a route \( r \in R^\ell \), parameter \( a^\ell_{ir} \) is equal to 1 if customer \( i \) is included in the route with \( \ell \) deliverymen. The mathematical formulation based on the SPF that defines robust feasible routes of minimum cost is written as follows:

\[
\begin{align*}
\text{min} & \quad \sum_{\ell=1}^L \sum_{r \in R^\ell} c^\ell_r \lambda^\ell_r, \\
\text{s.t.} & \quad \sum_{\ell=1}^L \sum_{r \in R^\ell} a^\ell_{ir} \lambda^\ell_r = 1, \quad i \in \mathcal{N}, \\
& \quad \sum_{\ell=1}^L \sum_{r \in R^\ell} \ell \lambda^\ell_r \leq E, \\
& \quad \sum_{\ell=1}^L \sum_{r \in R^\ell} \lambda^\ell_r \leq K, \\
& \quad \lambda^\ell_r \in \{0, 1\}, \quad \ell = 1, \ldots, L, \quad r \in R^\ell.
\end{align*}
\]

Objective function (4.16) consists of minimizing the costs associated with the selected routes. Parameter \( c^\ell_r \) defines the cost of using route \( r \) in mode \( \ell \). This cost is determined as follows:

\[
c^\ell_r = p_1 + p_2 \ell + p_3 \sum_{i=1}^k d_{v_i, v_{i+1}},
\]

where \( p_1, p_2 \) and \( p_3 \) are defined as before. Constraints (4.17) require that customer \( i \) be included in just one of selected routes. Constraints (4.18) ensure that the total number of deliverymen used in a solution does not exceed the maximum quantity available at the depot. Finally, the binary domain of the decision variables is imposed in constraints (4.20).
A drawback of formulation (4.16)-(4.20) entails the typically huge number of routes in each set $R_\ell$ for all $\ell = 1, \ldots, L$. Another drawback is that the routes in each set are unknown a priori, and any strategy for determining them can be as difficult as solving the optimization problem. For this reason, we resort to the column generation technique within a BPC framework to solve the set partitioning formulation (4.16)-(4.20).

5. Branch-price-and-cut algorithm

In this section, we describe the main components of the BPC algorithm proposed to solve the formulation (4.16)-(4.20). Among the components are the column generation (CG) technique, a robust bidirectional labeling algorithm, a cutting plane strategy and branching rules.

5.1. Column generation technique

The CG technique is performed on a relaxation of the set partitioning formulation (4.16)-(4.20), called the restricted master problem (RMP). The RMP considers set $\tilde{R}_\ell \subset R_\ell$ that contains relatively few columns of set $R_\ell$ and disregards the requirement that decision variables $\lambda_\ell r$ be integers. Given sets $\tilde{R}_\ell$ for all $\ell = 1, \ldots, L$, the RMP can be written as

\[
\min \sum_{\ell=1}^{L} \sum_{r \in \tilde{R}_\ell} c_{\ell r} \lambda_\ell r \quad (5.1)
\]

\[
s.t. \sum_{\ell=1}^{L} \sum_{r \in \tilde{R}_\ell} a_{\ell ri} \lambda_\ell r = 1, \quad i \in N, \quad (5.2)
\]

\[
\sum_{\ell=1}^{L} \sum_{r \in \tilde{R}_\ell} \Omega_{r} \lambda_\ell r \leq E, \quad (5.3)
\]

\[
\sum_{\ell=1}^{L} \sum_{r \in \tilde{R}_\ell} \lambda_\ell r \leq K, \quad (5.4)
\]

\[
\lambda_\ell r \geq 0, \quad \ell = 1, \ldots, L, \quad r \in \tilde{R}_\ell. \quad (5.5)
\]

The initial routes of sets $\tilde{R}_\ell$ can be generated by using heuristic procedures or considering only trivial routes, i.e., an exclusive route for each customer. The CG technique can be regarded as an extension of the simplex method to solve linear optimization problems with a huge number of continuous variables. For a given iteration of the CG algorithm, the decision variables of the RMP that assume strictly positive values ($\lambda_\ell r > 0$) form a basis. The question at this point is if the current basis is optimal. Therefore, the pricing subproblem is used to determine if there are routes in $R_\ell \setminus \tilde{R}_\ell$, for each $\ell = 1, \ldots, L$, such that their corresponding columns have negative reduced costs. If the pricing subproblem returns routes (columns) satisfying this condition, then the corresponding variables should be included in the basis, and therefore they are added to the respective $\tilde{R}_\ell$. Next, the resulting RMP is solved, and the described process is repeated. This iterative procedure is followed until the pricing subproblem does not result in columns with negative reduced costs.

Similarly to the simplex method, the pricing step in a given iteration of the CG algorithm also requires dual information from the corresponding RMP. Let $\bar{\pi} \in \mathbb{R}^n$, $\bar{v} \in \mathbb{R}^+$ and $\bar{\vartheta} \in \mathbb{R}^+$ be the dual solutions associated with constraints (5.2), (5.3) and (5.4) of the RMP, respectively, where $n = |N|$ denotes the number
of customers. Then, for a given iteration of the CG algorithm, the pricing subproblem corresponding to mode $\ell$ can be written as

$$z^\ell_{SP}(\bar{\pi}, \bar{v}, \bar{\vartheta}) = \min \left\{ (p_1 - \bar{\vartheta}) \sum_{j \in \mathcal{N}} x^\ell_{r_0 j} + (p_2 - \bar{v}) \ell \sum_{j \in \mathcal{N}} x^\ell_{r_0 j} + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} (p_3 c^\ell_{ij} - \bar{\pi}_i) x^\ell_{r_0 ij} \mid r \in \mathcal{R}^\ell \right\}.$$  \hspace{1cm} (5.6)

In the optimization problem (5.6), $x^\ell_{r_0} = \{x^\ell_{r_0 ij} \mid i,j \in \mathcal{N}\}$ corresponds to a binary vector with components that denote the binary flow variable of the problem and have the value of 1 if and only if the vehicle designated to travel along route $r$ in mode $\ell$ visits customers $i$ and $j$ in sequence. For each mode $\ell$, this problem corresponds to the robust counterpart of the shortest path problem with resource constraints (SPPRC), as set $\mathcal{R}^\ell$ contains all robust routes operating in mode $\ell$. The SPPRC has already been shown to be NP-hard \cite{Dror1994}, and therefore its robust version is also in the same class, as it is an extension of the SPPRC. Various formulations and solution methods have been proposed to solve the SPPRC exactly \cite{Irnich2006, Drezel2014, Lozano2016} and currently the labeling algorithm based on dynamic programming is one of the most effective in practice. The labeling algorithm used here is referred to as the robust labeling algorithm since static robust optimization is the paradigm used to incorporate uncertainty in the method. In the next section, we describe an extension of the algorithm to the robust version of the SPPRC in the context of the VRPTWMD.

5.2. Robust labeling algorithm

The standard labeling algorithm is used to solve various types of shortest path problems \cite{Irnich2005}. This algorithm represents the partial paths from the initial depot 0 of graph $\mathcal{G}$ using arrays called labels. Starting with the initial label $\mathcal{F}_0$ at the initial depot, the algorithm enumerates partial paths by propagating labels across graph $\mathcal{G}$ using extension functions. The paths starting from the depot and ending at the same node are compared using the dominance rules to eliminate paths for which it can be proven that they cannot lead to an optimal route or a route starting from the initial depot 0 and ending at the final depot $n + 1$. The robust labeling algorithm proceeds similarly to the standard algorithm, with the difference that robustness must be taken into account during propagation of partial paths.

5.2.1. Label choice

A partial path $r^\ell_{v_i}$ ending at node $v_i$ and operating at mode $\ell$ is represented by a label with $1 + n + (\Gamma^q + 1) + (\Gamma^{(\ell,s)} + 1)$ components that are described as follows:

- A component for reduced cost $\tilde{C}^\ell_{v_i}$.

- $n$ binary components $V^\ell_{v_i,1}, \ldots, V^\ell_{v_i,j}, \ldots, V^\ell_{v_i,n}$ that indicate using the value of 1 if the corresponding customer is part of partial path $r^\ell_{v_i}$. Component $V^\ell_{v_i,j}$, for all $j \in \mathcal{N}$, can also have the value of 1 if customer $j$ is not reached, i.e., if an extension of partial route $r^\ell_{v_i}$ to that customer leads to infeasibility of at least one resource.

- $\Gamma^q + 1$ components $Q^\ell_{v_i,\gamma}$, where $\gamma = 0, \ldots, \Gamma^q$, representing vehicle load when the vehicle is leaving customer $v_i$, assuming that demand values of $\gamma$ customers in partial path $r^\ell_{v_i} = (v_0, v_1, \ldots, v_i)$ attain their respective worst-case values.
5.2. Extension functions

Deterministic components $\tilde{C}_{v_i}^{\ell}, V_{v_i,1}^{\ell}, \ldots, V_{v_i,n}^{\ell}$ are equivalent to those of the standard labeling algorithm [Irnich and Desaulniers, 2005]. On the other hand, $Q_{v_i,\gamma}^{\ell}$ and $T_{v_i,\gamma}^{\ell}$ are defined similarly to variables of recursive Equations (4.3) and (4.4), respectively. Therefore, these parameters are used to take into consideration the robustness of load and time resources. Using components $\tilde{C}_{v_i}^{\ell}, V_{v_i,1}^{\ell}, \ldots, V_{v_i,n}^{\ell}, Q_{v_i,\gamma}^{\ell}$ and $T_{v_i,\gamma}^{\ell}$, the label associated with partial path $r$ with $\ell$ deliverymen can be then represented as vector $\mathcal{F}_{v_i}^{\ell} = (\tilde{C}_{v_i}^{\ell}, V_{v_i,1}^{\ell}, \ldots, V_{v_i,n}^{\ell}, Q_{v_i,\gamma}^{\ell}, T_{v_i,0}^{\ell}, \ldots, T_{v_i,\Gamma(\ell,s)}^{\ell})$.

5.2.2. Extension functions

Given a node $v_j$ and label $\mathcal{F}_{v_i}^{\ell}$ associated with partial path $r_{v_i}^{\ell}$ with $V_{v_i,v_j}^{\ell} = 0$, the extension functions for components $Q_{v_i,\gamma}^{\ell}$ and $T_{v_i,\gamma}^{\ell}$ operate the same way as in Equations (4.3) and (4.4), respectively. Reduced cost $\tilde{C}_{v_j}^{\ell}$ is extended as $\tilde{C}_{v_j}^{\ell} = \tilde{C}_{v_j}^{\ell} + p_d_{v_i,v_j} - \tilde{\pi}_{v_i}$, where $\tilde{\pi}_{v_i}$ corresponds to the dual solutions associated with constraints (1.17) from the RMP. Components $V_{v_i,k}^{\ell}$ for all $k \in \mathcal{N}$ are extended using the approach similar to that used in the standard labeling algorithm. Finally, after the extension the resulting label $\mathcal{F}_{v_i}^{\ell}$ is accepted as a robust feasible extension if $V_{v_i,v_j}^{\ell} = 0, Q_{v_i,\gamma}^{\ell} \leq Q$ for $\gamma = 0, \ldots, \Gamma^q$ and $T_{v_i,\gamma}^{\ell} \leq w^b_{v_j}$ for $\gamma = 0, \ldots, \Gamma(\ell,s)$.

5.2.3. Robust dominance rule

Certain rules are used to eliminate partial paths known not to be part of the optimal path being sought. The basic rule is the dominance rule. Let $r_{v_i}^{(1,1)}$ and $r_{v_i}^{(1,2)}$ be two feasible robust partial paths operating in mode $\ell$ and ending at the same node $v_i$, represented by labels $\mathcal{F}_{v_i}^{(1,1)}$ and $\mathcal{F}_{v_i}^{(1,2)}$, respectively. We say that $\mathcal{F}_{v_i}^{(1,1)}$ dominates $\mathcal{F}_{v_i}^{(1,2)}$ if the following conditions are true: (C1) any robust feasible extension $e$ of $\mathcal{F}_{v_i}^{(1,2)}$ ending in a given node $v_j$ is also robust feasible for $\mathcal{F}_{v_i}^{(1,1)}$, and (C2) for any extension $e$, the inequality $\tilde{C}_{v_j}^{1,\mathcal{F}} \leq \tilde{C}_{v_j}^{2,\mathcal{F}}$ is satisfied, where $\tilde{C}_{v_j}^{h,\mathcal{F}}$ defines the reduced cost obtained by extending partial path $r_{v_i}^{(h)}$ for $h = 1, 2$.

As mentioned before, the labeling algorithm described in this paper corresponds to an extension of the standard labeling algorithm since a robust version of SPPRC is solved here. Therefore, the dominance rule has to be adapted to take robustness into account. The dominance rule for the robust labeling algorithm is stated in Proposition 5.1.

**Proposition 5.1** (Robust dominance rule). If labels $\mathcal{F}_{v_i}^{(1,1)}$ and $\mathcal{F}_{v_i}^{(1,2)}$ associated with robust feasible partial paths $r_{v_i}^{(1,1)}$ and $r_{v_i}^{(1,2)}$, respectively, ending at customer $v_i$ are such that

i) $\tilde{C}_{v_j}^{1,\mathcal{F}} \leq \tilde{C}_{v_j}^{2,\mathcal{F}}$,

ii) $V_{v_j,1}^{1,\mathcal{F}} \leq V_{v_j,1}^{2,\mathcal{F}}$ for all $j \in \mathcal{N}$,

iii) $Q_{v_j,\gamma}^{1,\mathcal{F}} \leq Q_{v_j,\gamma}^{2,\mathcal{F}}$ for all $\gamma = 0, \ldots, \Gamma^q$, and

iv) $T_{v_j,\gamma}^{1,\mathcal{F}} \leq T_{v_j,\gamma}^{2,\mathcal{F}}$ for all $\gamma = 0, \ldots, \Gamma(\ell,s)$,

then, label $\mathcal{F}_{v_i}^{(1,1)}$ dominates label $\mathcal{F}_{v_i}^{(1,2)}$ in terms of C1 and C2.

**Proof.** We need to show that C1 and C2 are satisfied under hypotheses i)-iv). Consider extension $e$ from partial routes $r_{v_i}^{(1,1)}$ and $r_{v_i}^{(1,2)}$ to customer $v_j$, resulting in partial routes denoted by $r_{v_i}^{(1,1)} \otimes e$ and $r_{v_i}^{(1,2)} \otimes e$, respectively.
respectively. Let \(e_0, e_1, \ldots, e_t\) be the nodes in extension \(e\), where \(e_0 = v_i\) and \(e_t = v_j\). Assume that extended path \(r_{v_i}^{(t,1)} \otimes e\) is robust feasible, i.e., for each \(e_h\), where \(h = 1, \ldots, t\), we have \(V_{e_{(h-1),e_h}} = 0\), \(Q_{e_{h},\gamma}^{(2,\mathcal{F})} \leq Q\) for \(\gamma = 0, \ldots, \Gamma^q\) and \(T_{e_{h},\gamma}^{(2,\mathcal{F})} \leq w_{e_h}^b\) for \(\gamma = 0, \ldots, \Gamma^{(t,s)}\). By hypotheses ii) and iii) for \(\gamma = 0\), hypothesis iv) for \(\gamma = 0\) and the fact that the extension functions for components \(V_{v_i,1}, \ldots, V_{v_i,n}\), \(Q_{v_i,0}\) and \(T_{v_i,0}\) are nondecreasing, it follows that extended path \(r_{v_i}^{(t,1)} \otimes e\) is also deterministic and feasible, and thus, \(C1\) is partially met. On the other hand, according to this supposition, we have

\[
\mathcal{Q} \geq Q_{e_{1},\gamma}^{(2,\mathcal{F})} \text{ for } \gamma = 1, \ldots, \Gamma^q \\
= \max\{Q_{e_{1},\gamma}^{(2,\mathcal{F})} + \bar{q}_{e_{1}}, Q_{v_i,\gamma-1}^{(2,\mathcal{F})} + \bar{q}_{e_{1}} + \bar{q}_{e_1}\} \text{ for } \gamma = 1, \ldots, \Gamma^q \implies (\text{by expression } (4.3)) \\
\geq \max\{Q_{v_i,\gamma}^{(1,\mathcal{F})} + \bar{q}_{e_{1}}, Q_{v_i,\gamma}^{(1,\mathcal{F})} + \bar{q}_{e_{1}} + \bar{q}_{e_1}\} \text{ for } \gamma = 1, \ldots, \Gamma^q \implies (\text{by condition } iii)) \\
= Q_{v_i,\gamma}^{(1,\mathcal{F})} \text{ for } \gamma = 1, \ldots, \Gamma^q \implies (\text{by expression } (4.3))
\]

The previous reasoning can be applied sequentially to every node \(e_h\) \((h = 2, \ldots, t)\) of extension \(e\), resulting in the conclusion that \(Q_{e_{h},\gamma}^{(1,\mathcal{F})} \leq Q\) for all \(\gamma = 1, \ldots, \Gamma^q\). Applying a similar reasoning for component \(T_{e_{h},\gamma}^{(1,\mathcal{F})}\) for \(\gamma = 0, \ldots, \Gamma^{(t,s)}\), we obtain that \(T_{e_{h},\gamma}^{(1,\mathcal{F})} \leq w_{e_h}^b\) for all \(\gamma = 0, \ldots, \Gamma^{(t,s)}\) and \(h = 1, \ldots, t\). Thus, all robust feasible extensions of \(r_{v_i}^{(t,2)}\) are also robust feasible for \(r_{v_i}^{(t,1)}\), and therefore \(C1\) is fully satisfied. Analyzing the reduced costs for customer \(e_1\) of \(e\), we obtain

\[
\tilde{C}_{e_1}^{(1,\mathcal{F})} = \tilde{C}_{e_1}^{(1,\mathcal{F})} + p_3 d_{v_i,e_1} - \bar{p}_{e_1} \text{ (by the definition of the extension function for the reduced cost)} \\
\leq \tilde{C}_{v_i}^{(2,\mathcal{F})} + p_3 d_{v_i,e_1} - \bar{p}_{e_1} \text{ (by hypothesis } i)) \\
= \tilde{C}_{e_1}^{(2,\mathcal{F})} \text{ (by the definition of the extension function for the reduced cost)}
\]

The above procedure can be performed for all \(h = 2, \ldots, t\), leading to \(\tilde{C}_{e_h}^1 \leq \tilde{C}_{e_h}^2\). Thus, \(C2\) is satisfied, and the proof is completed. 

To speed up the generation of negative reduced cost columns, we use a bidirectional labeling algorithm introduced by [Righini and Salani (2008)]. This algorithm gradually extends forward and backward paths until a halfway point is reached. Forward and backward paths are subsequently combined and checked for feasibility and negativity of the reduced cost. The bidirectional labeling algorithm is described in more detail in Appendix A.

5.3. Valid inequalities, branching decisions and a primal heuristic

The subset-row (SR) inequalities are commonly used in BPC algorithms for VRP variants in an effort to tighten the lower bounds of the column generation procedure [Jepsen et al. (2008)]. In this paper, the SR inequalities introduced in [Jepsen et al. (2008)] are modified as in [Munari and Morabito (2018)] to include various operating modes of routes. Given set \(\delta = \{i_1, i_2, i_3\} \subset \mathcal{N}\), the SR inequalities can be defined as follows:

\[
\sum_{l=1}^{L} \sum_{r \in I_\delta} \lambda^r_l \leq 1, \quad (5.7)
\]

where \(I_\delta\) corresponds to the set of columns/routes in the RMP with at least two nodes in \(\delta\). The separation of these inequalities is performed by a simple enumeration procedure that examines all columns of the RMP for each set \(\delta\) of three nodes.
If branching is required in the BPC search tree, we apply three types of branching decisions. First, if the number of vehicles is fractional, we branch on that number and create two child nodes in the BPC search tree by imposing a constraint in the RMP. Second, if the number of vehicles is an integer, branching is performed on the total number of deliverymen, provided that number is fractional. Once again, two child nodes are created in the BPC search tree by adding a constraint in the RMP. Third, we finally proceed to branching in terms of arc-flow variables $\sum_{\ell=1}^{L} x_{ij}^{\ell}$ if the numbers of vehicles and deliverymen are both integers. In this case, two child nodes are created by imposing exactly one of the two constraints, $\sum_{\ell=1}^{L} x_{ij}^{\ell} = 0$ and $\sum_{\ell=1}^{L} x_{ij}^{\ell} = 1$, in the RMP associated with each node. The latter branching rule modifies all SPs as well as the RMPs of the child nodes as follows: in one of the child nodes, arcs $(i, j)$ are removed in all SPs, whereas in the other node, all arcs leaving node $i$ are removed except for the arc that directly connects with $j$. In the RMP associated with $\sum_{\ell=1}^{L} x_{ij}^{\ell} = 0$, we remove all columns associated with routes that visit node $i$ and go directly to node $j$. Finally, we remove all columns associated with routes that visit node $i$ and go directly to a node different from node $j$ in the RMP associated with $\sum_{\ell=1}^{L} x_{ij}^{\ell} = 1$.

The RMP associated with each node of BPC is run until the relative gap falls below a threshold, and thus, an approximate solution $\tilde{\lambda}$ is obtained. For this reason, before proceeding to define the branching decisions, the following two cases must be considered. First, if the objective value corresponding to solution $\tilde{\lambda}$ is smaller than the current upper bound and if there is at least one branching candidate, branching is performed as explained before. Second, if the objective value of solution $\tilde{\lambda}$ is greater than or equal to the current upper bound, we solve the RMP of the node until an optimal solution is obtained, and the branching process is restarted for the optimal solution of the RMP.

We also use a primal heuristic based on MIP, requiring that master variables be integers and solving the resulting problem by a general-purpose MIP solver. The purpose is to obtain feasible solutions of the problem to improve the upper bound, which may then lead to a better performance of the BPC method. Further details on SR inequalities, branching decisions, and primal heuristic are available in Álvarez and Munari (2017); Munari and Morabito (2018).

6. Computational results

In this section, we present the results of computational experiments performed with the following objectives: 1) investigate the potential of the RO approach to address the cost-risk trade-off, and 2) analyze the algorithms proposed for solving the RVRPTWMD in terms of solution quality and computational efficiency. The BPC algorithm was implemented in C++ using the PDCGM library (Gondzio et al., 2013, 2016) that offered a stabilized CG method based on well-centered dual solutions obtained by the primal-dual interior point method. The BPC search tree was managed using the interior point branch-and-price framework described in Munari and Gondzio (2013). In addition, it used the IBM CPLEX Optimization Studio v.12.8 to solve mixed-integer programming problems. The robust flow formulation (4.5)-(4.14) and the RH1 were also implemented in C++ using the Concert library of CPLEX. The procedures for separating the 1- and 2-path valid inequalities were also implemented in C++ and resort to the Push-Relabel algorithm available in the Concorde library, and the bidirectional dynamic programming labeling algorithm described in Subsection 5.2 to solve the RTSPTW. All experiments were run on a Linux PC with 16 GB of RAM and an Intel Core i7 4790 CPU operating at 3.6 GHz.

We present numerical results for two sets of problem instances adapted from the well-known instances proposed by Solomon (1987). In set A we only consider instances with 50 customers and keep the original
vehicle capacity, while in set B we reset the vehicle capacity $Q$ to 100 and 200 for instances with 25 and 50 customers, respectively. In this second set we followed a common practice used in the literature to modify the parameter values of Solomon’s instances in order to better visualize the cost-risk trade-off (Lee et al. 2012; De La Vega et al. 2019). The geographic distribution of customers in the Solomon’s instances varies according to the instance class as follows: 1) a grouped distribution (classes C1 and C2); 2) a random distribution (classes R1 and R2); and 3) a combination of grouped and random distribution (classes RC1 and RC2). We used the original demands and travel times as the nominal values of these parameters. Nominal service times were generated as in Pureza et al. (2012). Preliminary computational results showed evidence of the difficulty of solving these instances with 100 customers even using the BPC algorithm. For this reason, we focus on instances with 25 and 50 customers. Furthermore, preliminary computational experiments revealed the difficulty of the BC algorithm to solve instances with 50 customers. Therefore, we report the results of computational experiments with this method using the instances with 25 customers.

In instance set A (original vehicle capacity), we set the uncertainty levels $\%\delta^d$ and $\%\delta(t,s)$ at 50. The values used for uncertainty budgets and $(\Gamma^q, \Gamma(t,s))$ were $(0,0)$, $(1,1)$, $(2,2)$ and $(5,5)$. A total of 244 instances can be found in this first set of instances, obtained by combining the 56 instances of Solomon with each of the four values attributed to the uncertainty budgets. We also set $m = |K| = 20$ and $E = 40$.

Table 1 summarizes the characteristics of 2016 instances of set B (modified vehicle capacity). This count results from combining 18 configurations of robust parameters with 56 modified Solomon instances (column No.Inst.) for each value of $n = 25, 50$. Values of parameters $m$ and $E$ are also shown in this table.

<table>
<thead>
<tr>
<th>Inst. configuration</th>
<th>Robust Parameters</th>
<th>No.Inst.</th>
<th>$n = 25$</th>
<th>$n = 50$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\Gamma^q$</td>
<td>$\delta^d$</td>
<td>$\Gamma(t,s)$</td>
<td>$\delta(t,s)$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0.0</td>
<td>1</td>
<td>0.3</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0.0</td>
<td>3</td>
<td>0.3</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0.0</td>
<td>5</td>
<td>0.3</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0.0</td>
<td>8</td>
<td>0.3</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0.0</td>
<td>10</td>
<td>0.3</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0.3</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>0.3</td>
<td>1</td>
<td>0.3</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>0.3</td>
<td>3</td>
<td>0.3</td>
</tr>
<tr>
<td>16</td>
<td>5</td>
<td>0.3</td>
<td>5</td>
<td>0.3</td>
</tr>
<tr>
<td>17</td>
<td>8</td>
<td>0.3</td>
<td>8</td>
<td>0.3</td>
</tr>
<tr>
<td>18</td>
<td>10</td>
<td>0.3</td>
<td>10</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Table 1: Parameter values used in the computational experiments with instances in set B.

The RHI1 method described in Subsection 4.3 was initialized according to the criteria of the farthest customer, and the values of the RHI1 parameters were $\phi_1 = 0.6$, $\phi_2 = 0.4$, $\mu = 1$ and $\lambda = 1$. The meanings of these parameters are described in Solomon (1987) and De La Vega et al. (2019). For the uncertain parameters (demand and travel and service times), the corresponding deviations were determined as percentages of their nominal values, i.e., $\hat{q}_i = \text{trunc}(\%\delta^d \cdot \bar{q}_i/100)$, $\hat{t}_{ij} = \text{trunc}(\%\delta(t,s) \cdot \bar{t}_{ij}/100)$ and $\hat{s}_i = \text{trunc}(\%\delta(t,s) \cdot \bar{s}_i/100)$, in which parameters $\%\delta^d$ and $\%\delta(t,s)$ correspond to the uncertainty levels of demand and service and travel times, respectively. To avoid rounding errors, we decided to truncate parameter values $\hat{q}_i$, $\hat{t}_{ij}$ and $\hat{s}_i$. 

18
6.1. Results for instance set A

In this subsection, we analyze the computational performance of the robust heuristic II (RHI1) and the branch-price-and-cut (BPC) solution approaches when solving instances of set A. Table 2 reports, for each class and each value of $\Gamma_q$ and $\Gamma^{(t,s)}$, the average values of the upper bound (UB) of the objective value, number of vehicles ($nV$), total number of deliverymen ($nD$), total traveled distance (Dist.) and optimality gap ($%GAP$) for the RHI1 and BPC solution approaches. Additionally, this table reports the number of times ($N_{RHI1}$) that the RHI1 solution approach found a solution with an objective value lower than the objective value of the BPC, the average time in seconds (Time) that the BPC took to find the solutions, and the number of solutions solved to optimality (Nopt). All results in Table 2 were obtained with the uncertainty levels $%\delta_q = %\delta^{(t,s)} = 50$.

<table>
<thead>
<tr>
<th>Class</th>
<th>$\Gamma_q$</th>
<th>$\Gamma^{(t,s)}$</th>
<th>UB</th>
<th>nV</th>
<th>nD</th>
<th>Dist.</th>
<th>%GAP</th>
<th>N_{RHI1}</th>
<th>UB</th>
<th>nV</th>
<th>nD</th>
<th>Dist.</th>
<th>%GAP</th>
<th>Time (s)</th>
<th>Nopt</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>0</td>
<td>0</td>
<td>5.5688</td>
<td>5</td>
<td>5</td>
<td>688.2</td>
<td>0.59</td>
<td>0</td>
<td>5.5362</td>
<td>5</td>
<td>5</td>
<td>362.3</td>
<td>0.00</td>
<td>560</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>5.5942</td>
<td>5</td>
<td>5</td>
<td>719.4</td>
<td>3.06</td>
<td>0</td>
<td>5.5437</td>
<td>5</td>
<td>5</td>
<td>436.5</td>
<td>2.12</td>
<td>1725</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>6.6998</td>
<td>6</td>
<td>6</td>
<td>775.7</td>
<td>20.65</td>
<td>0</td>
<td>5.6828</td>
<td>5</td>
<td>5</td>
<td>605.5</td>
<td>2.32</td>
<td>2422</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>6.9808</td>
<td>6</td>
<td>7</td>
<td>807.9</td>
<td>6.43</td>
<td>1</td>
<td>6.7061</td>
<td>6</td>
<td>7</td>
<td>505.7</td>
<td>2.28</td>
<td>1499</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>10.6606</td>
<td>8</td>
<td>22</td>
<td>939.1</td>
<td>21.39</td>
<td>0</td>
<td>8.8580</td>
<td>7</td>
<td>18</td>
<td>830.2</td>
<td>0.45</td>
<td>795</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>13.5129</td>
<td>11</td>
<td>26</td>
<td>1045.7</td>
<td>25.81</td>
<td>0</td>
<td>10.6139</td>
<td>9</td>
<td>19</td>
<td>889.3</td>
<td>0.00</td>
<td>288</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>14.7510</td>
<td>12</td>
<td>28</td>
<td>1092.9</td>
<td>25.06</td>
<td>0</td>
<td>11.8180</td>
<td>10</td>
<td>22</td>
<td>930.2</td>
<td>0.25</td>
<td>734</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>16.4294</td>
<td>13</td>
<td>32</td>
<td>1127.2</td>
<td>24.89</td>
<td>0</td>
<td>13.0894</td>
<td>10</td>
<td>26</td>
<td>977.7</td>
<td>0.09</td>
<td>493</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>10.8221</td>
<td>9</td>
<td>22</td>
<td>971.2</td>
<td>14.85</td>
<td>0</td>
<td>9.4736</td>
<td>8</td>
<td>19</td>
<td>860.8</td>
<td>0.25</td>
<td>1030</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>13.3989</td>
<td>11</td>
<td>28</td>
<td>1114.0</td>
<td>17.25</td>
<td>0</td>
<td>11.407</td>
<td>9</td>
<td>23</td>
<td>981.8</td>
<td>0.00</td>
<td>291</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>15.6492</td>
<td>12</td>
<td>32</td>
<td>1242.3</td>
<td>13.82</td>
<td>0</td>
<td>13.8237</td>
<td>11</td>
<td>26</td>
<td>1111.9</td>
<td>0.00</td>
<td>237</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>17.5950</td>
<td>14</td>
<td>36</td>
<td>1325.3</td>
<td>11.48</td>
<td>0</td>
<td>16.0829</td>
<td>13</td>
<td>31</td>
<td>1204.1</td>
<td>1.98</td>
<td>951</td>
<td>6</td>
</tr>
<tr>
<td>RC1</td>
<td>0</td>
<td>0</td>
<td>2.2721</td>
<td>2</td>
<td>2</td>
<td>720.7</td>
<td>38.78</td>
<td>0</td>
<td>2.2445</td>
<td>2</td>
<td>2</td>
<td>445.1</td>
<td>37.28</td>
<td>3261</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>2.2748</td>
<td>2</td>
<td>2</td>
<td>747.9</td>
<td>34.44</td>
<td>0</td>
<td>2.2463</td>
<td>2</td>
<td>2</td>
<td>462.5</td>
<td>32.78</td>
<td>3117</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>2.2750</td>
<td>2</td>
<td>2</td>
<td>750.3</td>
<td>31.03</td>
<td>0</td>
<td>2.2451</td>
<td>2</td>
<td>2</td>
<td>451.3</td>
<td>29.39</td>
<td>3617</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>2.2662</td>
<td>2</td>
<td>2</td>
<td>662.2</td>
<td>23.93</td>
<td>1</td>
<td>2.2443</td>
<td>2</td>
<td>2</td>
<td>443.0</td>
<td>22.79</td>
<td>3614</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>2.6946</td>
<td>2</td>
<td>5</td>
<td>946.2</td>
<td>18.55</td>
<td>0</td>
<td>2.5040</td>
<td>2</td>
<td>4</td>
<td>767.7</td>
<td>10.85</td>
<td>3612</td>
<td>0</td>
</tr>
<tr>
<td>R2</td>
<td>1</td>
<td>1</td>
<td>2.9315</td>
<td>2</td>
<td>5</td>
<td>951.4</td>
<td>21.57</td>
<td>2</td>
<td>2.6148</td>
<td>2</td>
<td>4</td>
<td>784.8</td>
<td>8.92</td>
<td>3608</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>3.2311</td>
<td>3</td>
<td>6</td>
<td>947.6</td>
<td>31.48</td>
<td>1</td>
<td>2.7228</td>
<td>2</td>
<td>5</td>
<td>773.8</td>
<td>10.83</td>
<td>3484</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>3.3456</td>
<td>3</td>
<td>6</td>
<td>910.3</td>
<td>33.54</td>
<td>2</td>
<td>3.2853</td>
<td>2</td>
<td>5</td>
<td>811.1</td>
<td>13.64</td>
<td>3396</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>3.8225</td>
<td>3</td>
<td>7</td>
<td>1100.4</td>
<td>40.10</td>
<td>1</td>
<td>3.1257</td>
<td>3</td>
<td>5</td>
<td>881.8</td>
<td>14.44</td>
<td>2380</td>
<td>3</td>
</tr>
<tr>
<td>RC2</td>
<td>1</td>
<td>1</td>
<td>3.8274</td>
<td>3</td>
<td>7</td>
<td>1024.4</td>
<td>33.02</td>
<td>0</td>
<td>2.9369</td>
<td>2</td>
<td>5</td>
<td>869.2</td>
<td>0.53</td>
<td>2609</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>4.1154</td>
<td>3</td>
<td>8</td>
<td>1029.1</td>
<td>32.06</td>
<td>1</td>
<td>3.2111</td>
<td>3</td>
<td>6</td>
<td>861.4</td>
<td>2.05</td>
<td>2293</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>4.4031</td>
<td>4</td>
<td>8</td>
<td>1031.4</td>
<td>29.13</td>
<td>0</td>
<td>3.6558</td>
<td>3</td>
<td>6</td>
<td>808.1</td>
<td>4.26</td>
<td>2371</td>
<td>6</td>
</tr>
</tbody>
</table>

| Average | 7.2968 | 6 | 13 | 944.6 | 23.04 | 9 |
| Total   | 6.3018 | 5 | 11 | 752.3 | 8.33 | 2016 |
| 127     |

Table 2: Average results obtained by the RHI1 and BPC solution approaches in each class of instance set A (original vehicle capacity) with 50 customers and uncertainty levels $%\delta_q = %\delta^{(t,s)} = 50$.

In the deterministic problem, RHI1 finds solutions with upper bounds close to those provided by the BPC method in classes C1, C2, RC1 and RC2. These solutions have the same number of routes, while overestimates either the total number of deliverymen or the total traveled distance. A different behavior is observed in the solutions for instances in classes R1 and R2 in which, in addition to overestimating the number of deliverymen and total traveled distance, they also overestimate the number of routes. The advantage of RHI1 is that robust feasible solutions are found on average time in less than 1 second. A noteworthy observation is that the heuristic found solutions with objective value lower than the objective value provided by the BPC approach in nine instances (due to the time limit imposed for the BPC).
Regarding the BPC method, it solved to optimality 127 of the 244 instances (around 52%) of this first set of instances. Among the instances solved optimally, 94 of them are in classes C1, R1 and RC1. It confirms the results reported in the literature regarding the difficulty of BPC-based algorithms to solve instances with a long planning horizon, requiring the design of routes with many customers. Indeed, this feature often increases the computational time of the dynamic programming labeling algorithm because more extensions are needed to optimally solve the RSPPRC. This behavior is even worse in the RVRPTWMD, as the subproblem is solved \( L \) times (one for each mode \( \ell = 1, \ldots, L \)) in each iteration of the column generation technique. The difficulty in solving the instances in classes C2, R2 and RC2 is confirmed by the high values of the optimality gaps in comparison to the values obtained for classes C1, R1 and RC1.

6.2. Results for instance set B with 25 customers

The following results were obtained after performing numerical experiments based on the instances with 25 customers.

6.2.1. Impact of parameters \( \Gamma^q \) and \( \Gamma^{(t,s)} \)

The purpose of this subsection is to evaluate the impact of parameters \( \Gamma^q \) and \( \Gamma^{(t,s)} \) on the routing decisions, cost and risk based on results obtained by solving the RVRPTWMD using three different approaches: RHI1; the branch-and-cut (BC) algorithm taking as a starting point the solution provided by the RHI1 (BC with RHI1); and the BPC algorithm. The BC with RHI1 algorithm is used to solve the robust flow formulation (4.5)-(4.14). A time limit of 3600 seconds was established for all approaches.

Tables 3, 4 and 5 present, for each configuration of robust parameters and for each approach, the average results in terms of the upper bound (UB) of the objective function, the number of routes generated (nV), the total number of deliverymen (nD), the total distance traveled (Dist), the computational time in seconds (time), the proportion of samples that proved to be infeasible in the Monte-Carlo simulation (%Prob), the price of robustness (%PR), the solution gap (%GAP) and the number of instances solved to optimality (Nopt) for the instances of classes C1-R1, RC1-C2 and R2-RC2, respectively. We omit the Nopt column from results for the RHI1 because this method cannot guarantee optimality. On the other hand, the %RG column was added to the results of the BC with RHI1 to show that BC with RHI1’s additional relative gain with respect to solutions obtained by the RHI1 method.

There are several data that need to be clarified in these tables. The values in columns %PR referring to the price of robustness were determined as follows: \( \%PR = (z^{rob}/z^{det} - 1) \times 100 \), where \( z^{rob} \) corresponds to the objective function value taking robustness into account, and \( z^{det} \) represents the value of the objective function of the nominal or deterministic problem. Thus, %PR describes the relative increase in the value of the objective function due to the preference for a risk-averse solution over the risk-neutral solution. The average gap for the algorithms RHI1 and BC with RHI1 was determined as \( (UB/UB^{BPC} - 1) \times 100 \) based on the upper bound \( UB^{BPC} \) given by the BPC algorithm. The values of column %RG were calculated as \( (z^{RHI1}/z^{RTIFF} - 1) \times 100 \), where \( z^{RHI1} \) corresponds to the objective function value of the solution given by the RHI1, and \( z^{RTIFF} \) is the objective function value of the solution found by the BC with RHI1 after 3600 seconds of computational time.

The values in column %Prob, which represent the solution risk, were determined using a Monte Carlo simulation as follows: 10000 random and independent samples were generated, each containing arrays of demand and travel and service times. Component value \( (i,j) \) of the travel time matrix was generated uniformly in the interval \( [\bar{t}_{ij} - \hat{t}_{ij}, \bar{t}_{ij} + \hat{t}_{ij}] \), which represents the extremes of the range in which random
<table>
<thead>
<tr>
<th>Config.</th>
<th>UB</th>
<th>nV</th>
<th>nD</th>
<th>Dist</th>
<th>%Prob</th>
<th>%PR</th>
<th>%GAP</th>
<th>UB</th>
<th>nV</th>
<th>nD</th>
<th>Dist</th>
<th>%Prob</th>
<th>%PR</th>
<th>%GAP</th>
<th>UB</th>
<th>nV</th>
<th>nD</th>
<th>Dist</th>
<th>Time(s)</th>
<th>%Prob</th>
<th>%PR</th>
<th>Nopt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6.5510</td>
<td>5</td>
<td>12</td>
<td>593.5</td>
<td>79.2</td>
<td>0.1</td>
<td>15.4</td>
<td>6.1171</td>
<td>5</td>
<td>10</td>
<td>504.5</td>
<td>3004.9</td>
<td>32.3</td>
<td>1.3</td>
<td>8.9</td>
<td>5.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>6.6344</td>
<td>5</td>
<td>10</td>
<td>494.0</td>
<td>26.7</td>
<td>63.3</td>
<td>0.3</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>7.1196</td>
<td>6</td>
<td>12</td>
<td>612.3</td>
<td>2.2</td>
<td>8.8</td>
<td>23.4</td>
<td>6.5026</td>
<td>6</td>
<td>9</td>
<td>526.1</td>
<td>3005.0</td>
<td>1.6</td>
<td>7.6</td>
<td>14.7</td>
<td>8.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7.1708</td>
<td>6</td>
<td>12</td>
<td>624.9</td>
<td>0.0</td>
<td>9.6</td>
<td>13.6</td>
<td>6.5945</td>
<td>6</td>
<td>9</td>
<td>527.9</td>
<td>3010.9</td>
<td>0.0</td>
<td>9.2</td>
<td>4.9</td>
<td>8.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6.5415</td>
<td>5</td>
<td>12</td>
<td>581.4</td>
<td>100.0</td>
<td>0.0</td>
<td>15.6</td>
<td>6.0406</td>
<td>5</td>
<td>10</td>
<td>489.4</td>
<td>2791.6</td>
<td>65.5</td>
<td>0.0</td>
<td>7.8</td>
<td>7.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6.9237</td>
<td>6</td>
<td>13</td>
<td>569.9</td>
<td>78.6</td>
<td>5.8</td>
<td>13.4</td>
<td>6.6040</td>
<td>6</td>
<td>10</td>
<td>539.9</td>
<td>2515.9</td>
<td>93.0</td>
<td>9.3</td>
<td>9.8</td>
<td>4.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8.2107</td>
<td>7</td>
<td>16</td>
<td>607.4</td>
<td>0.0</td>
<td>25.5</td>
<td>19.1</td>
<td>7.7561</td>
<td>7</td>
<td>12</td>
<td>560.6</td>
<td>3003.4</td>
<td>0.0</td>
<td>28.4</td>
<td>14.9</td>
<td>5.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8.2777</td>
<td>7</td>
<td>16</td>
<td>610.1</td>
<td>0.0</td>
<td>26.5</td>
<td>19.8</td>
<td>7.6900</td>
<td>6</td>
<td>12</td>
<td>566.8</td>
<td>3005.3</td>
<td>0.0</td>
<td>27.3</td>
<td>13.5</td>
<td>8.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>7.7338</td>
<td>6</td>
<td>14</td>
<td>587.8</td>
<td>13.1</td>
<td>18.2</td>
<td>14.4</td>
<td>7.1470</td>
<td>6</td>
<td>12</td>
<td>553.4</td>
<td>3002.9</td>
<td>22.0</td>
<td>18.3</td>
<td>7.0</td>
<td>8.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>7.9440</td>
<td>6</td>
<td>15</td>
<td>606.9</td>
<td>0.2</td>
<td>21.4</td>
<td>13.1</td>
<td>7.4724</td>
<td>6</td>
<td>12</td>
<td>557.2</td>
<td>3004.5</td>
<td>0.3</td>
<td>23.7</td>
<td>7.4</td>
<td>6.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>8.2370</td>
<td>7</td>
<td>15</td>
<td>620.5</td>
<td>0.0</td>
<td>25.9</td>
<td>16.5</td>
<td>7.8814</td>
<td>7</td>
<td>12</td>
<td>563.6</td>
<td>3010.5</td>
<td>0.0</td>
<td>30.5</td>
<td>13.1</td>
<td>4.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>7.3617</td>
<td>6</td>
<td>13</td>
<td>598.4</td>
<td>31.6</td>
<td>12.5</td>
<td>15.6</td>
<td>6.8691</td>
<td>6</td>
<td>11</td>
<td>534.0</td>
<td>2930.6</td>
<td>25.7</td>
<td>13.8</td>
<td>7.8</td>
<td>6.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Results for classes C1 and R1 of instance set B with 25 customers.
<p>| Config. | UB nV nD Dist %Prob %PR %GAP UB nV nD Dist Time(s) %Prob %PR %GAP %RG Nopt UB nV nD Dist Time(s) %Prob %PR Nopt |
|--------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| 1      | 7.2716 6 5 501.3 400.0 100.0 0.0 3.0 7.0700 6 10 574.7 3152.0 100.0 0.0 0.2 2.9 4 7.0572 6 10 571.8 115.6 100.0 0.0 8   |
| 2      | 9.6224 8 15 848.7 1.3 32.3 6.5 9.0973 8 10 723.4 3601.6 0.5 28.7 0.5 5.8 1 9.0489 8 10 739.2 971.7 0.5 28.2 7   |
| 3      | 7.7699 6 16 574.5 0.0 6.9 5.5 7.3914 6 13 539.5 3151.0 0.1 4.5 0.3 5.1 5 7.3670 6 13 545.4 11.0 0.1 4.4 8   |
| 4      | 10.1545 8 17 794.5 4.9 39.6 6.6 9.5357 8 15 732.0 3161.4 1.4 34.9 0.1 6.5 0 9.5235 8 15 734.9 243.2 5.8 34.9 8   |
| 5      | 10.7774 9 17 773.5 25.1 48.2 11.4 10.1377 9 16 752.3 3315.5 0.0 43.4 4.7 6.5 1 9.6871 8 16 745.5 631.9 0.0 37.3 7   |
| Average| 9.0054 7.700 3.8 248.8 8.2 366.4 3.3 8.3646 7.12 646.8 4.3 6.53 3.0 6.5188 5.9 5.9 409.3 108.7 22.8 17.7 8   |</p>
<table>
<thead>
<tr>
<th>Column</th>
<th>UB w/o DHT</th>
<th>UB w/ DHT</th>
<th>UB w/ DHT+ Tarcriptions</th>
<th>UB w/ DHT+ Tarcriptions+SPH-NPR</th>
<th>UB w/ DHT+ Tarcriptions+SPH-NPR+NP</th>
<th>UB w/ DHT+ Tarcriptions+SPH-NPR+NP+PP</th>
<th>BC w/ DHT+ Tarcriptions+SPH-NPR+NP+PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>2</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>3</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>4</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>5</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>6</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>7</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>8</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>9</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>10</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>11</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>12</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>13</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>14</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>15</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>16</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>17</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
<tr>
<td>18</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
<td>4.4667</td>
</tr>
</tbody>
</table>

Table 5: Results for classes C2 and RC2 of instance set B with 25 customers.
variable $t_{ij}$ may vary. Similarly, component values of the service time and demand vectors were generated uniformly between their respective minimum and maximum values of variation, i.e., $\hat{s}_i \in [\hat{s}_i - \hat{\delta}_i, \hat{s}_i + \hat{\delta}_i]$ and $\hat{q}_i \in [\hat{q}_i - \hat{\gamma}_i, \hat{q}_i + \hat{\gamma}_i]$ for all $i \in N$. The solution risk was determined by testing how many of 10000 generated samples became infeasible. Infeasibility of a sample can be easily verified by examining the time windows of customers and the depot as well as the vehicle capacity of routes using the travel time matrix and the service time and demand vectors of the current sample.

Tables 3, 4 and 5 are designed to evaluate three cases: 1) Demand is the only uncertain parameter, in which case $\Gamma^{(t,s)} = 0$ and $\Gamma^q$ is varied. 2) Travel and service times are considered uncertain, but demand is represented by its nominal value; hence, in this case, $\Gamma^q = 0$ and $\Gamma^{(t,s)}$ varies. 3) The three parameters are considered uncertain; thus, both parameters $\Gamma^q$ and $\Gamma^{(t,s)}$ are varied. Regardless of the class and the solution approach used to solve the RVRPTWMD, the results in the tables show the following:

- The solutions of the risk-neutral or nominal problem ($\Gamma^q = \Gamma^{(t,s)} = 0$) tend to fail with high probability because their level of risk ($\%\text{Prob}$) reaches its maximum value with respect to solutions with positive values of $\Gamma^q$ and $\Gamma^{(t,s)}$, and this value is high relative to the others. This indicates that when this solution is implemented, any small variation in the uncertain parameters of the problem could make it infeasible.

- Solution risk ($\%\text{Prob}$) is significantly reduced by the robust approach, i.e. if $\Gamma^q > 0$ and $\Gamma^{(t,s)} > 0$, but leads to an increase in the value of the objective function, as can be observed from the values of $\%\text{PR}$. This is why the price of robustness ($\%\text{PR}$) can also be regarded as the lowest relative additional cost that can be charged to a decision maker for providing a solution protected against uncertainties. These results highlight the potential of the static RO approach to address the cost-risk trade-off. This is also corroborated by the trade-off curves shown in Figure 1 for various values of uncertainty budgets $\Gamma^q$ and $\Gamma^{(t,s)}$ at $\Gamma^q = \Gamma^{(t,s)} = 0$ (configuration 1) and $\Gamma^q = \Gamma^{(t,s)} = 1, 3, 5, 8, 10$ (configurations 14-18). The plotted curves represent the results of the BPC algorithm, but the analysis is also valid for the results of the RHI1 and BC with RHI1 algorithms. Considering these curves, it is important to emphasize the two scenarios that result in their extreme values; if $\Gamma^q = \Gamma^{(t,s)} = 0$, all parameters assume their nominal/expected values, causing solutions to be entirely unprotected against uncertainties. Note that at this point, the risk in each of the curves reaches its maximum value. At the other extreme, i.e., if $\Gamma^q = \Gamma^{(t,s)} = 10$, all parameters reach their worst-case values, matching the method of Soyster with solutions that are fully protected against uncertainties. Note that at this point the risk is 0, but the price of robustness, on the other hand, reaches its maximum value. Thus, the curves of Figure 1 can also be called Pareto curves since they provide a set of optimal solutions in which the choice of a solution depends on how risk-averse the decision maker is. For example, if the decision maker is risk-averse, then he/she should select solutions located to the right of each curve, i.e., solutions in which parameters $\Gamma^q$ and $\Gamma^{(t,s)}$ assume their highest values. Conversely, solutions located to the left of each curve can be chosen by a decision maker who is neutral or indifferent to risk because in such solutions parameters $\Gamma^q$ and $\Gamma^{(t,s)}$ have lower values.

- In all classes except R1, $\%\text{PR}$ reaches higher values if parameter $\Gamma^q$ is varied, i.e., if case 1 is considered (demand corresponds to the only parameter considered uncertain). Note that the number of vehicles is the factor that contributes the most to $\%\text{PR}$, as, in general, this number increases in comparison to the number of vehicles in the nominal solution. On the other hand, if $\Gamma^q$ is fixed at 0 and $\Gamma^{(t,s)}$ is varied, in general, the number of vehicles used in the deterministic solution remains stable. However,
Figure 1: Cost-risk trade-off according to various values of uncertainty budgets $\Gamma^q$ and $\Gamma^{(t,s)}$ for $\Gamma^q = \Gamma^{(t,s)} = 0$ (configuration 1) and $\Gamma^q = \Gamma^{(t,s)} > 0$ (configurations 14-18) for all classes of instance set B with 25 customers solved by the BPC method.
the number of deliverymen or the total distance traveled increases. As these two indicators do not represent an excessive cost in the objective function, the increase of %PR in these cases is not enough to be considered significant. As to the instances in class R1, they are known to have a short planning horizon as well as fairly narrow time windows. Therefore, there is a very high chance that a perturbation of parameter $\Gamma(t,s)$ will lead to customers not being served within the requisite time windows following the solution of the nominal problem. For this reason, the robust solution increases either the number of routes or the number of deliverymen to serve all customers within the requisite time windows, leading to a significant increase in %PR since these two alternatives are the most expensive. Therefore, for this class, it can be concluded that %PR values of problems reach higher values if parameter $\Gamma(t,s)$ varies.

6.2.2. Analysis of the robust II heuristic

Regardless of the values assigned to parameters $\Gamma_q$ and $\Gamma(t,s)$, the RHI1 always determines a robust feasible solution for the RVRPTWMD within a very short computational time of less than 1 second on average. The heuristic gap was determined with respect to the LB given by the BPC algorithm. Note that the RHI1 in general does not perform well if customers are randomly scattered. This is demonstrated by the relatively large average gaps in instances of classes R1, RC1, R2 and RC2. However, for the instances of classes C1 and C2, the RHI1 can obtain solutions with the same number of routes as in the optimal solution yet an increased number of deliverymen or the total distance covered. Since these are the cheapest terms in the objective function of the problem, the change in the gap is not enough to be considered significant. As already mentioned, in the instances of classes C1 and C2 the customers along the same route are typically those in the same cluster, and therefore the RHI1 simply defines the sequence of these customers, considering only the cheapest alternatives in the objective function, the number of deliverymen and the total distance traveled.

6.2.3. Analysis of the BC with RHI1 algorithm

The results in column %RG of the tables show that the BC with RHI1 was able to significantly improve the solutions provided by the robust heuristic II in the instances of classes R1 and RC1. On average, the BC with RHI1 improved RHI solutions by 7.1%. This gain was attained because the combined solution approach, in addition to reducing the solutions’ total distance traveled, also reduces the total number of used vehicles and deliverymen. Due to the lexicographic classification of costs in the objective function of the model, any reduction in the number of routes and deliverymen can significantly reduce the total cost. For instances of classes C1, C2, R2 and RC2, the BC with RHI1 attained slightly better results on average, as it was unable to significantly improve the RHI1 solutions. The average gain was less than 2%, which can be regarded as negligible since the BC with RHI1 was run for 3600 seconds, while the RHI1 found its solutions in less than 1 second on average. For most instances, the RHI1 heuristic approach can determine the optimal number of routes. The defined capacity for instances in classes C1, C2, R2 and RC2 lead to routes that cover few customers per route. In addition, the instances have a long planning horizon, which indicates that waiting contributes significantly to the total duration of the route. In such cases, to improve a solution provided by the RHI1, it is sufficient that the BC find solutions that reallocate customers between routes without needing to use more than one deliveryman to comply with the time windows. For this reason, the gains in the instances of these classes are insignificant since only the total distance traveled is being reduced. Given the lexicographic classification of the objective, any improvement in distance does not have a significant impact on the value of the objective function.
In general, the BC with RHI1 based on the RTIFF performed well, although it took an average time of over 2331 seconds to find the solutions. The average gaps of solutions of instances of classes C1, C2, R2 and RC2 are insignificant, less than 0.01%. For the other instances, the average gap was 5%. Considering all classes, the BC with RHI1 solved 60% of the instances. For the remaining 40%, an average gap of less than 3% was obtained. These results show that the BC with RHI1 based on the compact formulation (4.5)–(4.14) is an attractive alternative for generating high-quality solutions for small instances of the RVRPTWMD.

To verify the performance of the proposed BC method regarding the impact of generating 1- and 2-path inequalities, we ran additional computational experiments with the general-purpose BC method of CPLEX. We also initialize CPLEX with the solution provided by the RHI and thus we refer to the resulting approach as CPLEX with RHI1. Hence, the only difference to our BC approach (BC with RHI1) is the insertion of 1- and 2-path cuts. Table 6 reports the average results obtained by both BC approaches, grouped by instance class. The headings UB, nV, nD and Dist. have the same meaning as in Tables 3–5. The headings Time, %GAP and Nopt mean, respectively, the average computational time (in seconds) considering only instances solved to optimality by both methods, the optimality gap (in percentage) and the number of instances solved to optimality. For the BC with RHI1, we also report the number of 1- and 2-path cuts added. As presented in the table, the BC with RHI1 solved 575 instances to optimality, 16 more than the other approach. Moreover, it required less average computational time to solve instances with optimality proven by both approaches.

We also note that the BC with RHI1 provided, on average, smaller optimality gaps than CPLEX with RHI1. This is because the quality of the solutions provided by the former is never worse than that provided by the latter. Additionally, the BC with RHI1 generated far more 2-path cuts than 1-path ones. This result was expected since in our algorithm we only check 1-path inequalities if there are no violated 2-path inequalities. In short, these results revealed that the addition of k-path inequalities brings benefit to the BC approach, as the BC with RHI1 presented a superior overall performance with respect to CPLEX with RHI1.

<table>
<thead>
<tr>
<th>Class</th>
<th>CPLEX with RHI1</th>
<th>BC with RHI1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UB</td>
<td>nV</td>
</tr>
<tr>
<td>C1</td>
<td>6.5118</td>
<td>6</td>
</tr>
<tr>
<td>R1</td>
<td>6.9094</td>
<td>6</td>
</tr>
<tr>
<td>RC1</td>
<td>8.4283</td>
<td>7</td>
</tr>
<tr>
<td>C2</td>
<td>6.5190</td>
<td>6</td>
</tr>
<tr>
<td>R2</td>
<td>4.8099</td>
<td>4</td>
</tr>
<tr>
<td>RC2</td>
<td>7.7619</td>
<td>7</td>
</tr>
<tr>
<td>Average</td>
<td>6.8234</td>
<td>6</td>
</tr>
<tr>
<td>Total</td>
<td>559</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Average results obtained by CPLEX with RHI1 and BC with RHI1 for instance set B with 25 customers

Further computational experiments were performed to also compare the performance of CPLEX with RHI1 with that obtained by a similar solution approach based on the robust assignment formulation (RAF) introduced by [De La Vega et al. (2019)]. These numerical experiments were performed on the instances proposed in the cited study. It should be mentioned that in [De La Vega et al. (2019)], only customer demand was considered uncertain. Therefore, the need to consider parameters \( \Gamma^{(t,s)} \) and \( \delta^{(t,s)} \) in the RTIFF was avoided by simply fixing \( \Gamma^{(t,s)} = 0 \) and \( \delta^{(t,s)} = 0.0\% \). Relative to the RAF, the CPLEX with RHI1 approach went from providing average gaps of 15.9% and 19.6% to 4.24% and 3.89% for class R1 at \( \%\delta^q = 20 \) and \( \%\delta^q = 30 \), respectively. Moreover, the average gaps in class C1 went from 12.50% to 0.006% at \( \%\delta^q = 20 \) and from 13.20% to 0.007% at \( \%\delta^q = 30 \). These results show that the CPLEX with RHI1 approach performs significantly better if it relies on the RTIFF instead of the RAF.
6.2.4. Analysis of the branch-price-and-cut algorithm

The results for the BPC algorithm in Tables 3, 4 and 5 illustrate that the instances of class RC2 are the most challenging for this algorithm, particularly for $\Gamma^q > 0$ and $\Gamma^{(t,s)} > 0$. On average, the algorithm took approximately 3120 seconds to solve them. In fact, there are some instances in RC2 (62 instances out of 144) for which the BPC algorithm did not prove optimality. Some customers in instances of this class are scattered and others are clustered, leading to many routes with similar costs. Moreover, customers’ time windows are fairly open in this class, which also leads to many feasible routes. This causes the use of more labels in the robust labeling algorithm, and therefore many more comparisons in the robust dominance rule are made. On the other hand, the algorithm optimally solves all instances of class R1 (216 instances) within an average computational time of only 23 seconds, which demonstrates that such instances are the easiest for the algorithm. In contrast to instances of classes with clustered customers, the customers of instances in R1 are randomly distributed, leading to a few routes with a similar cost; as customers’ time windows are quite narrow, there are few feasible routes. Thus, fewer labels are used, and fewer comparisons in the robust dominance rule are made.

Finally, the BPC algorithm optimally solved 930 out of 1008 problems with 25 customers. Compared to 575 problems solved optimally by the BC with RHI1, it can be concluded that the BPC algorithm was the algorithm that attained the best performance in terms of the quality of the solution as well as computational efficiency. On average, the BPC algorithm was 5 times faster than the BC with RHI1.

To summarize the main results of Tables 3, 4 and 5, we have plotted two bar charts in Figure 2. The bar chart on the left presents the average of the objective function (OF) values, while the one on the right shows the total number of instances solved to optimality (Nopt), in each class, by each solution approach. The charts clearly illustrate that RHI1 provides solutions as good as those provided by the other two solution approaches, for instances in classes C1, C2, R2 and RC2. For instances in class R1, the BPC was the solution approach that obtained the best results, while for class RC1 both BPC and BC with RHI1 provided good quality solutions. Finally, the BPC was the algorithm with the largest number of instances solved to optimality, particularly for classes R1 and RC1.

6.3. Results for instance set B with 50 customers

This section presents the numerical results related to instances with 50 customers in set B. It is important to note that these experiments were only performed for the RHI1 and BPC algorithms. Here, we omit the discussion of the impact of parameters $\Gamma^q$ and $\Gamma^{(t,s)}$ on the routing decisions, cost and risk since the results behave similarly to those reported in Subsection 6.2.1 for instances with 25 customers. For this reason, it was decided to simply discuss the quality of solutions obtained by each algorithm as well as the computational
time required by algorithms to determine solutions. Figure 3 shows two curves – one for RHI1 and one for the BPC algorithm – related to the values of the objective function of 1008 problems. These curves show that BPC remains the algorithm that determines solutions with the smallest costs; however, it requires, as expected, a much longer time (2237 seconds) to determine them, on average. The RHI1 requires less than 1 second on average. These curves also show that although the RHI1 is dominated by the BPC algorithm, it provides solutions as good as those obtained by the BPC algorithm for classes C1, C2, R2 and RC2. In contrast, for problems of classes R1 and RC1 the BPC algorithm clearly dominates the RHI1. A total of 771 problems out of 1008 were solved optimally by the BPC algorithm, confirming its strong computational performance in solving problems exactly.

![Objective value for instances with 50 customers in set B obtained using the RHI1 and BPC solution approaches.](image)

Finally, we also assessed the impact of the vehicle capacity on the computational performance of the proposed BPC algorithm by comparing the results obtained by solving instances of set A against the results of instances of set B. For both sets, we considered the instances with 50 customers and in class 2. Additionally, we used $\Gamma^q, \Gamma^{(t,s)} = 0, 1, 2, 5$ and $\%\delta^q = \%\delta^{(t,s)} = 50$. From the 108 instances in each set, the method was able to solve to optimality 33 instances from set A (within 2345.2 seconds, on average) and 57 instances from set B (within 2113.2 seconds, on average). These results show a better computational performance of the BPC algorithm in the modified instances rather than in the original ones. This is explained by the fact that the modified instances result in routes with fewer customers, on average, because of the reduction in the vehicle capacity.

### 7. Conclusions

In this paper, we addressed the RVRPTWMD with uncertain demand and travel and service times. Uncertainties were taken into account using a static RO approach. We proposed a robust vehicle flow formulation based on the linearization of recursive equations, which resulted in a model with fewer variables and constraints than what would be the case if the dualization scheme was used. To solve this model, we proposed a BC with RHI1 that consisted of extending the I1 heuristic of Solomon to the RVRPTWMD.
to generate robust feasible solutions to initialize a general-purpose optimization solver. Additionally, we modeled the RVRPTWMD using a set partitioning formulation and developed a BPC algorithm. We showed how to extend the labeling algorithm to solve the robust resource-constrained shortest path problem that arose as the subproblem in the BPC algorithm.

In light of the results obtained by the proposed approaches, we observed that the variation of parameters $\Gamma^q$ and $\Gamma^{(t,s)}$ had an impact on routing and scheduling decisions, cost and risk. These results corroborate those obtained in previous studies and, therefore, show that the static RO approach is a powerful tool for considering uncertainties and is quite useful in dealing with the trade-off between cost and risk. Despite the time needed to perform computations being long on average, good-quality solutions were obtained by the BC with RHI1 based on the robust flow formulation, with average gaps of 2%. Moreover, we observed that the BC with RHI1 obtained better results when using the robust flow formulation proposed in this paper than when using the robust assignment formulation proposed by [De La Vega et al., 2019].

The BPC algorithm proved optimality for more than 84% of the instances with 25 customers of the first set and needed less time to perform computations than did the BC with RHI1. We conclude that the BPC algorithm results in an effective strategy for solving the RVRPTWMD exactly in modified instances with 25 and 50 customers. On the other hand, the computational results also showed the difficulty of the BPC algorithm in optimally solving the original Solomon instances with 50 customers, especially in the instances with a long planning horizon and hence routes that visit many customers. In practical environments where decisions must be made quickly, one option is the RHI1 heuristic that determines robust solutions to the problem while needing very little time to perform computations. An interesting direction of future research would be to use the uncertainty set introduced in Poss (2013, 2014), known as the variable budgeted uncertainty set, to reduce the level of conservatism of solutions. Another direction of future research is to use other approaches such as two-stage stochastic programming with recourse as well as develop exact methods based on Benders decomposition or (other decomposition technique) to deal with uncertainties.

It is worth mentioning that the extension of the proposed BPC method to consider a heterogeneous fleet requires including a vehicle type index in variables of the set partitioning model and defining one subproblem for each type of vehicle, in addition to each mode. Another interesting extension that, however, would involve major changes in the approaches proposed here would be to consider pickup and delivery services simultaneously, which is applicable in real-world situations, such as the distribution of returnable beverage packaging.

8. Acknowledgments

The authors thank the anonymous reviewers whose constructive comments helped to improve the original version of this paper. This work was supported by the São Paulo Research Foundation (FAPESP) under grants 15/14582-7 and 16/01860-1; the Brazilian National Council for Scientific and Technological Development (CNPq), under grant 04601/2017-9; and the Coordination for the Improvement of Higher Education Personnel (CAPES). This support is gratefully acknowledged.

References


Gendreau, M., Jabali, O., Rei, W., 2016. 50th anniversary invited article - future research directions in stochastic vehicle routing. Transportation Science 50, 1163–1173.


Poss, M., 2013. Robust combinatorial optimization with variable budgeted uncertainty. 4OR 11, 75–92.
Appendix A. Robust bidirectional labeling algorithm

The robust labeling algorithm proposed here resorts to the the bidirectional extension technique proposed by [Righini and Salani (2008)]. It is based on two sets of labels for each customer $v_i$. The first set contains the robust feasible paths that start from the initial depot 0 and end at customer $v_i$. The other set of labels stores the robust feasible paths that depart from $v_i$ and ends at the final depot $n + 1$. Paths in the first set are called forward partial paths while those in the second set are called backward partial paths. In this appendix, we show how carry out the backward extensions in the context of proposed robust approach.

Label $\mathcal{B}_{v_i}^\ell$ is used to represent the backward partial path $r_{v_i}^{\ell} = (v_j, v_{j+1}, \ldots, v_{k+1})$, which operates in mode $\ell$, starts at customer $v_j$ and ends at the final depot $v_{k+1} = n + 1$. The states, label selection, extension functions and dominance rules of the backward partial paths are symmetrical to those presented for the forward partial paths (see Section 5.2). Therefore, vector $(\hat{c}_{v_i}^{\ell}, c_{v_{j+1}}, \ldots, c_{v_{k+1}}, f_{v_{j+1}}, \ldots, f_{v_{k+1}}, T_{v_{j+1}}, \ldots, T_{v_{k+1}}, Q_{v_j, \gamma}, \ldots, Q_{v_{k-1}})$ is used to represent the label $\mathcal{B}_{v_i}^\ell$. The time resource components, denoted by $T_{v_{j+1}, \gamma}$, with $0 = 0, \ldots, \Gamma(t,s)$, are now defined as:

- $T_{v_{j+1}, \gamma}$: latest time possible to begin the service at customer $v_j$ given that the service and travel times of up to $\gamma$ arcs or nodes on the backward partial path $(v_j, v_{j+1}, \ldots, v_{k+1})$, with $v_{k+1} = n + 1$, are allowed to vary around its nominal value.

Equation (A.1) corresponds to the extension function to determine the values of the components $T_{v_{j+1}, \gamma}^{\ell}$ for $\gamma = 0, \ldots, \Gamma(t,s)$, as a result of extending label $\mathcal{B}_{v_i}^\ell$ associated to a given partial backward path to customer $v_i$. Note that the extension function takes into consideration the four inspected cases in Section
4.2 to determine the recursive equations (4.4). When extending label $B_v^\ell$ to customer $v_i$, the resulting $B_v^\ell$ is accepted as a robust feasible extension if $V_{v_i,v_i}^B = 0$; $Q_{v_i,\gamma}^F \leq Q$ for $\gamma = 0, \ldots, \Gamma^q$ and $T_{v_i,\gamma}^B \geq w_{v_i}^a$ for $\gamma = 0, \ldots, \Gamma^{(t,s)}$.

\[
T_{v_i,\gamma}^B = \begin{cases} 
\min\{w_{v_i}^b, T_{v_j,\gamma}^B - s_{v_i}^f - \bar{t}_{v_i,v_j}\}, & \text{if } \gamma = 0, \\
\min\{w_{v_i}^b, T_{v_j,\gamma}^B - s_{v_i}^f - \bar{t}_{v_i,v_j}, T_{v_j,(\gamma-1)}^B - s_{v_j}^f - \bar{t}_{v_j,v_i}, T_{v_j,(\gamma-2)}^B - s_{v_j}^f - \bar{t}_{v_j,v_i} - \bar{t}_{v_i,v_j}\}, & \text{if } \gamma = 1, \\
\min\{w_{v_i}^b, T_{v_j,\gamma}^B - s_{v_i}^f - \bar{t}_{v_i,v_j}, T_{v_j,(\gamma-1)}^B - s_{v_j}^f - \bar{t}_{v_j,v_i}, T_{v_j,(\gamma-2)}^B - s_{v_j}^f - \bar{t}_{v_j,v_i} - \bar{t}_{v_i,v_j}, T_{v_j,(\gamma-3)}^B - s_{v_j}^f - \bar{t}_{v_j,v_i} - \bar{t}_{v_i,v_j} - \bar{t}_{v_i,v_j}\}, & \text{otherwise,}
\end{cases}
\]

(A.1)

The bidirectional labeling algorithm is generally more effective than its unidirectional version to solve the SPPRC ([Righini and Salani, 2008, Baldacci el al., 2010]). An important question in the bidirectional labeling algorithm lies in the joining of forward and backward labels. A route starting at the initial depot 0 and ending at the final depot $n+1$ is obtained when a forward label, $F_v^\ell$, and another backward one, $B_v^\ell$, are feasible together at node $v_i$. Due to the consideration of robustness, the joining step between the labels $F_v^\ell \in B_v^\ell$ needs to be modified appropriately. In this context, the robust feasible conditions to join the forward label with the backward label are

1) $Q_{v_i,\gamma}^F + Q_{v_i+1,(\Gamma^q-\gamma)}^B \leq Q$, for $\gamma = 0, \ldots, \Gamma^q$;

2) $T_{v_i,\gamma}^F \leq T_{v_i,(\Gamma^{(t,s)}-\gamma)}^B$, for $\gamma = 0, \ldots, \Gamma^{(t,s)}$;

3) $V_{v_i,j}^F + V_{v_i,j}^B \leq 1$, for $j = 1, \ldots, n$ and $j \neq v_i$.

Condition 1) takes into account the loads on the forward and backward paths considering that the demand of up to $\gamma$ and $\Gamma^q - \gamma$ customers reach their worst-case value, respectively. Condition 2) expresses the joining of the forward and backward labels that leads to a consistent route in time. Finally, condition 3) ensures that the joining generates only elementary routes.