CIRRELT

CIRRELT-2022-34

Compact Formulations for the Robust Vehicle Routing Problem with Time Windows under Demand and Travel Time Uncertainty

Rafael Campos
Pedro Munari
Leandro C. Coelho

November 2022

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2022-010.

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

Compact Formulations for the Robust Vehicle Routing Problem with Time Windows under Demand and Travel Time Uncertainty

Rafael Campos[1], Pedro Munari[1], Leandro C. Coelho[2,*]

1. Production Engineering Department, Federal University of São Carlos, São Carlos, SP, 13561-353, Brazil
2. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Canada Research Chair in Integrated Logistics

Abstract. We provide new compact formulations for the robust vehicle routing problem with time windows (RVRPTW) under cardinality- and knapsack-constrained demand and travel time uncertainty. Particularly, we propose the first compact model that addresses the RVRPTW under travel time uncertainty considering the knapsack uncertainty set. Our models use different types of constraints to control time propagation based on the well-known Miller-Tucker-Zemlin and single commodity flow constraints. The latter has not been explored even for the deterministic variant of the problem, so we first state them explicitly. We also design tailored branch-and-cut algorithms based on the proposed formulations, which rely on a dynamic programming algorithm to verify if a solution is robust feasible with respect to demand and time, and use specific as well as standard separation methods found in the literature. We present detailed computational results on RVRPTW instances, compare the performance of our models and algorithms, and evaluate the impact and advantages of implementing each studied uncertainty set.

Keywords: vehicle routing problem with time windows, uncertainty, robust optimization, commodity flow, knapsack uncertainty set.

Acknowledgements. This work was supported by São Paulo Research Foundation (FAPESP) [grant numbers 19/22235-6, 19/23596-2, 16/01860-1], Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) [Finance Code 001] and the National Council for Scientific and Technological Development (CNPq) [grant number 313220/2020-4], and the Natural Sciences and Engineering Council of Canada (NSERC)) [grant number 2019-00094].

---

* Corresponding author: leandro.coelho@fsa.ulaval.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2022

1 Introduction

In distribution problems, it is often assumed that all parameters are known and deterministic when solving a vehicle routing problem (VRP) [Gendreau et al., 2016]. However, in reality, significant uncertainty is associated with these problems, such as on the travel time or on each customer's demand. If their probability distributions are known, one can resort to stochastic optimization [Birge and Louveaux, 1997]. However, when these are unknown, or one wants to ensure a feasible solution is guaranteed for all scenarios, robust optimization (RO) is often more appropriate [Ordonez, 2010].

One perspective of RO seeks to protect the solution against the worst-case variation of the input data [Ben-Tal and Nemirovski, 1999]. Yet, "worst-case" is problem-dependent. One popular way to perceive worst-case is the realization of uncertain parameters that deteriorates the objective function the most. This approach then focuses on providing optimized solutions that are feasible for any possible realization of the input data. While approaches such as the Distributionally RO may require knowledge on some statistical data on the uncertain parameters, such as the mean and variance [Rahimian and Mehrotra, 2022], most standard approaches do not need this information. In fact, several works in the RO literature require only an estimate of the worst-case values of the uncertain parameters [Agra et al., 2012, Gounaris et al., 2016, Munari et al., 2019]. To avoid overconservative solutions, with excessive costs due to over-protection against uncertainty, different *uncertainty sets* can model the uncertain parameters realization whose worst-case behavior can be controlled by the decision-maker. Different uncertainty sets often result in solutions with different costs and robustness levels [Subramanyam et al., 2020]. From a decision-maker standpoint, these different characteristics are attractive since they can choose the one that best suits their strategy. Some examples of uncertainty sets used in the literature are the *cardinality-constrained* uncertainty set [Bertsimas and Sim, 2004], in which the decision-maker limits the number of parameters simultaneously attaining their worst-case value, and the *knapsack* uncertainty set [Minoux, 2009], in which the total deviation of a set of parameters is limited instead.

In the VRP literature, the use of different uncertainty sets has been little explored in the context of travel time variability, as they appear on the robust VRP with time windows (RVRPTW). While there are studies on different uncertainty sets for VRP under variability in demand [Gounaris et al., 2013, Subramanyam et al., 2020], all RVRPTW work under travel time variability

have considered only the cardinality-constrained uncertainty set [Agra et al., 2012, Lee et al., 2012, Munari et al., 2019]. Recently, Bartolini et al. [2021] proposed a column generation algorithm that considers uncertainty on travel times with a knapsack uncertainty set for the robust traveling salesman problem with time windows, a variant of the RVRPTW with a single vehicle. Yet, there are still no compact formulations for the RVRPTW under uncertainty on travel times.

This paper provides the first compact formulations for the RVRPTW under demand and travel time uncertainty following the knapsack-constrained sets. Our main contributions are to:

- propose a new compact commodity flow (CF) formulation for the deterministic VRPTW. We are not aware of any compact formulation for this variant in which both load and time propagation are modeled based on CF constraints;
- introduce two compact formulations for the RVRPTW under demand and travel time uncertainty following the cardinality-constrained and the single knapsack uncertainty sets. We also extend these formulations to account for the multiple knapsacks case;
- design tailored branch-and-cut (BC) algorithms based on the proposed formulations, in which robustness is verified for each integer solution using a polynomial-time dynamic programming (DP) algorithm.

Our compact formulation based on CF constraints presents stronger linear programming (LP) relaxations than those of Munari et al. [2019] and better performance for the deterministic case. We perform detailed computational analyses on RVRPTW instances, showing that, while harder to solve, instances considering the knapsack uncertainty set may provide solutions with similar level of robustness as some configurations of the cardinality-constrained set while costing less, highlighting some advantages of its use. The BC algorithm also showed positive results, solving almost twice as many instances to optimality as a commercial solver applied on the compact models.

The remainder of this paper is organized as follows. In Section 2, we introduce some definitions and concepts used in this work. In Section 3, we introduce a novel compact formulation based on CF constraints for the deterministic VRPTW and extend it to consider uncertainty on the demand and travel times following a recent modeling strategy in the RO literature [Munari et al., 2019]. In addition, we also describe a new modeling strategy that considers uncertainty on travel times with the single and multiple

knapsack-constrained uncertainty sets and derive two compact formulations using them. In Section 4, we describe the proposed BC algorithms. The computation results of the experiments are shown in Section 5. Finally, the concluding remarks are discussed in Section 6.

2 Background and Literature Review

In this section, we start with some background definitions and in Section 2.1 we describe the two uncertainty sets used in this paper. Section 2.2 reviews the existing compact formulations developed for the RVRPTW. Note that they only consider the traditional cardinality-constrained uncertainty set, as there are no works in literature with a compact formulation using knapsack uncertainty to model travel time variability.

The notation used is as follows. Let $C = \{1, \ldots, n\}$ be the set of n customers; $N = C \cup \{0, n+1\}$ be the set of nodes including two copies of the depot; and $A = \{(i,j) \mid i,j \in N, i < n+1, j > 0, i \neq j\}$ be the set of arcs. For each arc $(i,j) \in A$, we define c_{ij} as the travel cost from i to j and assume they satisfy the triangle inequality. The set of homogeneous vehicles is K, each with capacity Q. Each customer $i \in C$ has a demand $d_i \leq Q$.

The incorporation of time windows also requires parameters $[a_i, b_i]$, which indicate the earliest and latest times a vehicle may start serving customer i, respectively; s_i is the service time of node i; and t_{ij} is the travel time of arc $(i,j) \in A$. To account for these new characteristics, in compact models one often resorts to the use of constraints inspired by the Miller-Tucker-Zemlin (MTZ) formulation [Miller et al., 1960].

2.1 Uncertainty Sets

In this section, we review the cardinality-constrained and the knapsack uncertainty sets.

2.1.1 Cardinality-Constrained Set.

The cardinality-constrained set works with a budget Γ representing the threshold of the total scaled variation of the uncertain parameters. Particularly, if Γ is integer, we can interpret it as the number of worst-case realizations that can simultaneously occur in a route [Bertsimas and Sim, 2004]. As in most

RVRPs, we assume that the worst-case value of the demand and travel time parameters occur when they assume their highest possible value since this impairs the feasibility of the route the most. Then, a solution is considered robust feasible for a budget Γ if it is feasible when up to any Γ uncertain parameters simultaneously assume their worst-case value. Under demand uncertainty, that means the solution does not violate the vehicle capacity when the demands of any Γ customers assume their largest values. Likewise, if travel time is the uncertain parameter, the solution is robust feasible if all time windows are satisfied for any combination of Γ arcs attaining their worst-case travel times.

In the cardinality-constrained set, travel time uncertainty is modeled with random variables γ_{ij}^t, $0 \leq \gamma_{ij}^t \leq 1$, which indicate the normalized scale deviation for the travel time of arc (i, j). The sum over these random variables is limited by the travel time budget Γ^t. Thus, the uncertainty set designed for travel times, U^t, is represented by:

$$U^t = \{t \in \mathbb{R}^{|A|} \mid t_{ij} = \bar{t}_{ij} + \hat{t}_{ij}\gamma_{ij}^t, \forall (i,j) \in A; \sum_{(i,j)\in A} \gamma_{ij}^t \leq \Gamma^t; 0 \leq \gamma_{ij}^t \leq 1, \forall (i,j) \in A\},$$

where the travel time t_{ij} for each arc $(i, j) \in A$ ranges from its nominal value $\bar{t}_{ij}$ up to $\bar{t}_{ij} + \hat{t}_{ij}$, with $\hat{t}_{ij}$ being its maximum deviation. A similar idea applies for demand uncertainty, using a demand budget Γ^d and random variables γ_i^d, which leads to the uncertainty set:

$$U^d = \{d \in \mathbb{R}^{|N|} \mid d_i = \bar{d}_i + \hat{d}_i\gamma_i^d, i \in N; \sum_{i\in N} \gamma_i^d \leq \Gamma^d; 0 \leq \gamma_i^d \leq 1, i \in N\}.$$

2.1.2 Knapsack Set.

Unlike the cardinality-constrained set, the knapsack uncertainty set does not limit the number of worst-case realizations. Instead, it limits the total absolute deviation on a route, considering one or more knapsacks [Subramanyam et al., 2020]. Each knapsack l involves a subset of nodes for demands or arcs for travel times, with its own budget of uncertainty Δ_l. Usually, the RVRP literature models these knapsacks by relating the realizations to geographical regions [Gounaris et al., 2013, Subramanyam et al., 2020]. For example, one may group the nodes/arcs into four different quadrants (NE, SE, NW, SW) and set a budget for each quadrant based on its characteristics. For instance, a quadrant with higher variability in demand may have larger budget than

others. Additionally, one usually builds the multiple knapsacks as disjoint sets, meaning that their nodes/arcs do not overlap.

Let L be the set of knapsacks and S_l the set of arcs in each knapsack $l \in L$. We define Δ_l^t as the budget of uncertainty for travel times in knapsack $l \in L$. Then, we can model this set for travel time, U_L^t, using the following expression:

$$U_L^t = \{t \in \mathbb{R}^{|A|} \mid \bar{t}_{ij} \leq t_{ij} \leq \bar{t}_{ij} + \hat{t}_{ij}, \forall (i,j) \in A; \sum_{(i,j) \in S_l} (t_{ij} - \bar{t}_{ij}) \leq \Delta_l^t, l \in L\}.$$

In this definition, the travel time realization t_{ij} ranges from $\bar{t}_{ij}$ to $\bar{t}_{ij} + \hat{t}_{ij}$, and the sum of all deviations in knapsack l is limited by Δ_l^t. A similar expression can be generated for problems under demand uncertainty, using a budget Δ_l^d, and defining S_l as the set of nodes in each knapsack $l \in L$:

$$U_L^d = \{d \in \mathbb{R}^{|N|} \mid \bar{d}_i \leq d_i \leq \bar{d}_i + \hat{d}_i, \forall i \in N; \sum_{i \in S_l} (d_i - \bar{d}_i) \leq \Delta_l^d, l \in L\}.$$

In practical settings, this type of representation might be more appropriate than the cardinality-constrained set, especially regarding time uncertainty. It is often easier for a driver to estimate how late he or she is when traveling to a specific region than to tell how many streets usually achieve their worst-case traffic.

A variant of this uncertainty set we study in this work is the *single-knapsack* uncertainty set, in which a single knapsack encompasses every arc/node for each uncertain parameter (i.e., $|L| = 1$). Thus, budget Δ^t limits the total deviation in travel times in a route. This set is most appropriate in systems where there is no clear distinction of behavior among customers, or if the decision-maker wants to introduce a limitation for each vehicle that is independent of the route.

To the best of our knowledge, there is no paper considering the knapsack uncertainty set for uncertain travel times in the RVRPTW literature. The works that use this uncertainty set only consider uncertainty on the demand [Gounaris et al., 2013, Pessoa et al., 2021, Subramanyam et al., 2020]. It is worth noting that it is not trivial to extend these methods to consider uncertainty on travel times because they were designed based on the assumption that if a knapsack is completely filled, it does not matter where in the route the load deviation occurred. However, we cannot make this assumption in problems with uncertainty on travel times, as travel time deviations can be

absorbed by the customer's opening time windows and waiting time. Indeed, even after a long deviation on a given arc, the vehicle may still have to wait for the time window to open. Therefore, we cannot make the same assumption as in the case of uncertain demand, requiring new modeling strategies and algorithms.

2.2 Robust Optimization Formulations for VRPs

The dualization scheme introduced by Bertsimas and Sim [2003] is the most traditional approach in the RO literature to derive robust counterparts of continuous or discrete linear optimization problems when relying on cardinality-constrained sets. This strategy is based on replacing the protection function of each constraint with its corresponding dual problem, ensuring the robust counterpart remains linear. More specifically, each constraint with an uncertain parameter has a protection function that consists of a continuous linear optimization problem that determines the worst-case realization for the uncertain parameters in that constraint. This subproblem is then replaced by its dual problem, which introduces new (continuous) variables and constraints to the robust counterpart. We refer to Bertsimas and Sim [2004] for further details.

In the RVRP literature, Agra et al. [2012] present a compact model derived from the dualization approach based on the so-called layered formulation of the VRPTW. The authors considered uncertainty on travel times only, inspired by a maritime transportation problem with no capacity constraints. The layered formulation is based on creating a flow problem where a graph is defined for every vehicle, and each one of them has n layers. Each l-th layer represents the node where the vehicle is after visiting $l-1$ nodes on a path from the origin. For each layer, there is a set of possible arcs that the vehicle can traverse, based on the feasibility of the time windows constraints and the nodes previously visited on the route. The authors reported computational results for the obtained robust counterpart considering small-scale instances with 10 to 20 cargoes and 1 to 5 vehicles. These instances were all solved to optimality when using an additional strategy that reduces the maximum number of layers, with average computational times of 225.29 seconds to solve the model, plus 107.64 seconds to run the layer-reduction algorithm. Munari et al. [2019] tested the same formulation on instances from Solomon's benchmark set with 25 customers to compare it with their own formulation (that is not based on the dualization approach). They ob-

served that the layered-based formulation could not prove optimality for any instance and found feasible solutions or proved infeasibility for only 59.48% of them within the time limit of 3600 seconds.

Lee et al. [2012] also used the dualization scheme to model the RVRP with deadlines, a variant of the RVRPTW which does not consider the opening time windows. As reported by the authors, preliminary computational experiments showed that the solver could not process the compact formulation even for small-scale instances, so the authors resorted to Dantzig-Wolfe decomposition. Another limitation of their formulation is that it is not easily extendable to the RVRPTW, as it considers that a worst-case scenario is the one that has the maximum delay within the chosen budget, which may not be true when the opening time windows are available, as the waiting times may absorb part of the deviation.

Recently, Munari et al. [2019] proposed an alternative for the dualization scheme when using cardinality-constrained sets in the context of RVRP variants. This approach is based on the linearization of recursive equations that model the worst-case realizations of the uncertain parameters. This linearization results in constraints that guarantee a robust feasible solution in a compact model. Their work is based on the cardinality-constrained uncertainty set and assumes that the budget of uncertainty is integer and thus defined as the maximum number of worst-case values taken simultaneously by the uncertain parameters. For instance, considering the budget Γ^t for travel time uncertainty and a given route $r = (v_0, v_1, \ldots, v_h)$, the variable $w_{v_j\gamma}$ represents the worst case for service start time at node v_j in route r, when the travel times at any $\gamma \leq \Gamma^t$ arcs up to node v_j attain their worst-case values simultaneously. The worst case for service start time at node v_j is the latest between its opening time window (a_{v_j}) and the worst-case arrival time. Let $\bar{t}_{v_{j-1}v_j}$ be the nominal travel time on arc (v_{j-1}, v_j) and $\hat{t}_{v_{j-1}v_j}$ be its maximum deviation. Then, the value of $w_{v_j\gamma}$ can be computed using the following recursive equation [Munari et al., 2019]:

$$w_{v_j\gamma} = \begin{cases} a_{v_0}, & \text{if } j = 0, \\ \max\{a_{v_j}, w_{v_{j-1}\gamma} + \bar{t}_{v_{j-1}v_j} + s_{v_{j-1}}\}, & \text{if } \gamma = 0, \\ \max\{a_{v_j}, w_{v_{j-1}\gamma} + \bar{t}_{v_{j-1}v_j} + s_{v_{j-1}}, & \\ \qquad w_{v_{j-1}\gamma-1} + \bar{t}_{v_{j-1}v_j} + \hat{t}_{v_{j-1}v_j} + s_{v_{j-1}}\}, & \text{otherwise,} \end{cases} \tag{1}$$

for all $j = 0, \ldots, h$ and $0 \leq \gamma \leq \Gamma^t$. We can convert these equations into the following linear constraints for a two-index compact formulation, which

guarantee a robust feasible solution with respect to time propagation and time window satisfaction:

$$w_{j\gamma} \geq w_{i\gamma} + (s_i + \bar{t}_{ij})x_{ij} - M_{ij}(1 - x_{ij}), \qquad (i,j) \in A, \gamma = 0, \ldots, \Gamma^t, \quad (2)$$

$$w_{j\gamma} \geq w_{i(\gamma-1)} + (s_i + \bar{t}_{ij} + \hat{t}_{ij})x_{ij} - M_{ij}(1 - x_{ij}), \qquad (i,j) \in A, \gamma = 1, \ldots, \Gamma^t, \quad (3)$$

$$a_i \leq w_{i\gamma} \leq b_i, \qquad i \in N, \gamma = 0, \ldots, \Gamma^t, \quad (4)$$

where x_{ij} is the commonly used binary variable that assumes the value of 1 if, and only if, a vehicle traverses arc $(i,j) \in A$, and M_{ij} is a sufficiently large number that can be set as $\max\{0, b_i - a_j\}$, for each $(i,j) \in A$. For each $\gamma = 0, \ldots, \Gamma^t$, constraints (2) and (3) guarantee that, if $x_{ij} = 1$ for a given $(i,j) \in A$, the service starting time at node v_j when γ travel times attain their worst-case values is computed by choosing the largest between two possibilities: γ worst-case realizations already happened in arcs that precede arc (i,j), as represented by constrains (2); or $\gamma - 1$ worst-case realizations happened before (i,j) and, thus, the travel time on arc (i,j) attains its worst-case value, as represented by constraints (3). Constraints (4) impose the satisfaction of the time windows. We can apply the same strategy to derive similar constraints for demand uncertainty. Because this robust counterpart yields fewer constraints and variables than those derived using the dualization approach [Munari et al., 2019], it performs significantly better on general-purpose linear optimization solvers. Yu et al. [2022] also report superior results when using this approach to model a robust variant of the team orienteering problem. For this reason, besides the difficulties of extending dualization methods to problems with travel time variability, the models introduced in this work are based on the linearization approach.

3 New Compact Formulations for the RVRPTW

We present novel RO models for the RVRPTW which resort to the linearization technique of recursive equations proposed by Munari et al. [2019]. First, we develop in Section 3.1 a new CF formulation for the deterministic VRPTW. Then, we derive its robust counterpart using the cardinality-constrained uncertainty set to obtain a new compact RO model for the RVRPTW in Section 3.2. Recall that an MTZ-based formulation already exists for this problem [Munari et al., 2019], but no CF-based formulation

has been proposed thus far. Then, in Section 3.3, we propose the first CF- and an MTZ-based models for the single and multiple knapsack uncertainty sets.

3.1 CF Formulation for the VRPTW

While there are formulations based on CF constraints for several VRP variants in literature [Gouveia, 1995, Letchford and Salazar-González, 2006, 2015], we did not find any compact formulation explicitly defined for the VRPTW in which both load and time propagation are modeled using this type of constraints. There is, however, one for the TSP with time windows [Langevin et al., 1993] and another for the split delivery VRPTW [Bianchessi and Irnich, 2019], both in a deterministic context, but none of them models the two types of propagation using CF constraints. In what follows, we present our developments to introduce time windows and time flow constraints.

Let x_{ij} be the binary variable that assumes the value of 1 if, and only if, a vehicle traverses arc $(i, j) \in A$, which is commonly used to define two-index vehicle flow formulations. We define a continuous variable f_{ij} to represent the load of the vehicle that traverses arc $(i, j) \in A$, inspired by the variables introduced by Gavish [1984]. To adapt this model to the VRPTW, we treat time as a second commodity. Thus, we introduce continuous variables g_{ij} that represent the elapsed time of a route when the vehicle enters arc (i, j) after serving node i. The model is then given by:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij}, \tag{5}$$

$$\text{s.t.} \sum_{i:(i,j)\in A} x_{ij} = 1, \qquad j \in C, \tag{6}$$

$$\sum_{h:(h,i)\in A} x_{hi} = \sum_{j:(i,j)\in A} x_{ij}, \qquad i \in C, \tag{7}$$

$$\sum_{j:(i,j)\in A} f_{ij} = d_i + \sum_{h:(h,i)\in A} f_{hi}, \qquad i \in C, \tag{8}$$

$$d_i x_{ij} \leq f_{ij} \leq (Q - d_j)x_{ij}, \qquad (i, j) \in A, \tag{9}$$

$$\sum_{j:(i,j)\in A} g_{ij} \geq s_i + \sum_{h:(h,i)\in A} \left(g_{hi} + t_{hi}x_{hi}\right), \qquad i \in C, \tag{10}$$

$$(a_i + s_i)x_{ij} \leq g_{ij} \leq (b_i + s_i)x_{ij}, \qquad (i,j) \in A, \qquad (11)$$

$$x_{ij} \in \{0,1\}, \qquad (i,j) \in A. \qquad (12)$$

The objective function (5) consists of minimizing the total traveling costs. Constraints (6) ensure that each customer is visited only once, whereas (7) guarantee the correct vehicle flow through the nodes. Constraints (8) ensure the load flow propagation, enforcing that the load on the vehicle that leaves node i increases by its demand d_i. These constraints also forbid subtours. Constraints (9) prevent the vehicle from exceeding its capacity. Constraints (10) act similarly to (8) and guarantee the time flow propagation through the visited nodes of a route. Constraints (11) ensure that time windows are met. They also guarantee that the variable g_{ij} is non-negative if arc (i,j) is traversed. Finally, constraints (12) define the binary domain of variables x_{ij}.

3.2 CF Formulation for the Cardinality-Constrained RVRPTW

We obtain the robust counterpart of model (5)–(12) following the idea of the linearization technique [Munari et al., 2019]. Similarly to the steps performed in the MTZ-based formulation, we add an index γ to the variables that control the load and time propagation. Hence, let variables $f_{ij\gamma}$ and $g_{ij\gamma}$ represent the worst-case load and elapsed time of the vehicle that traverses arc $(i,j) \in A$, considering that γ parameters attain their worst case simultaneously. With these variables, we can redefine the load and time flow propagation constraints of the model. First, constraints (8) and (9) are replaced with:

$$\sum_{j:(i,j)\in A} f_{ij\gamma} \geq \bar{d}_i + \sum_{h:(h,i)\in A} f_{hi\gamma}, \qquad i \in C, \gamma \leq \Gamma^d, \qquad (13)$$

$$\sum_{j:(i,j)\in A} f_{ij\gamma} \geq \bar{d}_i + \hat{d}_i + \sum_{h:(h,i)\in A} f_{hi(\gamma-1)}, \qquad i \in C, 1 \leq \gamma \leq \Gamma^d, \qquad (14)$$

$$\bar{d}_i x_{ij} \leq f_{ij\gamma} \leq (Q - \bar{d}_j)x_{ij}, \qquad (i,j) \in A, \gamma \leq \Gamma^d. \qquad (15)$$

Constraints (13) and (14) guarantee the propagation of the worst-case vehicle load, according to the following two cases: the demand of γ nodes attained their worst-case previously, and then only the nominal demand of node i happens, as computed in the right-hand side of (13); or $\gamma-1$ worst-case

realizations occurred previously and then the demand of node i also attains its maximum deviation (hence, one more worst-case value considered), as calculated in the right-hand side of (14). Constraints (15) ensure that the vehicles' capacity is satisfied for all possible number of worst-case realizations. Note that, in the deterministic case ($\Gamma^d = 0$), constraints (14) are not defined and constraints (13) and (15) are the same as (8) and (9).

Likewise, we can apply the same process to the constraints associated with time flow and, then, we obtain the following RO model for the RVRPTW with uncertainty on demands and travel times. The model consists of the objective function (5) subject to (6), (7), (12)–(15), and to:

$$\sum_{j:(i,j)\in A} g_{ij\gamma} \geq s_i + \sum_{h:(h,i)\in A} (g_{hi\gamma} + \bar{t}_{hi}x_{hi}), \qquad i \in C, \gamma \leq \Gamma^t, \quad (16)$$

$$\sum_{j:(i,j)\in A} g_{ij\gamma} \geq s_i + \sum_{h:(h,i)\in A} (g_{hi\gamma-1} + (\bar{t}_{hi} + \hat{t}_{hi})x_{hi}), \quad i \in C, 1 \leq \gamma \leq \Gamma^t, \quad (17)$$

$$(s_i + a_i)x_{ij} \leq g_{ij\gamma} \leq (b_i + s_i)x_{ij}, \qquad (i,j) \in A, \gamma \leq \Gamma^t. \quad (18)$$

Constraints (16) and (17) act similarly to (13) and (14) but for time load propagation. Constraints (18) ensure that the time windows are respected.

Thanks to the capacity and time windows constraints (15) and (18), only one variable related to load ($f_{ij\gamma}$) and time ($g_{ij\gamma}$) propagation are allowed to have a non-null value for each i and γ, specifically the one related to arc (i,j) where $x_{ij} = 1$. Thus, suppose that in an optimal solution, we have $x_{i_1j_1} = 1$ and $x_{j_1k_1} = 1$. Then, timing constraints (16) and (17) related to node j_1, for any $\gamma > 0$, can be simply represented as follows:

$$g_{j_1k_1\gamma} \geq g_{i_1j_1\gamma} + \bar{t}_{i_1j_1} + s_{j_1}, \quad (19)$$

$$g_{j_1k_1\gamma} \geq g_{i_1j_1\gamma-1} + \bar{t}_{i_1j_1} + \hat{t}_{i_1j_1} + s_{j_1}. \quad (20)$$

Conversely, in the MTZ-based model for the RVRPTW [Munari et al., 2019], the time constraints related to node j_1 in path (i_1, j_1, k_1) can be written as follows:

$$w_{j_1\gamma} \geq w_{i_1\gamma} + \bar{t}_{i_1j_1} + s_{j_1}, \quad (21)$$

$$w_{j_1\gamma-1} \geq w_{i_1\gamma} + \bar{t}_{i_1j_1} + \hat{t}_{i_1j_1} + s_{j_1}, \quad (22)$$

where w_{j_1} is a continuous variable that represents the departure times from node j_1. Given that variables $g_{j_1k_1\gamma}$ and w_{j_1} have the same meaning, it is

possible to see that constraints (19)–(20) are equivalent to (21)–(22). A similar conclusion can be drawn for the load constraints. Hence, both models have equivalent load and time propagation constraints in an integer solution. However, our CF-based formulation avoids the weak big-M constraints presented in the MTZ-based models. Indeed, as we will demonstrate later, our new formulation yields a much tighter relaxation. Finally, it is worth mentioning that model (5)–(7), (12)–(15) is a valid RO formulation for the RCVRP under the cardinality-constrained uncertainty set.

3.3 Formulations for the Knapsack-Constrained RVRPTW

We can also derive MTZ- and CF-based formulations for the RVRPTW under single and multiple knapsack uncertainty sets using the linearization approach. Recall that Δ^d and Δ^t are the budgets of uncertainty in demand and travel times, respectively, used in the definition of the single knapsack uncertainty sets in Section 2.1.2. Similarly, we have Δ^d_l and Δ^t_l for the multiple knapsack uncertainty sets.

3.3.1 MTZ-Based Formulation for the Single Knapsack Uncertainty Set.

To generate the MTZ-based formulation for the RVRPTW under the single knapsack uncertainty set, we define the following decision variables:

- $u_{i\delta}$: represents the load on the vehicle up to and including node $i \in N$, considering up to a total deviation of $\delta \in \{0, 1, \ldots, \Delta^d\}$ units over the demands' nominal values for all nodes previously visited;
- $w_{i\delta}$: indicates the earliest time that a vehicle can start the service at node $i \in N$, considering up to a total deviation $\delta \in \{0, 1, \ldots, \Delta^t\}$ time units over the travel times' nominal values for all arcs previously traversed.

In these definitions, the index δ represents the total load/time over their nominal values accumulated in the route up to the current node. This interpretation requires integer budget and deviations, since an index is used to represent the deviations. Moreover, for large budgets, the number of variables and constraints becomes large as they grow pseudopolynomially. Later in this section we discuss some strategies to address these issues.

To derive constraints based on variables $u_{i\delta}$ and $w_{i\delta}$, we rely on the following interpretation based on DP, inspired by the discussion presented in Sec-

tion 2.2 for the cardinality-constrained uncertainty set. Let $r = (v_0, v_1, \ldots, v_h)$ be a route visiting $h-1$ customers. For this route, we can compute the values of $w_{v_j\delta}$ for $j = 0, \ldots, h$ and $\delta = 0, 1, \ldots, \Delta^t$ as follows:

$$w_{v_j\delta} = \begin{cases} a_{v_0}, & \text{if } j = 0, \\ \max\limits_{0 \le \lambda \le \min\{\delta, \hat{t}_{v_{j-1}v_j}\}} \{a_{v_j}, w_{v_{j-1}(\delta-\lambda)} + \bar{t}_{v_{j-1}v_j} + s_{v_{j-1}} + \lambda\}, & \text{otherwise.} \end{cases} \tag{23}$$

The first case in (23) is a boundary condition for the first node in the route, the depot, and defines the starting time of the route. The second computes the worst-case starting time of the service at node v_j, considering that the travel time deviation in arc (v_{j-1}, v_j), represented by λ, may attain any value from 0 to the minimum between the total budget δ and the maximum deviation $\hat{t}_{v_{j-1}v_j}$. Notice that this expression encompasses the deterministic cases for $\delta = 0$ and $\lambda = 0$. Additionally, this calculation accounts for the opening of the time window. To check the feasibility of the route with respect to time windows, we need to verify after each iteration of the DP algorithm if $w_{v_j\delta} \le b_{v_j}$ for each node v_j and $\delta \in \{0, 1, \ldots, \Delta^t\}$. In the compact model, these verifications are introduced as the upper bound of the time windows constraints.

It is possible to derive a similar expression for computing the values of variables $u_{v_j\delta}$. However, since the vehicle load is not subject to a behavior analogous to the opening of the time windows, the worst-case load at a given node can be computed by filling the knapsack with as much demand deviations as possible in the order they appear in the route. Hence, we use the following improved equation:

$$u_{v_j\delta} = \begin{cases} \bar{d}_{v_0}, & \text{if } j = 0, \\ u_{v_{j-1}\delta} + \bar{d}_{v_j}, & \text{if } \delta < \hat{d}_{v_j}, \\ \max\{u_{v_{j-1}\delta} + \bar{d}_{v_j}, u_{v_{j-1}(\delta-\hat{d}_{v_j})} + \bar{d}_{v_j} + \hat{d}_{v_j}\}, & \text{if } \hat{d}_{v_j} \le \delta < \Delta^d, \\ \max\limits_{0 \le \lambda \le \min\{\hat{d}_{v_j}, \Delta^d\}} \{u_{v_{j-1}(\Delta^d-\lambda)} + \bar{d}_{v_j} + \lambda\}, & \text{otherwise,} \end{cases} \tag{24}$$

for $j = 0, \ldots, h$ and $\delta = 0, 1, \ldots, \Delta^d$. The first two cases define boundary conditions: one sets the total load in the first node in the route, which is usually the depot; whereas the other applies when the total deviation considered in the route is lower than $\hat{d}_{v_j}$, thus resulting in adding only the nominal demand of node v_j to the total load. We do not introduce any

deviations in the variables with small δ because the algorithm focuses on finding $u_{v_j\Delta^d}$, which is properly calculated by the remaining cases. The third case corresponds to whether or not we consider the full deviation of the demand on node v_j, with the largest between both values being chosen. Finally, the last case corresponds to $\delta = \Delta^d$, and involves using the remaining budget of the knapsack. It considers all the possible deviations from 0 to $\hat{d}_{v_j}$, denoted as λ, as long as this value does not exceed the budget Δ^d. The value λ represents the deviation added to the knapsack to fill it up.

With this interpretation of the variables and relying on the linearization approach to convert the recursive equations (23) and (24) to linear constraints, we developed the following model for the RVRPTW under the single knapsack uncertainty set:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij}, \tag{25}$$

s.t. (6), (7), (12), and to

$$u_{j\delta} \geq u_{i\delta} + \bar{d}_j + Q(x_{ij} - 1), \qquad (i,j) \in A, \delta = 0, \ldots, \Delta^d, \tag{26}$$

$$u_{j\delta} \geq u_{i(\delta-\hat{d}_j)} + \bar{d}_j + \hat{d}_j + Q(x_{ij} - 1), \qquad (i,j) \in A, \delta = \hat{d}_j, \ldots, \Delta^d, \tag{27}$$

$$u_{j\Delta^d} \geq u_{i(\Delta^d-\lambda)} + \bar{d}_j + \lambda + Q(x_{ij} - 1), \qquad (i,j) \in A, \lambda = 0, \ldots, \min\{\hat{d}_j, \Delta^d\}, \tag{28}$$

$$0 \leq u_{j\delta} \leq Q, \qquad j \in N, \delta = 0, \ldots, \Delta^d, \tag{29}$$

$$w_{j\delta} \geq w_{i(\delta-\lambda)} + \bar{t}_{ij} + \lambda + s_i + b_{n+1}(x_{ij} - 1), \qquad (i,j) \in A, \delta = 0, \ldots, \Delta^t, \lambda = 0, \ldots, \min\{\delta, \hat{t}_{ij}\}, \tag{30}$$

$$a_j \leq w_{j\delta} \leq b_j, \qquad j \in N, \delta = 0, \ldots, \Delta^t. \tag{31}$$

The objective function (25) is the same as in the other formulations presented in this work. Constraints (26)–(28) determine the worst-case load at node j considering a non-negative integer budget $\delta \leq \Delta^d$, and they are the linear counterpart of the recursive equations (24). Constraints (29) ensure that the vehicle capacity is respected. Notably, we just need to verify the upper bound for $\delta = \Delta^d$. If we consider only these constraints and the ones related to the variable's domain, we formulate the RCVRP. Constraints (30) compute the worst-case service starting time using the same strategy as in the recursive equations (23). They compute the arrival time in node j considering the route's total deviation of δ. The right-hand side evaluates the quantity λ of time deviation that should be considered for that particular node, while the

remaining $\delta - \lambda$ units of deviation happened in previous nodes of the route, thus $w_{j\delta}$ is determined by the choice of λ that results in the worst-case arrival time. Finally, constraints (31) ensure the time windows are respected.

3.3.2 CF Formulation for the Single Knapsack Uncertainty Set.

To derive a robust counterpart of the deterministic CF model (5)–(11) considering knapsack-uncertainty on demand and travel time, we define the following load and time variables:

- $f_{ij\delta}$: the load carried on arc $(i,j) \in A$ with a total deviation of $\delta \in \{0, 1, \ldots, \Delta^d\}$ units over the nominal demand;
- $g_{ij\delta}$: earliest elapsed time of a route when the vehicle begins to traverse arc $(i,j) \in A$ after serving node i with a total deviation of $\delta \in \{0, 1, \ldots, \Delta^t\}$ units over nominal travel times.

Using a similar interpretation as that presented in the previous section, we can derive CF-based constraints from the recursive equations (23) and (24), resulting in the following new CF formulation for the RVRPTW.

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}, \tag{32}$$

s.t. (6), (7), (12), and to

$$\sum_{j:(i,j)\in A} f_{ij\delta} \geq \bar{d}_i + \sum_{h:(h,i)\in A} f_{hi\delta}, \qquad i \in N, \delta = 0, \ldots, \Delta^d, \tag{33}$$

$$\sum_{j:(i,j)\in A} f_{ij\delta} \geq \bar{d}_i + \hat{d}_i + \sum_{h:(h,i)\in A} f_{hi\delta-\hat{d}_i}, \qquad i \in N, \delta = \hat{d}_i, \ldots, \Delta^d, \tag{34}$$

$$\sum_{j:(i,j)\in A} f_{ij\Delta^d} \geq \bar{d}_i + \lambda + \sum_{h:(h,i)\in A} f_{hi\Delta^d-\lambda}, \qquad i \in N, \lambda = 0, \ldots, \min\{\Delta^d, \hat{d}_i\}, \tag{35}$$

$$\bar{d}_i x_{ij} \leq f_{ij\delta} \leq (Q - \bar{d}_j) x_{ij}, \qquad (i,j) \in A, \delta = 0, \ldots, \Delta^d, \tag{36}$$

$$\sum_{j:(i,j)\in A} g_{ij\delta} \geq s_i + \sum_{\substack{h:(h,i)\in A \\ \lambda \leq \hat{t}_{hi}}} (g_{hi\delta-\lambda} + (\bar{t}_{hi} + \lambda) x_{hi}), \qquad i \in N, \delta = 0, \ldots, \Delta^t, \lambda = 0, \ldots, \delta, \tag{37}$$

$$(a_i + s_i) x_{ij} \leq g_{ij\delta} \leq (b_i + s_i) x_{ij}, \qquad (i,j) \in A, \delta = 0, \ldots, \Delta^t. \tag{38}$$

The objective function (32) is the same as in the previous formulations. Constraints (33)–(35) are based on the recursive equations (24) and work similarly to (26)–(28), forbidding subtours and computing the worst-case load values using the CF variables. Capacity constraints are imposed via (36). Constraints (37) compute the elapsed time based on the idea of equations (23), and constraints (38) enforce the time windows.

We mention a few difficulties that the models based on single knapsack uncertainty sets might face and some possible solutions. The first issue regards the large number of variables when the budgets are large. However, budget values in real contexts are typically not large. For instance, Bartolini et al. [2021] consider $\Delta^d = 100$ minutes of deviation as an extreme case. This would be considered a volatile environment since the delay would take roughly 20% of a worker's day. Hence, smaller values of Δ^d are usually reasonable in practice. Another possible workaround is to change the order of magnitude of δ; for instance, instead of using the unit of δ as 1 minute, one can use it in units of 5 minutes. This strategy may reduce the model to a tractable size. Another limitation of the proposed formulations is the impossibility of using non-integer units for δ. The mentioned workaround may be used in this case as well, by multiplying the non-integer unit by a constant that turns it into an integer number, possibly at the cost of worsening the computational performance. Nevertheless, we believe the proposed approaches are still of theoretical and practical value, and may benefit researchers and practitioners interested in formulations for RVRP variants.

3.3.3 Formulations for the Multiple Knapsack Uncertainty Set.

In the multiple knapsack uncertainty set, we consider a set L of knapsacks and subsets of nodes (or arcs) associated with each knapsack (S_l, $l \in L$) with its own budget Δ_l. To extend the previous formulations to a multiple knapsack framework, we redefine the continuous variables related to demand ($u_{i\delta}$ and $f_{ij\delta}$) and time ($w_{i\delta}$ and $g_{ij\delta}$) to consider an index δ_l representing the total deviation in the route for each knapsack l. For the sake of conciseness, we present these formulations in the appendix.

4 Branch-and-cut Algorithm for the RVRPTW

In addition to the compact formulations, we designed a tailored BC algorithm to obtain better computational performance for the studied models. The cut separation is performed using a combination of strategies: a DP-based algorithm, a heuristic algorithm proposed for the RCVRP [Gounaris et al., 2013], and the CVRPSep package [Lysgaard et al., 2004]. We use the latter to generate rounded capacity inequalities (RCI) only, which are stated as follows for a given set of nodes V_S:

$$\sum_{i \in N \setminus V_S} \sum_{j \in V_S} x_{ij} \geq \left\lceil \frac{\sum_{j \in V_S} \bar{d}_j}{Q} \right\rceil, \qquad V_S \subset C. \tag{39}$$

These cuts ensure that the number of vehicles entering set V_S offer enough capacity to serve all the nodes in V_S. Since enumerating every possible set V_S requires a long computational time, CVRPSep uses an efficient heuristic algorithm to generate a limited number of sets and relevant cuts.

Other than that, for instances with uncertainty on demand we also use the Robust RCI (RRCI) introduced by Gounaris et al. [2013]. These cuts are a robust extension of the capacity cut previously explained, represented by the following inequalities:

$$\sum_{i \in N \setminus V_S} \sum_{j \in V_S} x_{ij} \geq \left\lceil \frac{1}{Q} \max_{d \in U^d} \sum_{j \in V_S} d_j \right\rceil, \qquad V_S \subset C. \tag{40}$$

Note that these are almost identical to the deterministic version (39), the main difference being that, in the robust case, we consider the maximum possible demand inside the uncertainty set instead of the nominal demand. Since enumerating every possible RRCI would be impractical, we implemented a heuristic algorithm to dynamically separate and insert these cuts as needed, as proposed by Gounaris et al. [2013]. We start with a solution and a randomly generated set of customers V_S; then, we iteratively perturb this set by inserting or removing a node from it. In each attempt of adding/removing a node to/from V_S, we analyze every potential customer and remove or insert the one with the highest impact in the difference between the right-hand side and the left-hand side of the corresponding RRCI constraint. We also maintain a tabu list of recently added/removed customers that are not allowed to be moved in or out of the set for some iterations, to avoid cycles. We stop

the algorithm when we do not improve the difference between the right- and the left-hand sides of the inequality for a given number of iterations.

An important factor in this algorithm is the need to efficiently compute the right-hand side of the inequality since this value must be frequently checked. To assist in this calculation, we create an auxiliary data structure that reduces the number of steps required. For the cardinality-constrained uncertainty set, we define an auxiliary vector $\mathcal{Q} \in \mathbb{R}^{|V_S|}$, $\mathcal{Q} = (\hat{d}_{[1]}, \ldots, \hat{d}_{[|V_S|]})$, containing the demand deviations of all nodes in V_S in non-increasing order, i.e., $\hat{d}_{[j-1]} \geq \hat{d}_{[j]}$, where $\hat{d}_{[j]}$ is the demand deviation in the jth position of this vector. We then compute D_{V_S}, the maximum demand of this set, using

$$D_{V_S} = \sum_{j \in V_S} \bar{d}_j + \sum_{\gamma=0}^{\min\{\Gamma^d, |V_S|\}} \mathcal{Q}_\gamma. \tag{41}$$

Then, for each iteration when we check the possibility of insertion/removal of a given node $j \in V_S$, we take the total demand of the current set V_S (D_{V_S}) and compute the new right-hand side for that particular node (RS_j) using the following expression:

$$RS_j = \begin{cases} D_{V_S} + \bar{d}_j + \max(\hat{d}_j - \mathcal{Q}_{\Gamma^d}, 0), & \text{if } j \notin V_S; \\ D_{V_S} - \bar{d}_j - \max(\hat{d}_j - \mathcal{Q}_{\Gamma^d+1}, 0), & \text{otherwise,} \end{cases} \tag{42}$$

After determining which node j will be inserted/removed from set V_S, we update D_{V_S}, which takes the value of RS_j, and insert/remove its deviation from $\mathcal{Q}$. Note that we did not need to compute D_{V_S} again with the equation (41), we only need to use it in the first iteration reducing the number of operations in the procedure.

For the single and multiple knapsack uncertainty sets, we used the formulas proposed by Gounaris et al. [2013] to calculate the maximum demand deviation, which is done in $\mathcal{O}(|V_S|)$. Let L be the set of knapsacks and S_l the subset of nodes inside each knapsack $l \in L$. The right-hand side of the RRCI is then given by:

$$\sum_{i \in V_S} \bar{d}_i + \sum_{l \in L} \min\{\Delta_l^d, \sum_{i \in V_S \cap S^l} \hat{d}_i\}. \tag{43}$$

Since this is a heuristic algorithm, it might miss a violated constraint. Thus, for integer solutions we also use a DP algorithm based on the recursive

equations (23), which can be executed in $\mathcal{O}(|N|\prod_{l\in L}\Delta_l^d)$. Whenever the heuristic is unable to find a new violated cut, this algorithm checks if the solution is feasible and inserts additional feasibility cuts if needed. If for a given route $r = (v_0, v_1, \ldots, v_h)$ the infeasibility is detected in node v_j for a given $j \in \{1, \ldots, h\}$ the following inequality is inserted into the problem:

$$\sum_{0<i\leq j} y_{v_{i-1}v_i} \leq j - 1. \tag{44}$$

We used a different approach for time uncertainty. We initialize the model with only the deterministic time flow propagation constraints and dynamically insert the robust constraints (16) and (17) whenever they are violated. To efficiently check the feasibility of a solution, we use DP algorithms based on equations (1) and (24) to compute the worst-case elapsed times and then identify whether time windows are violated in the route. These algorithms have complexity $\mathcal{O}(|N|\Gamma^t)$ for the cardinality constrained uncertainty set and $\mathcal{O}(|N|\prod_{l\in L}\Delta_l^t)$ for the knapsack set.

5 Computational Experiments

In this section, we present the results of extensive computational experiments performed to assess the proposed compact models and BC algorithms for the cardinality-constrained and single-knapsack uncertainty sets, using benchmark instances from the literature. We compare the computational performance of these approaches against the MTZ-based formulation of Munari et al. [2019] for the cardinality-constrained set, which is the state-of-the-art compact model for the RVRPTW. Additionally, we analyze the impact of robustness regarding the different uncertainty sets. All models and algorithms were coded in C++ using the Concert library of IBM CPLEX Optimization Studio version 20.1 with default parameters. All experiments were performed on a Linux PC with Intel Core i7-8700 CPU @ 3.60 GHz processor and 16GB of RAM. We imposed a time limit of 3600 seconds for each run.

We use the same benchmark instances of the RVRPTW with 25 customers introduced by Munari et al. [2019], who adapted the VRPTW instances of Solomon [1987] to include uncertainty in demands and travel times. Each uncertain parameter is defined by its nominal value and maximum deviation. The maximum deviations for demand (Dev^d) and travel time (Dev^t) are defined as 10%, 25%, and 50% of the nominal value, truncated on the first dec-

imal place. In the experiments with the cardinality-constrained uncertainty set, the budgets for demand (Γ^d) and travel time (Γ^t) assume the values of 0, 1, 5, and 10, where 0 is the deterministic case. We run experiments with three different configurations of these parameters: uncertainty on demand only ($\Gamma^d > 0, \Gamma^t = 0$), uncertainty on travel time only ($\Gamma^d = 0, \Gamma^t > 0$), and uncertainty on both demand and travel times ($\Gamma^d > 0, \Gamma^t > 0$). For the latter, the budgets for both parameters are the same, i.e., $\Gamma^d = \Gamma^t$. Similar configurations are used for the single-knapsack uncertainty set, with budgets $\Delta^d, \Delta^t \in \{0, 20, 40, 60\}$. In all experiments, if the budget of uncertainty for a specific parameter (demand or travel time) in the instance is zero, the maximum deviation for that parameter (Dev^d or Dev^t) will also be zero. These combinations of budget and deviations result in 56 instances for the deterministic cases (i.e., when $\Gamma^d = \Gamma^t = 0$ and $\Delta^d = \Delta^t = 0$) and 168 instances for each configuration of budgets with positive values. However, for experiments with the cardinality-constrained RVRPTW having positive Γ^t, five instances became infeasible and were discarded (hence, we have 163 instances in experiments with configurations in which $\Gamma^t > 0$). The detailed results are available online at `www.dep.ufscar.br/munari/rvrptw`.

5.1 Computational Performance of the Compact Formulations

5.1.1 Cardinality-Constrained Uncertainty Set.

Table 1 summarizes the results of the proposed CF formulation and the MTZ-based formulation of Munari et al. [2019] for the cardinality-constrained RVRPTW. We performed two experiments with both models: solving the LP relaxations of these models and solving the models using the general-purpose MIP solver. For the first experiment, the table presents the average objective value (Obj), the quality of the LP relaxation (QLR), and the average computational time in seconds (T). QLR is the percentage of the LP relaxation value relative to the obtained integer solution value. For the MIP solver results, in addition to the average objective values and computational times, the table shows the average optimality gap (Gap), as provided by the solver at the end of the execution, and the number of instances that were solved to optimality (Opt). The results are grouped according to different configurations of Γ^d and Γ^t.

Regarding the LP relaxation of the formulations, we observe that CPLEX

Table 1: Results of the MTZ-based and CF Formulations for the Cardinality-Constrained RVRPTW, Considering Different Configurations of Budgets

		LP Relaxation						MIP Model							
		MTZ			CF			MTZ				CF			
Γ^d	Γ^t	Obj	QLR (%)	T (s)	Obj	QLR (%)	T (s)	Obj	Gap (%)	T (s)	Opt	Obj	Gap (%)	T (s)	Opt
0	0	177.23	53.5	0.011	267.43	80.7	0.030	331.27	3.7	577.04	48	331.27	0.7	282.68	52
1	0	177.23	53.0	0.013	267.68	80.0	0.060	334.63	4.3	667.27	143	334.63	0.9	357.69	154
5	0	177.23	50.5	0.024	268.60	76.6	0.208	350.73	7.2	956.74	130	351.01	3.6	1223.47	119
10	0	177.25	50.2	0.026	270.19	76.6	0.481	352.75	8.4	1126.01	126	354.20	5.5	1675.54	102
0	1	177.37	53.6	0.014	261.58	79.1	0.083	330.74	3.5	564.19	142	330.31	0.9	325.05	151
0	5	177.50	52.7	0.028	262.73	77.9	0.296	337.08	4.8	697.58	135	335.23	1.9	921.29	131
0	10	177.52	52.4	0.032	263.53	77.8	0.747	338.75	5.6	835.98	132	337.47	3.3	1335.51	115
1	1	177.37	53.1	0.021	261.81	78.3	0.106	334.33	4.4	649.30	137	334.04	1.2	454.73	148
5	5	177.50	50.3	0.053	263.79	74.7	0.513	353.00	7.5	994.17	125	353.68	5.2	1700.40	97
10	10	177.01	49.9	0.058	266.00	75.0	3.973	354.83	8.8	1297.26	117	359.18	9.4	2276.58	73
All		177.32	51.9	0.028	265.33	77.7	0.650	341.81	5.8	836.55	1235	342.10	3.3	1055.29	1142

was considerably faster with the MTZ model, whereas the CF model resulted in significantly stronger linear relaxation. On average, the value of the linear relaxation for the CF model is 77.7% of the integer solution against only 51.9% for the MTZ formulation. This behavior was expected since the CF formulations in the literature are known for having tighter LP relaxations than those based on big-M parameters [Letchford and Salazar-González, 2015].

For the new MIP formulations, while the MTZ model yielded shorter run times and 8.1% more instances solved to optimality, the CF presented lower average optimality gaps in all but one of the combinations of Γ^d and Γ^t, better ensuring the quality of the solution obtained. Hence, when the MTZ model does not solve an instance to optimality, the gaps are considerably larger than those of the CF formulation, which is usually closer to proving optimality given its stronger linear relaxation. Furthermore, the CF formulation solved more instances to optimality in the deterministic case ($\Gamma^d = \Gamma^t = 0$) in about than half the time on average. The CF model also outperformed the MTZ one in terms of computational times and number of optimal solutions in instances where the uncertainty budgets (Γ^d and Γ^t) are less than or equal to 1. Finally, we also identified the traditional behavior found in the RO literature [Ordonez, 2010, Agra et al., 2013, Munari et al., 2019] where increasing the budget also tends to increase the time it takes to solve the problem. This is expected since the budget directly impacts the number of variables and constraints of the formulations. This behavior is noticeable in both models and for any combination of Γ^t and Γ^d.

5.1.2 Knapsack Uncertainty Set.

Similarly to the previous uncertainty set, we first compare our proposed formulations regarding computational performance. Table 2 follows a similar structure to Table 1, but for different configurations of the knapsack set budgets Δ^d and Δ^t.

Table 2: Results of the MTZ-based and CF formulations for the single knapsack-constrained RVRPTW, considering different configurations of budgets.

		LP Relaxation						MIP Model							
		MTZ			CF			MTZ				CF			
Δ^d	Δ^t	Obj	QLR (%)	T (s)	Obj	QLR (%)	T (s)	Obj	Gap (%)	T (s)	Opt	Obj	Gap (%)	T (s)	Opt
0	0	165.78	49.1	0.069	272.68	80.8	0.048	337.59	3.8	592.70	26	337.59	0.9	259.87	28
20	0	165.78	47.9	0.091	273.22	78.9	4.464	346.25	7.4	1126.48	71	349.17	5.7	1809.10	57
40	0	165.78	46.6	0.169	273.62	76.9	20.753	355.96	11.4	1630.44	52	366.64	14.6	2206.55	37
60	0	165.82	46.1	0.298	274.03	76.2	43.758	359.41	13.3	1727.39	51	371.22	17.2	2502.04	31
0	20	165.92	48.9	1.148	272.81	80.3	87.620	339.54	9.7	1491.93	56	493.90	24.2	2451.28	27
0	40	166.00	48.1	4.110	280.62	81.3	923.937	345.17	13.1	1840.01	51	1168.65	63.0	2815.86	20
0	60	166.04	47.4	6.162	293.67	83.8	1194.765	350.28	15.8	2131.26	42	1025.75	53.2	2780.25	14
20	20	165.92	47.4	1.220	273.35	78.2	106.106	349.73	12.1	1644.30	56	568.60	29.1	2543.69	27
40	40	166.00	45.2	3.983	281.61	76.7	1011.130	367.19	17.7	2110.46	41	1170.30	63.0	3105.86	16
60	60	166.94	44.8	5.668	318.16	85.4	2149.707	372.34	20.3	2307.84	38	1271.08	69.5	3128.97	6
All		166.00	47.2	2.292	281.38	79.9	554.229	352.35	12.4	1660.28	484	3422.91	34.0	2360.35	263

Similar conclusions as with the previous uncertainty set can be drawn from Table 2 regarding the quality of the LP relaxation and computational performance of the models. Particularly, the CF formulation presented stronger LP relaxations overall, with a QLR of 79.9%, but presented worse results as a MIP model, since it resulted in fewer instances solved to optimality (263) than the MTZ model (484), in longer running times (2360.35 seconds against 1660.28 seconds from the MTZ model), and in larger average gaps (34% versus 12.4%). This is a consequence of the CF model requiring longer running times when solving the LP relaxation for instances with larger Δ values (e.g., 1194.76 seconds for $\Delta^d = 0$ and $\Delta^t = 60$, and 2149.71 seconds for $\Delta^d = \Delta^t = 60$), hindering the CF formulation efficiency. Notably, instances with travel time deviation were harder to solve than those with deviation exclusively on demand, which is a consequence of the models having more constraints related to time flow propagation, as we cannot apply the constraint reduction strategy used for the demand uncertainty, in which we fill the knapsacks as quickly as possible.

5.2 Computational Performance of the BC Algorithms

5.2.1 Cardinality-Constrained Uncertainty Set.

We now analyze the results of the BC algorithms developed for both models introduced in Section 4. Table 3 summarizes the results using a similar structure to the Table 1. We do not show results for the LP relaxation, as they are similar to those presented in Table 1.

Table 3: Average Objective Values and Computational Times of the BC Algorithm for the Cardinality-Constrained RVRPTW

		MTZ				CF			
Γ^q	Γ^t	Obj	Gap (%)	T (s)	Opt	Obj	Gap (%)	T (s)	Opt
0	0	331.27	0.6	224.01	53	331.27	0.7	279.52	52
1	0	334.63	0.9	268.11	158	334.63	0.8	299.43	156
5	0	350.68	0.8	304.31	157	350.73	0.9	426.13	153
10	0	354.92	1.2	509.17	147	354.93	1.4	651.36	144
0	1	330.74	0.7	235.06	154	330.73	0.7	307.42	151
0	5	337.08	1.1	334.21	150	337.14	1.4	563.33	143
0	10	338.64	1.3	392.45	146	338.88	1.9	700.17	137
1	1	334.36	1.0	341.11	152	334.35	0.9	384.56	149
5	5	353.00	1.2	495.85	146	353.53	1.6	895.58	132
10	10	356.44	1.5	585.00	139	356.72	2.0	1085.92	123
All		342.18	1.0	368.93	1402	342.29	1.2	559.34	1340

As expected, the BC algorithms outperformed their corresponding compact models as we start with a deterministic model and add cuts as they are needed, thus having a model with considerably fewer constraints. The average solution times and gaps decreased, while the number of optimal solutions increased in all configuration of budgets. For example, the average solution time for the MTZ-based model decreased from 836.55 seconds to 368.93 seconds, the average gap reduced from 5.8% to 1.0%, and the number of instances solved to optimality increased from 1235 to 1402. Similarly, the average time of the BC algorithm for the CF model was almost halved, decreasing from 1055.29 to 559.34 seconds, the average gap decreased from 3.3% to 1.2%, and 198 more instances were solved to optimality, in comparison to solving the compact formulation.

The MTZ-based formulation still performed better than the CF one within the proposed BC, taking less time than the BC based on the CF model to solve the instances (368.93 seconds against 559.34) and finding more optimal solutions (1402 against 1340). Additionally, the BC algorithm with the

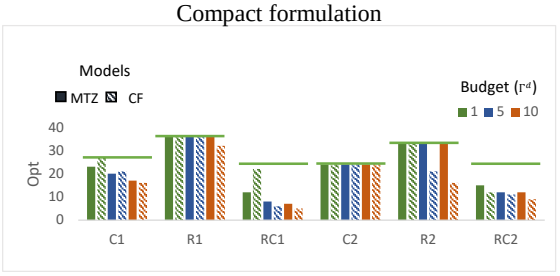
MTZ-based model was superior even when solving the compact formulation had some advantage, namely the results from instances with $\Gamma \leq 1$. Moreover, the optimality gaps were smaller, and more instances were solved to optimality in all configurations.

There are two main reasons for these superior results of the MTZ-based BC algorithm. The CVRPSep package strengthens the LP relaxation of the model, which is still quicker to solve than the LP relaxation of the CF model. Indeed, strengthening the LP relaxation with the valid inequalities of CVRPSep is particularly more helpful to the MTZ-based formulation, which usually has a weak LP relaxation. Consequently, this reduces one advantage of the CF model over the MTZ-based formulation. Moreover, since the solver usually obtains the optimal solution of the LP relaxation faster with the MTZ-based formulation, the search tree explores more nodes quicker and, hence, finds an optimal solution before the BC algorithm based on the CF model. Even when the BC algorithms do not prove optimality within the time limit, the MTZ-based BC algorithm usually shows a superior performance, as it tends to explore more nodes.

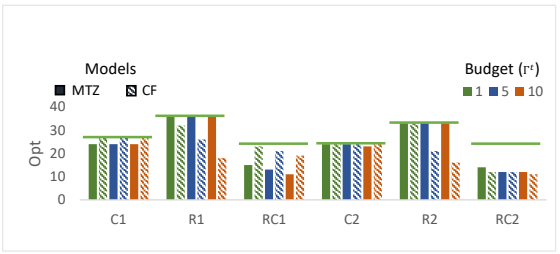
Figure 1 shows the overall performance of the compact formulations and BC algorithms, grouped by instance classes. Classes C1 and C2 include instances with clustered customers; R1 and R2 have instances with randomly positioned customers; and RC1 and RC2 have a combination of clustered and randomly positioned customers. Classes with suffix 2 have wider times windows and larger vehicle capacities. In each chart presented in the figure, a bar corresponds to the number of instances each approach could solve to optimality in that class and uncertainty budget. A horizontal line shows the number of instances in the class. As the charts indicate, the MTZ-based BC algorithm outperformed the CF-based one for most classes, even in C1 and RC1 where the compact models showed a better performance. Only in class C2 the CF-based method showed a slight advantage, as it solved all instances to optimality. Therefore, the MTZ-based formulation proved more suitable for the BC algorithm in the studied set of instances. Likewise, a decision-maker will probably find better results using this BC algorithm when working with the cardinality-constrained RVRPTW. If opting to use only the compact formulations, it is preferable to use the CF model if the instances have some degree of clustering.

Figure 1: Number of Optimal Solutions Obtained by the Compact Formulations and BC Algorithms per Instance Class and Uncertainty Budget for the Cardinality-Constrained RVRPTW

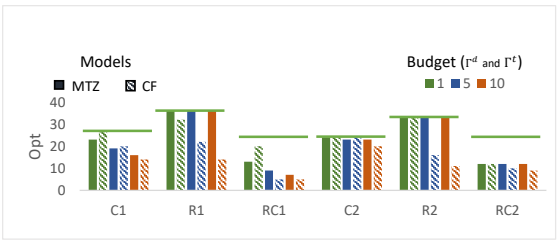
Compact formulation



a.1) Number of optimal solution per instance class for instances with deviation in demand

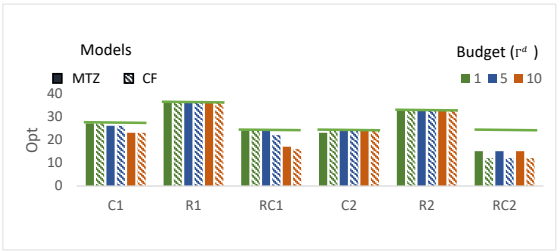


b.1) Number of optimal solution per instance class for instances with deviation in time

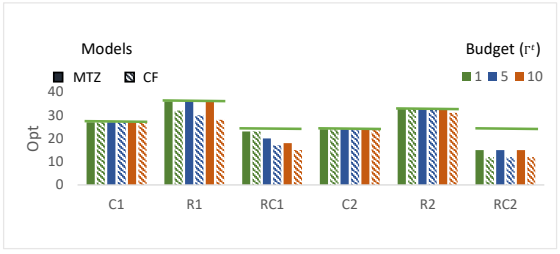


c.1) Number of optimal solution per instance class for instances with deviation in demand and time

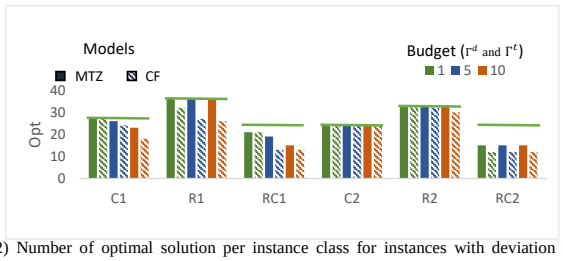
Branch-and-cut algorithm



a.2) Number of optimal solution per instance class for instances with deviation in demand



b.2) Number of optimal solution per instance class for instances with deviation in time



c.2) Number of optimal solution per instance class for instances with deviation in demand and time

Table 4: Average Objective Values and Computational Times of the BC Algorithm for the Single Knapsack-Constrained RVRPTW

		MTZ				CF			
Δ^d	Δ^t	Obj	Gap (%)	T (s)	Opt	Obj	Gap (%)	T (s)	Opt
0	0	331.27	0.6	224.01	49	331.27	0.6	224.01	53
20	0	341.58	7.7	1263.72	124	344.30	6.3	2104.34	91
40	0	350.00	11.1	1660.96	100	359.16	14.2	2480.07	58
60	0	352.93	12.7	1828.73	94	365.21	17.4	2733.56	49
0	20	333.88	9.7	1625.50	104	496.49	24.6	2799.64	38
0	40	341.72	14.5	2142.09	81	1233.65	69.8	3073.46	26
0	60	348.46	18.6	2478.64	63	1127.36	64.8	3045.10	19
20	20	345.43	12.2	1910.75	95	595.95	32.6	2951.88	34
40	40	363.23	19.0	2446.77	62	1247.16	70.3	3332.96	17
60	60	378.05	23.3	2645.94	52	1303.35	76.9	3386.76	6
All		318.84	12.9	1822.71	824	710.58	37.7	2613.18	391

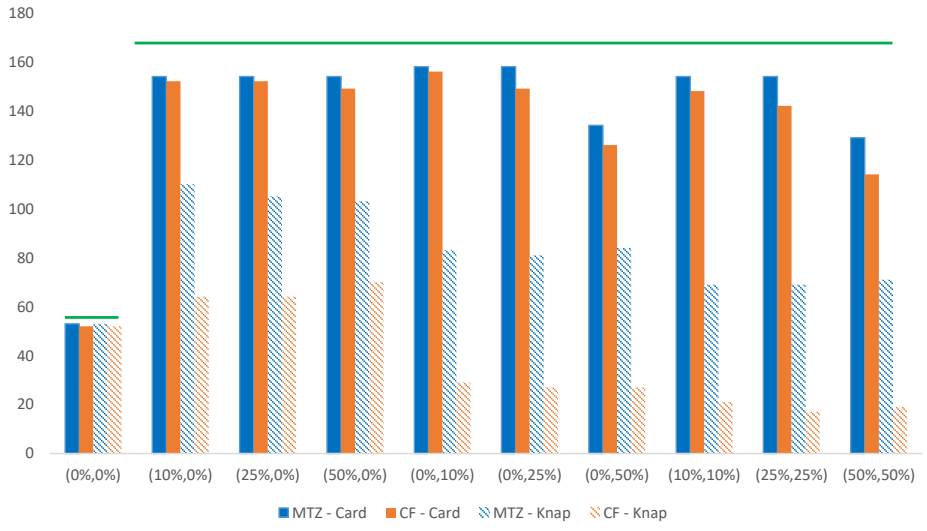
5.2.2 Knapsack Uncertainty Set.

Table 4 summarizes the results of the BC algorithms for the knapsack-constrained RVRPTW for different budget configurations. Most of the inferred conclusions for the cardinality-constrained uncertainty set also hold for the single knapsack set. The BC algorithms improved the computational performance of the models, indicated by the lower average running times and larger number of solved instances than using the compact models. More specifically, the BC algorithm with MTZ-based formulation solved 824 instances compared to only 484 solved by the general-purpose MIP solver with the compact formulation, and the BC with the CF formulation solved 391 instances to optimality compared to 263 instances with the compact model. Additionally, the better performance of the BC algorithm with the MTZ-based model is noticeable regarding computational times, number of instances solved to optimality, and optimality gaps. This is a consequence of the strengthening of the LP relaxation brought about by the algorithm, further assisted by the reduced size of the MTZ-based model compared to the CF one.

Both algorithms showed considerably greater difficulty in solving instances of the knapsack-constrained RVRPTW, compared to the deterministic counterpart as well as the cardinality-constrained RVRPTW. To help visualize this behavior, Figure 2 shows the number of instances solved to optimality for each pair of demand and travel time deviations (Dev^d, Dev^t), base mod-

els (MTZ and CF), and uncertainty sets (Card: cardinality constrained; or Knap: single knapsack). The horizontal green line over the bars shows the number of instances for that combination of Dev^d and Dev^t.

Figure 2: Number of Instances Solved to Optimality by the BC Algorithms for Each Pair of Demand and Travel Time Budgets (Dev^d,Dev^t) for the Cardinality- and Knapsack-Constrained RVRPTW



As the chart suggests, the BC algorithms on the cardinality-constrained uncertainty set proved optimality for considerably more instances than the methods with the knapsack uncertainty set. Since the proposed models index the uncertainty budget on the decision variables, by choosing larger budgets, the problem grows larger in number of variables and constraints, becoming more difficult to solve. Thus, as the chosen budgets for the knapsack uncertainty set (20, 40, and 60) are considerably larger than those used for the cardinality-constrained set (1, 5, and 10), instances defined for this set were considerably harder to solve. Moreover, in the knapsack uncertainty set, instances with deviation on travel time were considerably more difficult to solve, which is a consequence of the impossibility of reducing the DP equations for time, as can be done for the demand.

5.3 Robustness Analysis

We analyze the impact of the proposed RO approach regarding the objective function value and robustness of the solutions. The robustness of a solution is an important information in the decision-making process, as it can be used to evaluate the trade-off between cost and its "safety", which in this work means the chance of a solution becoming infeasible in practice. To estimate this, we applied a Monte Carlo simulation with 1000 scenarios in which the uncertain parameters follow a continuous uniform distribution from their nominal value to their maximum realization for a given Dev^d (or Dev^t).

5.3.1 Cardinality-Constrained Uncertainty Set.

Table 5 shows the average results for each budget configuration of the cardinality-constrained set, considering maximum deviations of 10%, 25%, and 50%. For each combination of budgets (Γ^d and Γ^t) and deviations (Dev), the table shows the average price-of-robustness (PoR), which constitutes the percentage increase in costs of the robust solution over the deterministic one, and the average percentage of infeasible scenarios in the Monte Carlo simulation (Risk). Furthermore, regarding the PoR, the table shows the minimum (Best) and maximum (Worst) values obtained among all instances under that combination, and the normalized standard deviation (SD). All values are presented as percentages. The SD values help us evaluate the solutions' variability because, although the average robust solution among the instances might be good, there is a risk of the worst-case performance being considerably more expensive.

We note that, by construction, greater budgets (Γ^d and Γ^t) enlarge the uncertainty sets, and therefore, the probability of infeasibility decreases, but the cost increases accordingly. This can be identified by comparing the PoR and the Risk values between solutions with same deviation but different budgets. The probability of constraint violation, on the other hand, tends to decrease for large uncertainty budgets Γ. It is up to the decision-maker to select the most appropriate level of robustness. Regarding the impact of the deviations (Dev), we observe an increase on the PoR as they increase (positive budgets only), as the robust solutions become more conservative to account for these additional deviations. Furthermore, the deviation strongly impacts the Risk, especially when considering smaller budgets, as the solutions are more likely to violate the capacity or time windows constraints.

Table 5: Average Probability of Constraint Violation (Risk) and Average Price-of-Robustness (PoR) of the Solutions Obtained for the Cardinality-Constrained RVRPTW considering Different Deviations. For Each Combination of Budgets and Deviation, Columns Best, Worst, and SD give the Minimum, Maximum, and Normalized Standard Deviation Regarding PoR. All Values are Presented as Percentages

Γ^d	Γ^t	Dev=10%					Dev=25%					Dev=50%				
		Risk	PoR	Best	Worst	SD	Risk	PoR	Best	Worst	SD	Risk	PoR	Best	Worst	SD
0	0	27.5	0.0	0.0	0.0	10.1	61.2	0.0	0.0	0.0	10.1	66.3	0.0	0.0	0.0	10.1
1	0	23.7	0.1	0.0	0.6	10.0	61.2	0.1	0.0	0.0	5.9	66.2	9.3	5.3	14.6	9.0
5	0	0.0	9.3	5.3	14.6	9.0	0.0	18.3	0.0	0.0	5.9	0.4	22.3	14.4	26.8	7.2
10	0	0.0	14.8	6.7	19.5	7.1	0.0	21.5	0.0	0.0	5.9	0.0	24.5	14.9	30.6	6.3
0	0	17.4	0.0	0.0	0.0	10.1	45.1	21.5	0.0	0.0	10.1	63.9	0.0	0.0	0.0	10.1
0	1	8.0	0.3	0.0	2.7	10.2	15.3	1.2	0.0	0.0	5.9	32.1	4.1	0.7	17.1	13.6
0	5	0.0	0.7	0.0	2.8	10.1	0.0	3.7	0.0	1.2	5.7	0.0	12.2	4.5	27.1	12.6
0	10	0.0	0.7	0.0	2.8	10.0	0.0	4.2	0.0	1.3	5.7	0.0	14.7	7.7	33.8	11.2
0	0	42.8	0.0	0.0	0.0	10.1	86.9	4.2	0.0	0.0	10.1	98.4	0.0	0.0	0.0	10.1
1	1	29.7	0.4	0.0	3.3	10.1	66.8	1.3	0.0	0.7	5.8	80.4	13.0	10.2	20.6	10.0
5	5	0.0	10.4	5.6	17.1	8.7	0.0	20.7	0.0	0.9	5.8	0.4	26.5	18.4	36.3	6.1
10	10	0.0	14.2	7.0	22.0	7.0	0.0	23.9	0.0	1.3	5.7	0.0	27.5	18.5	37.3	6.1
All		12.4	4.2	2.0	7.1	9.4	28.0	10.1	0.0	0.4	6.9	34.0	12.8	7.9	20.3	9.4

Interestingly enough, the SD values of the robust solutions are close to, or even smaller than, those of the deterministic solutions in most instances. Since they are small, there is a relatively small difference between the best and worst PoR within an instance class. From a decision-maker's standpoint, that means that using the RO approach usually results in solutions with similar increments in costs over the deterministic solution, with relatively low chances of large variability. This is particularly good for planning because, once the relevant uncertainty budgets are set, there is little need to test different RO parameters for the new data when working with a given cost target.

Figure 3 shows the plots of the PoR versus the Risk for instances in classes C1, R1, and RC1, considering different budgets of uncertainty (0, 1, 5, and 10) and deviations (0, 10, and 25%). The more convex the curve is, the better the trade-off between PoR and Risk is because the robust solution effectively reduces the risks with a slight cost increase. Thus, some of the solutions with the most expensive trade-off are those from instance class RC1 in Figures 3(c) and 3(f), where an increase of 30% in the costs is needed to neutralize the risks. We also note that in some cases, namely class C1 in Figures 3(a), 3(b), 3(e), and 3(f), class R1 in 3(c), and class RC1 in 3(d),

increasing the budget also increases the solution costs with no impact in the risks. These cases occur when the solution with a smaller budget already had a null risk, and thus using a more conservative choice only increases the cost of the solution. From a decision-maker perspective, there is no reason to use these over-conservative solutions over the ones with zero risk and lower costs. On the other hand, some robust solutions, particularly those with deviation in time, greatly improve the robustness with an excellent trade-off, voiding all risks with a worst-case deviation of 10% with an increase in costs of as little as 1.3%.

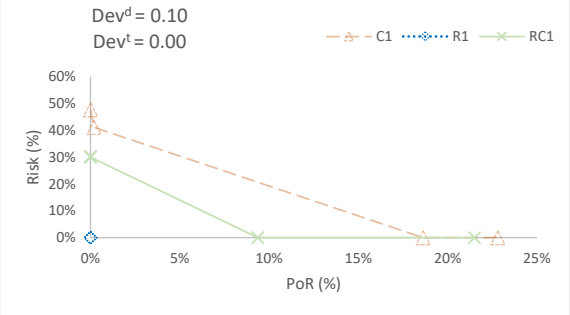
In summary, the results of the cardinality-constrained RVRPTW suggest that the RO approach can provide relevant solutions to support the decision-making process of choosing vehicle routes. The decision-maker can obtain solutions with different trade-offs regarding risks and costs. With this data, they can make an informed choice and opt for a more robust solution, which deteriorates the costs but ensures a better service level, or take a cheaper solution but accept more risks.

5.3.2 Knapsack Uncertainty Set.

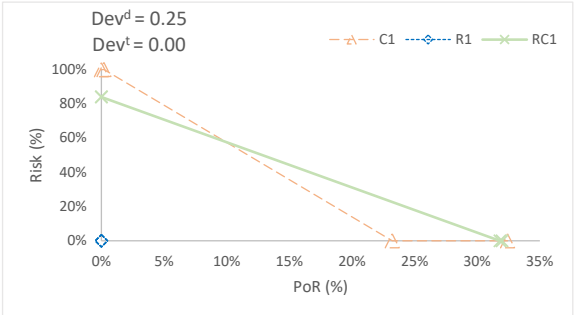
Table 6 summarizes the PoR and Risk values for the single knapsack- constrained RVRPTW according to each combination of budgets (Δ^d and Δ^t) and deviations (Dev). Similarly to the previous uncertainty set, as the budgets Δ^d and Δ^t increase, the solutions become more conservative, and thus the PoR tends to increase while the risks tend to lower. The increase in the deviation values (Dev) also significantly affects the PoR and Risk. For a given combination of Dev and Δ, the robust solutions have a similar increment in the PoR for all instances of the group. This is a positive behavior, as the decision-maker can have prior insight into the cost of the solution for that combination before running a new instance, allowing them to better direct computational efforts.

The results indicate that the solutions obtained for the single knapsack-constrained uncertainty set are similar to those obtained for the cardinality-constrained uncertainty set, when considering the same deviation level (Dev). This is confirmed in the charts presented in Figure 4, which show the average Risk and PoR for each combination of (Dev^d, Dev^t) of the solutions obtained considering the cardinality-constrained and the knapsack uncertainty set. Particularly, the solutions considering the cardinality-constrained set with $\Gamma = 1$ were relatively similar to those considering the single knapsack set

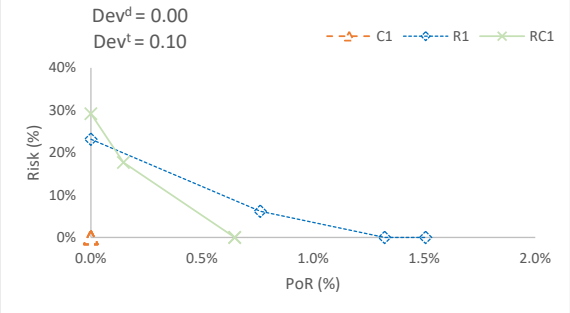
Figure 3: Trade-off Between Price-of-Robustness (PoR) and Probability of Constraint Violation (Risk) for the Instances in Classes C1, R1, and RC1



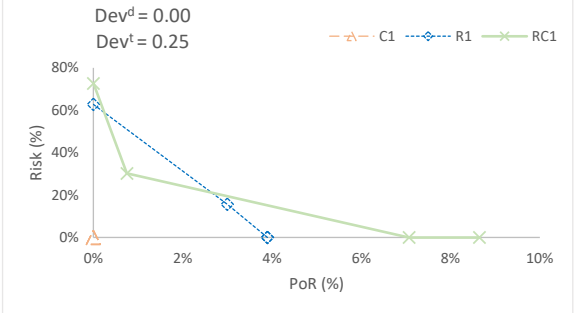
(a) PoR vs Risk for $Dev^d = 10\%$ and $Dev^t = 0\%$



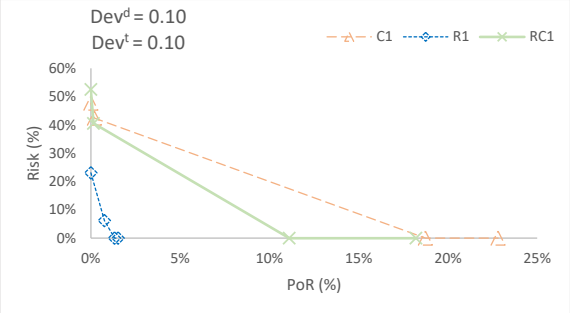
(b) PoR vs Risk for $Dev^d = 25\%$ and $Dev^t = 0\%$



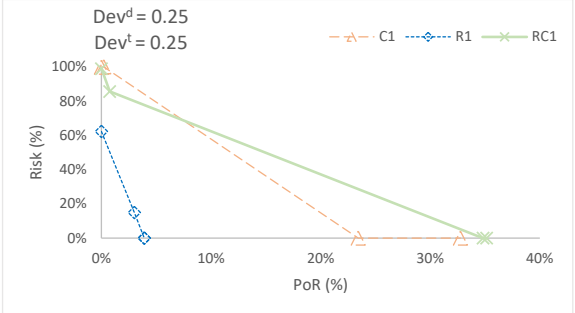
(c) PoR vs Risk for $Dev^d = 0\%$ and $Dev^t = 10\%$



(d) PoR vs Risk for $Dev^d = 0\%$ and $Dev^t = 25\%$



(e) PoR vs Risk for $Dev^d = 10\%$ and $Dev^t = 10\%$



(f) PoR vs Risk for $Dev^d = 25\%$ and $Dev^t = 25\%$

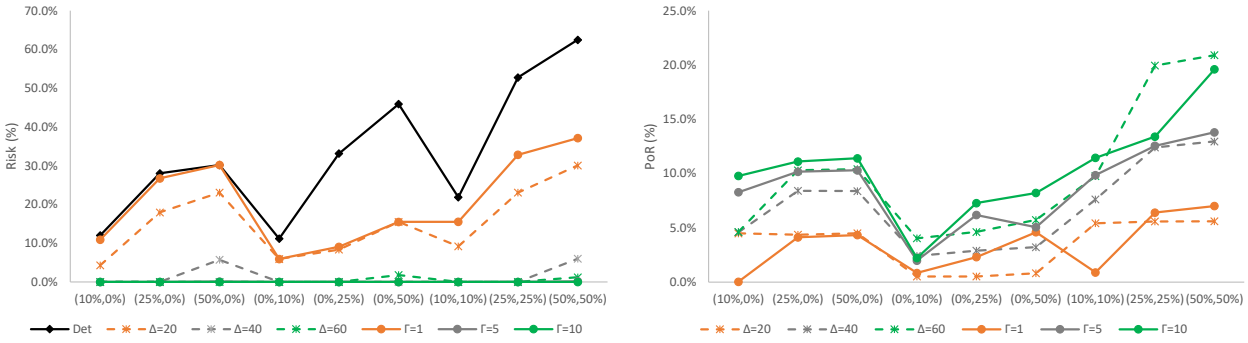
Table 6: Average Probability of Constraint Violation (Risk) and Average Price-of-Robustness (PoR) of the Solutions Obtained for the Single Knapsack-Constrained RVRPTW Considering Different Deviations. For each Combination of Budgets and Deviation, Columns Best, Worst and SD give the Minimum, Maximum and Normalized Standard Deviation Regarding PoR. All Values are Presented as Percentages

Δ^d	Δ^t	Dev=10%					Dev=25%					Dev=50%				
		Risk	PoR	Best	Worst	SD	Risk	PoR	Best	Worst	SD	Risk	PoR	Best	Worst	SD
0	0	27.5	0.0	0.0	0.0	10.1	61.2	0.0	0.0	0.0	10.1	66.3	0.0	0.0	0.0	10.1
20	0	10.0	9.6	5.3	14.7	8.7	50.2	9.4	5.3	14.7	8.9	66.3	9.6	5.3	14.7	8.7
40	0	0.0	9.7	5.3	14.8	8.7	0.0	18.6	10.0	24.1	6.9	21.3	18.5	10.0	23.8	7.1
60	0	0.0	9.8	5.3	15.2	8.6	0.0	22.5	14.3	28.0	6.6	0.0	22.8	14.4	30.2	6.4
0	0	17.4	0.0	0.0	0.0	10.1	45.0	0.0	0.0	0.0	10.1	63.9	0.0	0.0	0.0	10.1
0	20	8.0	0.6	0.0	4.8	9.7	15.3	0.7	0.0	3.3	10.6	32.1	0.9	0.0	4.6	10.5
0	40	0.0	2.2	0.0	10.1	9.2	0.0	2.6	0.0	6.3	9.6	0.0	3.6	0.1	10.5	10.5
0	60	0.0	3.7	0.0	13.4	8.7	0.0	5.1	0.4	16.5	10.0	0.0	6.2	0.4	22.0	11.1
0	0	42.8	0.0	0.0	0.0	10.1	86.9	0.0	0.0	0.0	10.1	98.4	0.0	0.0	0.0	10.1
20	20	15.6	10.6	5.3	18.7	8.3	56.9	11.0	5.3	18.3	9.0	80.6	10.7	5.3	17.9	9.1
40	40	0.0	13.3	5.6	21.1	7.2	0.0	23.7	11.2	35.2	6.6	22.3	24.6	12.5	36.6	7.6
60	60	0.0	15.1	6.4	26.0	6.3	0.0	29.1	14.8	44.9	6.1	4.5	30.9	15.6	49.2	6.8
All		10.1	6.2	2.8	11.6	8.8	26.3	10.2	5.1	15.9	8.7	38.0	10.7	5.3	17.5	9.0

and $\Delta = 20$, although the latter presented slightly lower risks. However, the solutions considering $\Delta = 20$ were relatively cheaper on instances with deviations exclusively on travel times, while retaining the same robustness level, highlighting the advantages of using the knapsack uncertainty set. This behavior is particularly noticeable in instances with deviation exclusively on travel time.

Similarly, the solutions considering $\Delta = 40$ and $\Delta = 60$ outperform those with $\Gamma = 5$ and $\Gamma = 10$ in terms of Risk and PoR in the groups with $Dev = 10\%$ and $Dev = 25\%$. In the instances with $Dev = 50\%$, on the other hand, there is a trade-off between the solutions with $\Gamma = 5$ and $\Gamma = 10$, which have zero risk, and the solutions with $\Delta = 40$ and $\Delta = 60$, which usually had slightly lower costs in exchange for slightly higher risks. These results show that the knapsack uncertainty set might be useful from the decision-maker's point of view, as it can provide different and competitive solutions that may outperform solutions related to the cardinality-constrained uncertainty set in some configurations. Moreover, this uncertainty set is often more intuitive to design as it is easier to estimate the maximum deviation in a route instead of the number of arcs/nodes attaining their worst-case value.

Figure 4: Average Risk and PoR of for Different Deviation Levels (Dev^d,Dev^t) and Uncertainty Sets



(a) Average risks for solutions considering the cardinality constrained (Γ) and knapsack (Δ) for diferent deviation levels

(b) Average price-of-robustness for solutions considering the cardinality constrained (Γ) and knapsack (Δ) for diferent deviation levels

6 Conclusions

We proposed new compact models for the RVRPTW with variability in demand and travel time by considering the cardinality-constrained and knapsack uncertainty sets. We are not aware of any other compact model considering a knapsack uncertainty set. The compact models were formulated using MTZ-based and CF constraints, and we exploited the linearization technique to model uncertainty. Additionally, we proposed tailored branch-and-cut algorithms based on these formulations, resulting in more efficient approaches. Moreover, the CF-based formulation was used for the first time to model the deterministic VRPTW.

Computational results with the proposed compact models using benchmark instances from the literature suggest that the MTZ-based formulation has the best overall performance for both uncertainty sets, even though the CF model exhibits stronger linear relaxations and performs better on the deterministic case. The tailored branch-and-cut algorithms considerably improve the computational results of the proposed models, especially the one that relies on the MTZ-based formulation. When comparing the results of the two uncertainty sets, we observe that the use of the knapsack uncertainty set provides interesting solutions that can present a robustness level similar to the cardinality-constrained set, traditionally used in literature, but with lower costs. Moreover, this uncertainty set may be easier to use in a practical

setting, since it requires only the estimation of the maximum expected deviation, a more realistic requirement than the number of parameters attaining their worst-case value.

There are interesting directions for future work. One would be to develop branch-price-and-cut algorithms for the knapsack-constrained RVRPTW, which would allow us to solve larger instances of the problem. These algorithms may be combined with heuristic approaches, leading to an effective exact hybrid method for this variant. Another interesting topic would be to extend the proposed compact models and branch-and-cut algorithms to other types of uncertainty sets used in the RO literature, such as the ellipsoidal and factor models, which may show more suitable features in a decision-making process.

References

Michel Gendreau, Ola Jabali, and Walter Rei. Future research directions in stochastic vehicle routing (50th anniversary invited article). *Transportation Science*, 50(4):1163–1173, 2016.

John R. Birge and François V. Louveaux. *Introduction to Stochastic Programming*. Springer Verlag, New York, 1997.

Fernando Ordonez. Robust vehicle routing. In John J. Hasenbein, Paul Gray, and Harvey J. Greenberg, editors, *Risk and Optimization in an Uncertain World*, chapter 7, pages 153–178. Informs, Catonsville, MD, 2010.

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

Hamed Rahimian and Sanjay Mehrotra. Frameworks and results in distributionally robust optimization. *Open Journal of Mathematical Optimization*, 3(4):1–85, 2022.

Agostinho Agra, Marielle Christiansen, Rosa Figueiredo, Lars Magnus Hvattum, Michael Poss, and Cristina Requejo. Layered formulation for the robust vehicle routing problem with time windows. In A. Ridha Mahjoub, Vangelis Markakis, Ioannis Milis, and Vangelis Th. Paschos, editors, *Combinatorial Optimization*, pages 249–260, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

Chrysanthos E. Gounaris, Panagiotis P. Repoussis, Christos D. Tarantilis, Wolfram Wiesemann, and Christodoulos A. Floudas. An adaptive memory programming

framework for the robust capacitated vehicle routing problem. *Transportation Science*, 50(4):1239–1260, 2016.

Pedro Munari, Alfredo Moreno, Jonathan De La Vega, Douglas Alem, Jacek Gondzio, and Reinaldo Morabito. The robust vehicle routing problem with time windows: Compact formulation and branch-price-and-cut method. *Transportation Science*, 53(4):1043–1066, 2019.

Anirudh Subramanyam, Panagiotis P. Repoussis, and Chrysanthos E. Gounaris. Robust optimization of a broad class of heterogeneous vehicle routing problems under demand uncertainty. *INFORMS Journal on Computing*, 32(3):661–681, 2020.

Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

Michel Minoux. On robust maximum flow with polyhedral uncertainty sets. *Optimization Letters*, 3(3):367–376, 2009.

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

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

Enrico Bartolini, Dominik Goeke, Michael Schneider, and Mengdie Ye. The robust traveling salesman problem with time windows under knapsack-constrained travel time uncertainty. *Transportation Science*, 55(2):371–394, 2021.

C.E Miller, A.W Tucker, and R.A Zemlin. Integer programming formulation of traveling salesman problems. *Journal of Association for Computing Machinery*, 7(1):326–9, 1960.

Artur Alves Pessoa, Michael Poss, Ruslan Sadykov, and François Vanderbeck. Branch-cut-and-price for the robust capacitated vehicle routing problem with knapsack uncertainty. *Operations Research*, 69(3):739–754, 2021.

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

Qinxiao Yu, Chun Cheng, and Ning Zhu. Robust team orienteering problem with decreasing profits. *INFORMS Journal on Computing*, forthcoming, 2022.

Luis Gouveia. A result on projection for the vehicle routing problem. *European Journal of Operational Research*, 85(3):610–624, 1995.

Adam N Letchford and Juan-José Salazar-González. Projection results for vehicle routing. *Mathematical Programming*, 105(2):251–274, 2006.

Adam N Letchford and Juan-José Salazar-González. Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. *European Journal of Operational Research*, 244(3):730–738, 2015.

André Langevin, Martin Desrochers, Jacques Desrosiers, Sylvie Gelinas, and François Soumis. A two-commodity flow formulation for the traveling salesman and the makespan problems with time windows. *Networks*, 23(7):631–640, 1993.

Nicola Bianchessi and Stefan Irnich. Branch-and-cut for the split delivery vehicle routing problem with time windows. *Transportation Science*, 53(2):442–462, 2019.

B Gavish. The delivery problem: New cutting planes procedures. In *TIMS XXVI Conference, Copenhagen*, 1984.

Jens Lysgaard, Adam Letchford, and Richard Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100:423–445, 06 2004.

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

Agostinho Agra, Marielle Christiansen, Rosa Figueiredo, Lars Magnus Hvattum, Michael Poss, and Cristina Requejo. The robust vehicle routing problem with time windows. *Computers & Operations Research*, 40(3):856–866, 2013.

A MTZ-Based Formulation for the RVRPTW Under a Multiple Knapsack Uncertainty Set

To adapt the previous MTZ-based model to the multiple knapsack uncertainty set, we need new load and time variables and budget parameters. Each knapsack has a budget Δ_l^d (demand) or Δ_l^t (time). We provide the model for two knapsacks, and the extension to more knapsacks is trivial. The variables are now $u_{i\delta_1\delta_2}$ to represent accumulated load of the vehicle up to node i, with a total deviation δ_1 over the demand's nominal value for the first knapsack and δ_2 for the second one; likewise, timing variables are now $w_{i\delta_1\delta_2}$ representing the earliest possible time to start the service at node i, considering a total deviation of δ_1 over the travel time's nominal value for the first knapsack and δ_2 for the second one. The model is then:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij}, \tag{45}$$

s.t. (6), (7), (12), and to

$$u_{j\delta_1\delta_2} \geq u_{i\delta_1\delta_2} + \bar{d}_j + Q(x_{ij}-1), \quad (i,j)\in A, 0\leq\delta_1\leq\Delta_1^d, 0\leq\delta_2\leq\Delta_2^d, \tag{46}$$

$$u_{j\delta_1\delta_2} \geq u_{i\delta_1-\hat{d}_j\delta_2} + \bar{d}_j + \hat{d}_j + Q(x_{ij}-1), \quad (i,j)\in A, j\in\mathcal{S}_1, j\notin\mathcal{S}_2, \hat{d}_j\leq\delta_1\leq\Delta_1^d, 0\leq\delta_2\leq\Delta_2^d, \tag{47}$$

$$u_{j\delta_1\delta_2} \geq u_{i\delta_1\delta_2-\hat{d}_j} + \bar{d}_j + \hat{d}_j + Q(x_{ij}-1), \quad (i,j)\in A, j\notin\mathcal{S}_1, j\in\mathcal{S}_2, 0\leq\delta_1\leq\Delta_1^d, \hat{d}_j\leq\delta_2\leq\Delta_2^d, \tag{48}$$

$$u_{j\delta_1\delta_2} \geq u_{i\delta_1-\hat{d}_j\delta_2-\hat{d}_j} + \bar{d}_j + \hat{d}_j + Q(x_{ij}-1), \quad (i,j)\in A, j\in\mathcal{S}_1, j\in\mathcal{S}_2, \hat{d}_j\leq\delta_1\leq\Delta_1^d, \hat{d}_j\leq\delta_2\leq\Delta_2^d, \tag{49}$$

$$u_{j\Delta_1^d\delta_2} \geq u_{i\Delta_1^d-\lambda\delta_2} + \bar{d}_j + \lambda + Q(x_{ij}-1), \quad (i,j)\in A, 0\leq\lambda\leq\hat{d}_j, 0\leq\delta_2\leq\Delta_2^d, j\in\mathcal{S}_1, j\notin\mathcal{S}_2, \tag{50}$$

$$u_{j\delta_1\Delta_2^d} \geq u_{i\delta_1\Delta_2^d-\lambda} + \bar{d}_j + \lambda + Q(x_{ij}-1), \quad (i,j)\in A, 0\leq\lambda\leq\hat{d}_j, 0\leq\delta_1\leq\Delta_1^d, j\notin\mathcal{S}_1, j\in\mathcal{S}_2, \tag{51}$$

$$u_{j\delta_1\Delta_2^d} \geq u_{i\delta_1-\lambda\Delta_2^d-\lambda} + \bar{d}_j + \lambda + Q(x_{ij}-1), \quad (i,j)\in A, \lambda\leq\hat{d}_j, j\in\mathcal{S}_1, j\in\mathcal{S}_2, \tag{52}$$

$$u_{j\Delta_1^d\Delta_2^d} \leq Q, \quad j\in N, \tag{53}$$

$$w_{j\delta_1\delta_2} \geq w_{i\delta_1-\lambda\delta_2} + \bar{t}_{ij} + s_i + \lambda + b_{n+1}(x_{ij}-1),$$

$$(i,j) \in A, 0 \leq \lambda \leq \min\{\hat{t}_{ij}, \delta_1\} \leq \Delta_1^t, 0 \leq \delta_2 \leq \Delta_2^t, j \in \mathcal{S}_1, j \notin \mathcal{S}_2, \quad (54)$$

$$w_{j\delta_1\delta_2} \geq w_{i\delta_1\delta_2-\lambda} + \bar{t}_{ij} + s_i + \lambda + b_{n+1}(x_{ij} - 1),$$

$$(i,j) \in A, 0 \leq \delta_1 \leq \Delta_1^t, \lambda \leq \min\{\hat{t}_{ij}, \delta_2\} \leq \Delta_2^t, j \notin \mathcal{S}_1, j \in \mathcal{S}_2, \quad (55)$$

$$w_{j\delta_1\delta_2} \geq w_{i\delta_1-\lambda\delta_2-\lambda} + \bar{t}_{ij} + s_i + \lambda + b_{n+1}(x_{ij} - 1),$$

$$(i,j) \in A, 0 \leq \lambda \leq \hat{t}_{ij}, \lambda \leq \delta_1 \leq \Delta_1^t, \lambda \leq \delta_2 \leq \Delta_2^t, j \in \mathcal{S}_1, j \in \mathcal{S}_2, \quad (56)$$

$$a_j \leq w_{j\delta_1\delta_2} \leq b_j, \qquad (i,j) \in A, 0 \leq \delta_1 \leq \Delta_1^t, 0 \leq \delta_2 \leq \Delta_2^t, \quad (57)$$

$$u_{i\delta_1\delta_2} \geq 0, \qquad i \in N, 0 \leq \delta_1 \leq \Delta_1^d, 0 \leq \delta_2 \leq \Delta_2^d, \quad (58)$$

$$w_{i\delta_1\delta_2} \geq 0, \qquad i \in N, 0 \leq \delta_1 \leq \Delta_1^t, 0 \leq \delta_2 \leq \Delta_2^t. \quad (59)$$

Similarly to the previous models, the objective function (45) seeks to minimize the total traveling costs. Constraints (46)–(52) compute the demand for the worst case and forbid subtours. The vehicle capacity is ensured by constraints (53). Constraints (54)–(56) are similar to (50)–(52), but for the travel time. The time windows constraints are imposed by (57) and the domains of the variables are defined in (58)–(59). Note that we can easily extend this model for k knapsacks by adding indices from δ_1 to δ_k.

B Commodity-Flow Formulation for the RVRPTW Under a Multiple Knapsack Uncertainty Set

Similarly to how it was done for the MTZ-based model, we extend formulation (32)–(38) to consider multiple knapsacks. Again, we present a formulation containing two knapsacks as the extension to k knapsacks is trivial, but the number of constraints grows quickly. This formulation uses the same sets and parameters as the MTZ-based model, while the load and time variables are now $f_{ij\delta_1\delta_2}$ and $g_{ij\delta_1\delta_2}$ with similar interpretations. The resulting model is given by:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij}, \quad (60)$$

s.t. (6), (7), (12), and to

$$\sum_{j:(i,j)\in A} f_{ij\delta_1\delta_2} \geq \bar{d}_i + \sum_{h:(h,i)\in A} f_{hi\delta_1\delta_2}, \qquad i \in N, 0 \leq \delta_1 \leq \Delta_1^d, 0 \leq \delta_2 \leq \Delta_2^d, \quad (61)$$

$$\sum_{j:(i,j)\in A} f_{ij\delta_1\delta_2} \geq \bar{d}_i + \hat{d}_i + \sum_{h:(h,i)\in A} f_{hi\delta_1-\hat{d}_i\delta_2},$$

$$i \in N, i \in \mathcal{S}_1, i \notin \mathcal{S}_2, \hat{d}_i \leq \delta_1 \leq \Delta_1^d, 0 \leq \delta_2 \leq \Delta_2^d, \quad (62)$$

$$\sum_{j:(i,j)\in A} f_{ij\delta_1\delta_2} \geq \bar{d}_i + \hat{d}_i + \sum_{h:(h,i)\in A} f_{hi\delta_1\delta_2-\hat{d}_i},$$

$$i \in N, i \notin \mathcal{S}_1, i \in \mathcal{S}_2, 0 \leq \delta_1 \leq \Delta_1^d, \hat{d}_i \leq \delta_2 \leq \Delta_2^d, \quad (63)$$

$$\sum_{j:(i,j)\in A} f_{ij\delta_1\delta_2} \geq \bar{d}_i + \hat{d}_i + \sum_{h:(h,i)\in A} f_{hi\delta_1-\hat{d}_i\delta_2-\hat{d}_i},$$

$$i \in N, i \in \mathcal{S}_1, i \in \mathcal{S}_2, 0 \leq \delta_1 \leq \Delta_1^d, \hat{d}_i \leq \delta_2 \leq \Delta_2^d, \quad (64)$$

$$\sum_{j:(i,j)\in A} f_{ij\Delta_1^d\delta_2} \geq \bar{d}_i + \lambda + \sum_{h:(h,i)\in A} f_{hi\Delta_1^d-\lambda\delta_2},$$

$$i \in N, i \in \mathcal{S}_1, i \notin \mathcal{S}_2, 0 \leq \delta_2 \leq \Delta_2^d, 0 \leq \lambda < \hat{d}_i, \quad (65)$$

$$\sum_{j:(i,j)\in A} f_{ij\delta 1\Delta_2^d} \geq \bar{d}_i + \lambda + \sum_{h:(h,i)\in A} f_{hi\delta_1\Delta_2^d-\lambda},$$

$$i \in N, i \notin \mathcal{S}_1, i \in \mathcal{S}_2, 0 \leq \delta_1 \leq \Delta_1^d, 0 \leq \lambda < \hat{d}_i, \quad (66)$$

$$\sum_{j:(i,j)\in A} f_{ij\Delta_1^d\Delta_2^d} \geq \bar{d}_i + \lambda + \sum_{h:(h,i)\in A} f_{hi\Delta_1^d-\lambda\Delta_2^d-\lambda},$$

$$i \in N, i \in \mathcal{S}_1, i \in \mathcal{S}_2, 0 \leq \lambda < \hat{d}_i, \quad (67)$$

$$d_i x_{ij} \leq f_{ij\delta_1\delta_2} \leq (Q - d_j)x_{ij}, \qquad (i,j) \in A, 0 \leq \delta_1 \leq \Delta_1^d, 0 \leq \delta_2 \leq \Delta_2^d, \quad (68)$$

$$\sum_{j:(i,j)\in A} g_{ij\delta_1\delta_2} \geq \sum_{\substack{h:(h,i)\in A \\ \lambda \leq \hat{t}_{hi}}} (g_{hi\delta_1-\lambda\delta_2} + (\bar{t}_{hi} + \lambda)x_{hi}) + s_i,$$

$$i \in N, i \in \mathcal{S}_1, i \notin \mathcal{S}_2, 0 \leq \lambda \leq \delta_1 \leq \Delta_1^t, 0 \leq \delta_2 \leq \Delta_2^t, \quad (69)$$

$$\sum_{j:(i,j)\in A} g_{ij\delta_1\delta_2} \geq \sum_{\substack{h:(h,i)\in A \\ \lambda \leq \hat{t}_{hi}}} (g_{hi\delta_1\delta_2-\lambda} + (\bar{t}_{hi} + \lambda)x_{hi}) + s_i,$$

$$i \in N, i \notin \mathcal{S}_1, i \in \mathcal{S}_2, 0 \leq \delta_1 \leq \Delta_1^t, 0 \leq \lambda \leq \delta_2 \leq \Delta_2^t, \quad (70)$$

$$\sum_{j:(i,j)\in A} g_{ij\delta_1\delta_2} \geq \sum_{\substack{h:(h,i)\in A \\ \lambda \leq \hat{t}_{hi}}} (g_{hi\delta_1-\lambda\delta_2-\lambda} + (\bar{t}_{hi} + \lambda)x_{hi}) + s_i,$$

$$i \in N, i \in \mathcal{S}_1, i \in \mathcal{S}_2, 0 \leq \lambda \leq \delta_1 \leq \Delta_1^t, 0 \leq \lambda \leq \delta_2 \leq \Delta_2^t, \quad (71)$$

$$(a_j + s_j)x_{ij} \leq g_{ij\delta_1\delta_2} \leq (b_j + s_j)x_{ij}, \qquad (i,j) \in A, 0 \leq \delta_1 \leq \Delta_1^t, 0 \leq \delta_2 \leq \Delta_2^t \quad (72)$$

$$x_{ij} \in \{0,1\}, \qquad (i,j) \in A, \quad (73)$$

$$f_{ij\delta_1\delta_2} \geq 0, \qquad (i,j) \in A, 0 \leq \delta_1 \leq \Delta_1^d, 0 \leq \delta_2 \leq \Delta_2^d, \quad (74)$$

$$g_{ij\delta_1\delta_2} \geq 0, \qquad (i,j) \in A, 0 \leq \delta_1 \leq \Delta_1^t, 0 \leq \delta_2 \leq \Delta_2^t. \quad (75)$$

The interpretation of this model is very similar to the previous one, but the variables are related to the arcs and not the nodes.