



New formulations for the robust vehicle routing problem with time windows under demand and travel time uncertainty

Rafael Campos^{1,3} · Leandro C. Coelho^{1,2} · Pedro Munari³

Received: 18 September 2023 / Accepted: 4 July 2024

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

We present new formulations for the robust vehicle routing problem with time windows (RVRPTW) under cardinality- and knapsack-constrained demand and travel time uncertainty. They are the first compact models to address the RVRPTW under travel time uncertainty while considering the knapsack uncertainty set. Moreover, our models employ different types of constraints to control time propagation based on Miller–Tucker–Zemlin and single commodity flow constraints, which are derived from the linearization of recursive equations. We develop branch-and-cut methods based on the proposed formulations, leveraging a dynamic programming algorithm to verify the robust feasibility of solutions concerning both demand and travel time uncertainty, in addition to specific and standard separation procedures from the literature. We present detailed computational results on RVRPTW benchmark instances to compare the performance of our models and algorithms. Furthermore, we evaluate the impact and advantages of implementing each studied uncertainty set.

Keywords Vehicle routing problem · Time windows · Uncertainty · Robust optimization · Commodity flow · Knapsack uncertainty set

✉ Pedro Munari
munari@dep.ufscar.br

¹ Faculté des Sciences de l'Administration, Université Laval, Québec, Canada

² Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT), Québec, Canada

³ Department of Production Engineering, Federal University of São Carlos (UFSCar), São Carlos, São Paulo, Brazil

1 Introduction

In distribution problems, it is often assumed that all parameters are known and deterministic when solving a vehicle routing problem (VRP). In reality, however, these problems are associated with significant uncertainties, such as on travel times or on each customer's demand (Boysen et al. 2021; Gendreau et al. 2014). If their probability distributions are known, one can resort to stochastic optimization (De La Vega et al. 2023; Hoogendoorn and Spliet 2023). When these are unknown, or one wants to ensure that a feasible solution is guaranteed for all scenarios, robust optimization (RO) is often more appropriate (Ordóñez 2010; Munari et al. 2019; Caunhye and Alem 2023).

Moreover, to avoid overly conservative solutions with excessive costs due to overprotection against uncertainty, different *uncertainty sets* can model the realization of uncertain parameters whose worst-case behavior can be controlled by the decision maker. The definition of an appropriate uncertainty set is a critical step when using RO, as different sets often result in solutions with different costs and robustness levels (Subramanyam et al. 2020; Bartolini et al. 2021). From a decision maker's perspective, 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, in which the decision maker limits the number of parameters that simultaneously attain their worst case value, and the *knapsack* uncertainty set, in which the total deviation of a set of parameters is limited instead (Bertsimas and Sim 2004; Minoux 2009; Gounaris et al. 2013).

In the VRP literature, the use of different uncertainty sets has been little explored in the context of travel time uncertainty, as it appears in the robust VRP with time windows (RVRPTW). While there are studies on different uncertainty sets for the VRP under uncertain demand (Gounaris et al. 2013; Subramanyam et al. 2020; Wang et al. 2021), the vast majority of RVRPTW studies under travel time uncertainty have considered only the cardinality-constrained uncertainty set (Agra et al. 2012; Lee et al. 2012; Munari et al. 2019; De La Vega et al. 2020). Wang et al. (2021) addressed the RVRPTW considering multiple uncertainty sets, but representing travel time deviations using only the discrete and the cardinality-constrained sets.

Bartolini et al. (2021) were the first to consider travel time uncertainty with a knapsack uncertainty set, but for the robust traveling salesman problem with time windows (i.e., considering a single vehicle). Hence, to the best of our knowledge, we are the first to address the RVRPTW under demand and travel time uncertainty using a knapsack uncertainty set. This set provides more flexibility in modeling uncertain parameters compared to the cardinality-constrained uncertainty set by limiting total deviation considering a subset of uncertain parameters, instead of assuming a fixed number of parameters attaining their worst case. In the VRP context, for example, it is often more natural to estimate how late a route usually is, rather than specifying how many roads or streets will experience their worst-case traffic delays. Moreover, as pointed out by Bartolini et al. (2021), a knapsack

uncertainty set can be parameterized to yield richer tradeoffs, in addition to providing more precise insights into the resilience to delays.

The main contributions of this paper are as follows:

- We introduce new compact formulations for the RVRPTW under demand and travel time uncertainty, using both cardinality-constrained and knapsack uncertainty sets. Notably, we consider single and multiple knapsack sets, and our formulation for the multiple knapsack uncertainty set does not require subsets to be disjoint, which distinguishes it from previous approaches. We are not aware of any paper that considers the knapsack uncertainty set for uncertain travel times in the RVRPTW;
- We show how to derive robust counterparts based on the well-known Miller–Tucker–Zemlin (MTZ) constraints (Miller et al. 1960) and single commodity flow (CF) constraints for both the cardinality-constrained and knapsack uncertainty sets. This involves the linearization of recursive equations that calculate the worst-case realizations of the uncertain parameters. Previous applications of the linearization technique have focused only on MTZ-based formulations and the cardinality-constrained set;
- We design tailored branch-and-cut (BC) algorithms based on the proposed formulations. These algorithms use a polynomial-time dynamic programming (DP) algorithm to verify the robust feasibility of integer solutions, for both the cardinality-constrained and knapsack uncertainty sets.

We perform detailed computational analyses on RVRPTW instances and show that, although harder to solve, instances considering the knapsack uncertainty set can provide solutions with a 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 to the compact models.

The remainder of this paper is organized as follows. In Sect. 2, we introduce some definitions and concepts, while in Sect. 3, we present a compact formulation based on CF constraints for the deterministic VRPTW and derive robust counterparts for demand and travel time uncertainty, considering the cardinality-constrained and knapsack uncertainty sets. In Sect. 4, we propose tailored BC algorithms. In Sect. 5, we present the results of computational experiments with benchmark instances. Finally, the concluding remarks are discussed in Sect. 6.

2 Background and literature review

In this section, we start with some background definitions and in Sect. 2.1 we describe the two uncertainty sets used in this paper. Section 2.2 reviews the existing compact formulations developed for the RVRPTW.

The notation used is as follows. Let $C = \{1, \dots, n\}$ 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 $d_i \leq Q$, and we assume $d_0 = d_{n+1} = 0$.

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 $i \in C$, respectively; s_i is the service time of node $i \in C$; and t_{ij} is the travel time of arc $(i, j) \in A$. For the sake of notation, we set $s_0 = s_{n+1} = 0$ and define time windows $[a_0, b_0] = [a_{n+1}, b_{n+1}]$ representing the earliest and latest times a vehicle can depart from and return to the depot. To account for these new characteristics, in compact models one often resorts to the use of constraints inspired by the MTZ formulation.

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 (Bertsimas and Sim 2004). Particularly, if Γ is integer, we can interpret it as the number of worst-case realizations that can simultaneously occur in a route. A solution is considered robust feasible for a budget Γ if it is feasible when up to any Γ uncertain parameters assume their worst-case value simultaneously. For the sake of clarity, we differentiate the budget associated with the demand and the one associated with the travel time using parameters Γ^d and Γ^t , respectively. For example, under demand uncertainty, this means the solution does not violate the vehicle capacity when the demands of any Γ^d 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^t 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:

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

Note that the worst-case values 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. Although a realization of an uncertain parameter could theoretically be lower than the nominal value, from the perspective of RO optimization, negative deviations can always be disregarded. This is because the worst-case scenario of an uncertainty set will never utilize negative deviations on parameters, as this would consume part of the budget and yield a lower total deviation than if no deviation were considered. This is consistent with other works in RVRP literature (Agra et al. 2012; Munari et al. 2019; Subramanyam et al. 2020). The same assumption also applies to the knapsack uncertainty set, which will be further discussed in the next topic.

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. 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 geographic regions (Gounaris et al. 2013; Subramanyam et al. 2020; Pessoa et al. 2021). 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. A quadrant with higher variability in demand may have a larger budget than others. Additionally, one usually builds the multiple knapsacks as disjoint sets, meaning that their nodes/arcs do not overlap.

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

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

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^d as the set of nodes in each knapsack $l \in L^d$:

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

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 drivers to estimate how late they are when traveling to a specific region than to tell how many streets or roads 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., $|L^t| = 1$). 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. As mentioned before, we are not aware of any paper considering the knapsack uncertainty set for uncertain travel times in the RVRPTW.

2.2 Compact formulations for RVRPs

To the best of our knowledge, Sungur et al. (2008) were the first to use RO to incorporate uncertainty into VRPs. Specifically, they proposed compact models based on convex hull, box, and ellipsoidal uncertainty sets for the capacitated VRP (CVRP) under demand uncertainty. These, however, simply replaced the nominal demand of the deterministic formulation with an augmented modified demand, resulting in overly conservative solutions when compared to stochastic programming approaches.

To avoid this overconservatism, some resorted to the cardinality-constrained set introduced by Bertsimas and Sim (2003), using the dualization scheme to derive robust counterparts. This scheme consists of 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.

Agra et al. (2012) used the dualization scheme based on the so-called layered formulation to derive a robust counterpart for the VRPTW under travel time uncertainty. 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 window 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. 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 observed that the layered 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 s.

The same problem was later revisited by Agra et al. (2013), who proposed two more efficient robust approaches, albeit not compact, capable of solving instances with up

to 50 customers. The first extends the so-called resource inequalities formulation by employing adjustable robust optimization techniques, whereas the second approach implicitly considers uncertainty in a path inequalities formulation. Both strategies presented similar computational performance and were more efficient than the layered formulation of Agra et al. (2012).

Gounaris et al. (2013) used the dualization scheme to obtain robust counterparts of several traditional formulations of the CVRP. The proposed modeling approach was able to introduce any convex uncertainty set into the problem. Additionally, they also introduced a robust extension for the traditional *rounded capacity inequalities* (RCI) widely used in the CVRP literature. To assess their method, the authors ran experiments with a commercial solver, in which the robust RCI were introduced globally through user-defined callbacks, considering the knapsack and the factor models uncertainty sets. The most efficient formulations were able to solve most benchmark instances with fewer than 50 customers, but struggled on larger instances.

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 considers the cardinality-constrained uncertainty set only 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, consider the budget Γ^t for travel time uncertainty and a given route $r = (v_0, v_1, \dots, v_h)$. Let $w_{v_j\gamma}$ be a variable representing the worst case for service start time at node v_j , when the travel times at any $\gamma \leq \Gamma^t$ arcs up to this node 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. Recall that $\bar{t}_{v_{j-1}v_j}$ is the nominal travel time on arc (v_{j-1}, v_j) ; $\hat{t}_{v_{j-1}v_j}$ represents the corresponding maximum deviation; and s_{v_j} is the service time at node v_j . Then, the value of $w_{v_j\gamma}$ can be computed using the following recursive equation (Agra et al. 2013; Munari et al. 2019):

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

for all $j = 0, \dots, h$ and $\gamma = 0, \dots, \Gamma^t$. 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:

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

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

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

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 $\gamma = 0, \dots, \Gamma^t$, 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 constraints (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 develop novel RO models for the RVRPTW that use the linearization technique of recursive equations. First, we present in Sect. 3.1 a 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 Sect. 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 Sect. 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; Munari and Savelsbergh 2022), 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:

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

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

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

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

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

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

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

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

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 variable $f_{ij\gamma}$ represent the load carried

on arc $(i, j) \in A$ considering that γ customers attained their worst-case demand; and $g_{ij\gamma}$ represent the earliest elapsed time of the route that traverses arc $(i, j) \in A$, after serving node i , considering that γ arcs attained their worst-case travel time 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:

$$\sum_{j:(i,j) \in A} f_{ij\gamma} \geq \bar{d}_i + \sum_{h:(h,i) \in A} f_{hi\gamma}, \quad i \in C, \gamma = 0, \dots, \Gamma^d, \tag{13}$$

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

$$\bar{d}_i x_{ij} \leq f_{ij\gamma} \leq (Q - \bar{d}_i) x_{ij}, \quad (i, j) \in A, \gamma = 0, \dots, \Gamma^d. \tag{15}$$

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:

$$\sum_{j:(i,j) \in A} g_{ij\gamma} \geq s_i + \sum_{h:(h,i) \in A} (g_{hi\gamma} + \bar{t}_{hi} x_{hi}), \quad i \in C, \gamma = 0, \dots, \Gamma^t, \tag{16}$$

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

$$(s_i + a_i) x_{ij} \leq g_{ij\gamma} \leq (b_i + s_i) x_{ij}, \quad (i, j) \in A, \gamma = 0, \dots, \Gamma^t. \tag{18}$$

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 $x_{i_j j_1} = 1$ and $x_{j_1 k_1} = 1$. Then, time constraints (16) and (17) related to node j_1 , for any $\gamma = 1, \dots, \Gamma^t$, can be simply represented as follows:

$$g_{j_1 k_1 \gamma} \geq g_{i_1 j_1 \gamma} + \bar{t}_{i_1 j_1} + s_{j_1}, \tag{19}$$

$$g_{j_1 k_1 \gamma} \geq g_{i_1 j_1 (\gamma-1)} + \bar{t}_{i_1 j_1} + \hat{t}_{i_1 j_1} + s_{j_1}. \tag{20}$$

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:

$$w_{j_1 \gamma} \geq w_{i_1 \gamma} + \bar{t}_{i_1 j_1} + s_{j_1}, \tag{21}$$

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

where $w_{j_1 \gamma}$ is a continuous variable that represents the departure time from node j_1 considering that up to γ nodes attained their worst-case value. Given that variables $g_{j_1 k_1 \gamma}$ 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 the computational results presented in Sect. 5 indicate, our new formulation yields a much tighter linear programming (LP) 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 Sect. 2.1.2. Similarly, we have Δ_l^d and Δ_l^t 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 $\delta \in \{0, 1, \dots, \Delta^d\}$ 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 $\delta \in \{0, 1, \dots, \Delta^t\}$ 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 Sect. 2.2 for the cardinality-constrained uncertainty set. Let $r = (v_0, v_1, \dots, v_h)$ be a route visiting $h - 1$ customers. For this route, we can compute the values of $w_{v_j\delta}$ for $j = 0, \dots, h$ and $\delta = 0, 1, \dots, \Delta^t$ as follows:

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

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 $\hat{\lambda}_{v_{j-1}v_j}$. 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 $w_{v_j\delta} \leq b_{v_j}$ for each node v_j and $\delta = 0, 1, \dots, \Delta^t$. 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:

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

for $j = 0, \dots, h$ and $\delta = 0, 1, \dots, \Delta^d$. 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}_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:

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

$$\begin{aligned} &\text{s.t. (6), (7), (12), and to} \\ &u_{j\delta} \geq u_{i\delta} + \bar{d}_j + Q(x_{ij} - 1), \quad (i,j) \in A, \delta = 0, \dots, \Delta^d, \end{aligned} \tag{26}$$

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

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

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

$$\begin{aligned} &w_{j\delta} \geq w_{i(\delta-\lambda)} + \bar{t}_{ij} + \lambda + s_i + M_{ij}(x_{ij} - 1), \\ &(i,j) \in A, \delta = 0, \dots, \Delta^t, \lambda = 0, \dots, \min\{\delta, \hat{t}_{ij}\}, \end{aligned} \tag{30}$$

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

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 $\delta \in \{0, 1, \dots, \Delta^d\}$ 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 $\delta \in \{0, 1, \dots, \Delta^t\}$ 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

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

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

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

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

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

$$\lambda \leq \hat{t}_{hi}$$

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

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 min, one can use it in units of 5 min. 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. Nonetheless, 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 . An interesting aspect of our formulation is that, unlike previous studies (Gounaris et al. 2013; Subramanyam et al. 2020), it does not require the knapsacks to be disjointed (i.e., non-overlapping). Thus, without sacrificing formulation correctness, we can model problems where nodes/arcs belong to several knapsacks simultaneously. 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 CVRPS_{SEP} 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 :

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

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, CVRPS_{SEP} 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:

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

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 $Q \in \mathbb{R}^{|V_S|}$, $Q = (\hat{d}_{[1]}, \dots, \hat{d}_{[|V_S|]})$, 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 j th position of this vector. We then compute D_{V_S} , the maximum demand of this set, using

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

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:

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

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:

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

Since we use a heuristic algorithm to generate subset V_S , this separation procedure 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 $r = (v_0, v_1, \dots, v_h)$ the infeasibility is detected in node v_j for a given $j \in \{1, \dots, h\}$ the following inequality is inserted into the problem:

$$\sum_{0 < i \leq j} x_{v_{i-1}v_i} \leq j - 1. \quad (44)$$

We use a different approach for time uncertainty. For the cardinality-constrained and single knapsack uncertainty sets 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. For the two-knapsack uncertainty set, preliminary results showed that the introduction of the robust constraints slowed down the model significantly, as a considerable number of constraints are usually introduced into the model on each cut, and thus we decided to use feasibility cuts (44) instead. This proved to be more effective for this particular uncertainty set. To efficiently check the feasibility of a solution, we use DP algorithms based on equations (1) and (23) 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 knapsack uncertainty sets, using benchmark instances from the literature. We consider two versions of the knapsack uncertainty set,

one with a single knapsack and the other with two knapsacks. 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. The computational experiments were conducted on a computer cluster with 2 x AMD Rome 7532 @ 2.40 GHz processor and 64GB of RAM. We set a time limit of 3600 s 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 in terms of 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 to the first decimal 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 and two-knapsack uncertainty sets, with budgets $\Delta^d, \Delta^t \in \{0, 20, 40, 60\}$. For the two-knapsack uncertainty set we assume that $\Delta^d = \Delta_1^d = \Delta_2^d$ and $\Delta^t = \Delta_1^t = \Delta_2^t$ for any budget. The nodes were divided into two groups, namely *north* and *south*, based on their geographical position in relation to the depot. 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 obtained with the MTZ-based and the CF formulations for the cardinality-constrained RVRPTW. We performed two experiments with both models: solving their LP relaxations only and solving them using the general-purpose MIP solver. For the first experiment, the table presents the average objective value (Obj), the quality of the LP relaxation as a percentage (QLR), and the average

Table 1 Results of the MTZ-based and CF formulations for the cardinality-constrained RVRPTW, considering different configurations of budgets

Γ^d	Γ^r	MIP model														
		LP relaxation					MIP model									
		MTZ		CF			MTZ		CF							
		Obj	QLR	T	Obj	QLR	T	Obj	Gap	T	Obj	Opt	Gap	T	Obj	Opt
0	0	224.93	67.5	0.01	267.43	81.1	0.04	331.27	3.2	536.16	331.27	49	0.6	235.29	331.27	53
1	0	224.94	66.7	0.02	267.68	80.5	0.06	334.63	3.9	576.90	335.23	146	0.6	315.68	335.23	156
5	0	225.00	64.0	0.03	268.60	76.2	0.39	350.73	6.4	843.72	350.72	135	2.6	912.80	350.72	137
10	0	225.03	63.7	0.05	270.19	76.2	0.66	352.63	7.0	938.76	352.71	128	3.9	1296.98	352.71	118
0	1	221.78	66.6	0.03	264.72	80.5	0.09	330.92	3.2	477.54	330.85	150	0.7	289.91	330.85	156
0	5	221.99	65.8	0.04	265.89	79.5	0.44	336.98	3.4	603.18	335.06	145	1.1	556.89	335.06	153
0	10	222.06	65.7	0.07	266.70	79.5	0.99	338.51	4.5	681.15	336.06	140	1.7	921.18	336.06	141
1	1	221.78	65.9	0.02	264.97	79.3	0.12	334.29	3.8	527.33	333.98	148	1.1	369.61	333.98	153
5	5	222.05	63.0	0.06	266.96	75.2	0.74	352.82	5.8	816.57	352.38	136	3.6	1223.87	352.38	125
10	10	222.14	62.7	0.10	269.21	75.1	3.50	354.42	7.2	947.98	355.52	129	5.3	1786.48	355.52	101
All		223.17	65.1	0.04	267.24	78.3	0.70	341.72	4.8	694.93	341.38	1306	2.1	790.87	341.38	1293

computation 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 computation times, the table shows the average optimality gap as a percentage (Gap), as provided by the solver at the end of the execution, and the number of instances 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 was considerably faster with the MTZ model, whereas the CF model resulted in stronger LP bounds. On average, the value of the LP relaxation for the CF model is 78.3% of the integer solution against 65.1% 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 MIP formulations, while the MTZ model yielded shorter run times and 13 more instances solved to optimality, the CF presented lower average optimality gaps in all combinations of Γ^d and Γ^t , better ensuring the quality of the solution obtained. Hence, when the use of the MTZ model does not lead to proven optimality, the gaps are considerably larger than those of the CF formulation, which is usually closer to proving optimality given its stronger LP relaxation. Furthermore, with the CF formulation, CPLEX solved more instances to optimality in the deterministic case ($\Gamma^d = \Gamma^t = 0$) in about half the time on average. The CF model was also superior in terms of computation times and number of optimal solutions in instances where the uncertainty budgets (Γ^d and Γ^t) are less than or equal to 1. Moreover, using the CF model resulted in more instances solved to optimality than the MTZ model in instances with $\Gamma^d = 5$ and $\Gamma^t = 0$ and in all instances under uncertainty exclusively on travel time ($\Gamma^d = 0$ and $\Gamma^t > 0$). These results suggest that the CF formulation should be the preferred choice for solving instances involving travel time uncertainty and for instances with demand uncertainty when the budget is sufficiently small.

We also identified the traditional behavior found in the RO literature (Ordóñez 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 for both formulations. This is expected since the budget directly impacts the number of variables and constraints of the formulations. This behavior is noticeable in both formulations and for any combination of Γ^t and Γ^d .

5.1.2 Knapsack uncertainty set

Table 2 follows a similar structure to Table 1, but for different configurations of the knapsack set budgets Δ^d and Δ^t . 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 77.8%, but as a MIP model it resulted in fewer instances solved to optimality (461) than the MTZ-based model (1013), in longer running times (2576.13 seconds against 1494.33 seconds from the MTZ model, on average), and in larger average gaps (38.7% versus 10.1%). This is a consequence of the CF model requiring longer running times when solving the LP

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

Δ^d	Δ^t	MIP Model																
		LP Relaxation						MTZ						CF				
		MTZ			CF			MTZ			CF			CF				
		Obj	QLR	T	Obj	QLR	T	Obj	QLR	T	Obj	QLR	T	Obj	QLR	T	Obj	QLR
0	0	224.93	67.5	0.06	267.43	81.1	0.07	331.27	3.2	540.64	49	332.40	0.4	171.80	53			
20	0	224.95	65.1	0.08	267.88	77.9	3.94	341.22	6.3	906.79	134	342.31	3.3	1517.92	117			
40	0	224.61	64.2	0.16	268.22	76.5	14.46	349.43	8.3	1194.35	127	354.59	9.8	2311.07	76			
60	0	225.00	63.7	0.25	268.73	75.9	27.38	351.77	9.4	1302.62	124	359.95	14.0	2561.33	61			
0	20	224.97	67.3	0.42	267.57	80.9	58.31	332.38	6.7	1141.45	128	560.94	28.1	2858.51	40			
0	40	225.06	66.6	2.16	268.34	80.1	525.98	336.83	10.5	1773.87	100	1225.11	66.0	3131.38	31			
0	60	225.11	65.8	4.22	264.33	79.4	1558.49	341.86	14.3	2134.56	83	1262.68	71.4	3363.39	20			
20	20	224.99	64.9	0.50	268.01	77.7	79.92	342.51	8.7	1429.91	116	884.53	46.3	2950.67	36			
40	40	225.10	63.0	2.21	270.46	74.9	623.74	358.40	14.6	2098.46	86	1245.05	68.9	3381.72	20			
60	60	225.18	62.0	4.43	272.19	74.1	1909.10	365.15	18.8	2420.65	66	1345.83	78.8	3513.53	7			
All		224.99	65.0	1.45	268.31	77.8	480.14	345.08	10.1	1494.33	1013	791.34	38.7	2576.13	461			

relaxation for instances with larger Δ values (e.g., 1558.49 seconds for $\Delta^d = 0$ and $\Delta^t = 60$, and 1909.10 seconds for $\Delta^d = \Delta^t = 60$, on average), hindering the solver efficiency. Notably, instances with positive 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.

5.1.3 Two-knapsack uncertainty set

We now present the results of experiments conducted using the two-knapsack uncertainty set. We note that the models designed for this set were particularly challenging to solve. This is because they contain considerably more decision variables and constraints than the models based on the single knapsack uncertainty set, as the load variables are indexed based on the budgets of the two knapsacks. Consequently, the solver tends to use more memory to store and process the model and branch-and-bound tree.

The results obtained with the CF and MTZ-based compact formulations using the two-knapsack uncertainty set are summarized in Table 3. Similarly to other uncertainty sets, we performed experiments based on the LP relaxations and the full MIP formulations. The table has the same structure as Tables 1 and 2. The QLR values were computed based on the best results obtained by the BC algorithm presented in the next section.

The first conclusion we can draw from these results is the increased difficulty in solving the formulations, as the solver presented considerably larger optimality gaps and longer computation times when compared to the use of other uncertainty sets. Additionally, fewer instances were solved to optimality overall. This is intensified in instances with larger uncertainty budgets, as the size of the problem significantly increases with the increase in budget.

Comparing the formulation based on MTZ constraints to that based on CF constraints, we get similar conclusions to the single knapsack uncertainty set. The MTZ formulation performed generally better, with lower average gaps. However, the LP relaxation of the model based on the MTZ constraints is generally weaker than that obtained with the model based on CF constraints. Notably, the CF-based formulation was particularly affected by the increase in problem size, where in some configurations, namely $\Delta^d = 0$ and $\Delta^t = 60$, and $\Delta^d = 60$ and $\Delta^t = 60$, the solver was unable to even find an optimal solution for the LP relaxation of the model, for any instance in these groups.

5.2 Computational performance of the BC algorithms

5.2.1 Cardinality-constrained uncertainty set

In this section, we analyze the results of the BC algorithms developed in Sect. 4, considering the cardinality-constrained uncertainty set. Table 4 summarizes the results using a similar structure to that of Table 1. We do not show results for the LP relaxation, as they are similar to those presented in Table 1.

Table 3 Results of the MTZ-based and CF formulations for the two-knapsack RVRPTW, considering different configurations of budgets

Δ^d	Δ^t	LP relaxation													
		MTZ						CF							
		Obj	QLR	T	Obj	QLR	T	Obj	QLR	T	Obj	QLR	T		
0	0	220.58	66.4	0.004	267.23	81.3	0.08	341.03	3.1	538.07	32	341.03	0.2	161.46	35
20	0	220.63	64.0	2.18	268.07	78.4	115.51	366.56	18.5	2493.45	42	619.83	33.2	3199.34	20
40	0	220.70	63.6	21.57	261.12	77.2	1590.01	436.94	38.0	3261.09	18	1373.55	88.1	3511.84	3
60	0	220.79	63.4	92.23	320.94	86.4	1878.53	1241.68	74.3	3437.94	6	1394.34	94.6	3504.09	3
0	20	220.58	66.2	9.99	332.32	93.8	1392.40	533.70	37.7	2873.23	24	1307.54	86.7	3403.12	9
0	40	220.27	72.6	143.92	332.67	88.7	3110.29	1248.25	70.9	3205.98	12	1130.40	96.4	3600.00	0
0	60	240.48	86.1	576.61	-	-	3600.00	829.39	44.7	2120.04	10	-	-	3600.00	0
20	20	220.63	63.8	12.84	339.95	87.5	1413.14	629.68	46.3	3013.06	21	1365.13	91.7	3517.28	3
40	40	215.73	69.9	255.73	333.89	86.8	3278.16	1316.01	77.6	3504.33	3	1130.40	92.5	3600.00	0
60	60	225.72	86.5	647.08	-	-	3600.00	1118.39	69.7	3055.09	3	-	-	3600.00	0
All		222.61	70.3	176.22	307.02	85.0	1997.81	806.17	48.1	2750.23	171	1082.78	72.9	3176.15	73

Table 4 Average objective values and computation times of the BC algorithm for the cardinality-constrained RVRPTW

Γ^q	Γ^r	MTZ				CF			
		Obj	Gap	T	Opt	Obj	Gap	T	Opt
0	0	331.27	0.6	222.90	53	331.27	0.6	272.67	53
1	0	334.72	0.6	178.37	159	334.72	0.5	224.40	159
5	0	351.63	0.5	159.69	159	351.62	0.4	230.77	159
10	0	355.13	0.7	316.31	151	355.12	0.7	438.40	145
0	1	330.91	0.6	228.38	159	330.91	0.6	346.03	156
0	5	337.01	0.9	316.87	155	337.03	1.2	529.30	150
0	10	338.63	1.1	354.34	154	338.64	1.4	537.79	148
1	1	334.30	0.7	248.18	158	334.30	0.7	360.40	154
5	5	352.25	0.8	323.58	154	352.48	1.0	622.94	144
10	10	356.06	1.1	422.73	147	356.19	1.3	746.08	139
All		342.19	0.8	277.13	1449	342.23	0.8	430.88	1407

As expected, the BC algorithms outperformed their corresponding compact models as they start with a model having considerably fewer constraints and add cuts to ensure robustness as they are needed. The average solution times and gaps decreased, while the number of optimal solutions increased in all budget configurations. For example, the average solution time for the MTZ-based model decreased from 694.93 seconds to 277.13 seconds, the average gap reduced from 4.8% to 0.8%, and the number of instances solved to optimality increased from 1306 to 1449. Similarly, the average time for solving the CF model was almost halved, decreasing from 790.87 to 430.88 seconds. The average gap decreased from 2.1% to 0.8%, and 114 more instances were solved to optimality compared to solving the compact formulation.

The BC method with the MTZ-based formulation performed better than that with the CF formulation, taking less computation time to solve the instances (277.13 seconds against 430.88) and finding more optimal solutions (1449 against 1407). Additionally, this method was slightly superior even when solving most instances where the compact formulation had some advantage, namely the instances under uncertainty exclusively on travel time. The CF model had a better performance in terms of optimality gaps in instances under uncertainty exclusively on demand and $\Gamma^d \leq 5$, but both formulations solved the same number of instances to optimality in these configurations.

There are two main reasons for these superior results of the MTZ-based BC algorithm. The RCI generated using the CVRPS_{SEP} package strengthen 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 is particularly more helpful to the MTZ-based formulation, as it has a weak LP relaxation. Consequently, this reduces one advantage of the CF model over the MTZ-based model. 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.

5.2.2 Knapsack uncertainty set

Table 5 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 compared to solving the compact models, indicated by the lower average running times and the larger number of solved instances. More specifically, the BC algorithm with MTZ-based formulation solved 1456 instances compared to only 1013 solved by the general-purpose MIP solver with the compact formulation, and the BC with the CF formulation solved 1318 instances to optimality compared to 461 instances with the compact model. Additionally, the better performance of the BC algorithm with the MTZ-based model is noticeable regarding computation times, number of instances solved to optimality, and optimality gaps. This behavior is a consequence of the strengthening of the LP relaxation in the BC algorithms, further assisted by the reduced size of the MTZ-based model compared to the CF model.

While algorithms with the knapsack uncertainty set proved to be more challenging to solve than their deterministic counterparts, their performance was comparable to those with the cardinality-constrained uncertainty set. Notably, the best-performing configuration with the knapsack uncertainty set, which is the BC algorithm based on an MTZ formulation, solved more instances than the best configuration of the BC algorithm for the cardinality-constrained uncertainty set. To help visualize this behavior, Fig. 1 shows the number of instances solved to optimality for each pair of demand and travel time deviations (Dev^d, Dev^t), base models (MTZ and CF), and uncertainty sets (Card:

Table 5 Average objective values and computation times of the BC algorithm for the single knapsack-constrained RVRPTW

Δ^d	Δ^t	MTZ				CF			
		Obj	Gap	T	Opt	Obj	Gap	T	Opt
0	0	331.27	0.6	222.46	53	331.27	0.5	240.42	53
20	0	341.27	0.9	361.43	156	341.23	1.0	511.19	155
40	0	349.38	0.9	440.79	155	349.67	1.3	818.14	145
60	0	351.63	1.1	536.45	152	351.79	1.5	1030.92	142
0	20	332.23	0.7	232.59	159	332.23	0.7	300.11	156
0	40	334.77	0.5	176.67	159	334.97	0.9	618.26	144
0	60	335.39	1.0	331.58	159	336.95	1.8	876.51	143
20	20	342.33	0.9	331.45	157	342.30	1.0	532.20	152
40	40	352.99	1.0	492.62	154	353.57	1.9	1338.81	127
60	60	355.57	1.1	638.08	152	359.90	4.3	1800.41	101
All		342.68	0.9	376.41	1456	343.39	1.5	806.70	1318

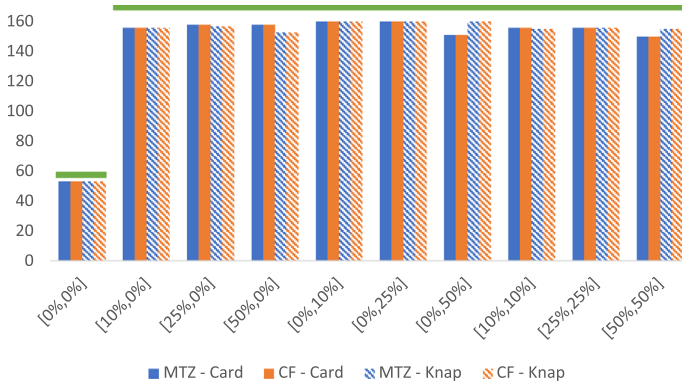


Fig. 1 Number of instances solved to optimality by the BC algorithms for each pair of demand and travel time budgets (Dev^d, Dev^t) for the cardinality- and knapsack-constrained RVRPTW

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 . As the chart suggests, the BC algorithms with the cardinality-constrained uncertainty set solved a similar number of instances to optimality to the knapsack uncertainty set. Thus, the strategy of relaxing the RO constraints and reintroducing them only if violated proved to be beneficial for both uncertainty sets.

5.2.3 Two-knapsack uncertainty set

Table 6 summarizes the results of the computational experiments with the BC algorithm using the two-knapsack uncertainty set. This table follows a similar structure to that of Tables 4 and 5. The results indicate that the BC algorithm vastly

Table 6 Average objective values and computation times of the BC algorithm for the two-knapsack RVRPTW

Δ^d	Δ^t	MTZ				CF			
		Obj	Gap (%)	T (s)	Opt	Obj	Gap (%)	T (s)	Opt
0	0	331.27	0.5	226.08	53	331.27	0.6	246.36	53
20	0	343.06	4.9	623.94	142	343.05	1.3	504.55	148
40	0	349.81	5.5	683.05	142	350.40	1.9	597.13	145
60	0	351.92	5.8	687.60	141	352.07	2.4	699.77	140
0	20	332.87	0.7	266.84	159	332.87	0.6	268.74	159
0	40	334.37	0.9	490.64	155	334.34	0.6	317.67	157
0	60	336.57	1.5	762.74	149	336.32	0.8	396.81	156
20	20	344.81	5.8	735.35	141	344.64	1.4	546.15	148
40	40	353.56	8.2	1165.81	128	353.19	2.5	799.32	140
60	60	357.70	10.3	1562.15	114	357.14	3.4	993.78	132
All		343.59	4.41	720.42	1324	343.53	1.54	537.03	1378

outperforms the compact formulations in terms of optimality gap, the number of instances solved to optimality, and the average computation times for this uncertainty set. This improvement is mainly because the base model of the BC algorithm is considerably smaller in terms of decision variables and constraints when compared to the compact formulations. Notably, the algorithm based on the CF formulations solves nearly 19 times more instances than its respective compact formulation.

One interesting observation is that the results of the two-knapsack uncertainty set show a notable discrepancy compared to the other sets studied: the BC algorithm based on the CF formulation consistently outperformed the algorithm based on the MTZ formulation, on average. This discrepancy can be attributed to the absence of (a large number of) robust time flow constraints, as they were replaced by feasibility cuts, in instances under travel time uncertainty. Additionally, this formulation has a stronger LP relaxation and generally better performance in the deterministic case compared to the MTZ-based formulation. This stronger LP relaxation helped the BC method to lift the lower bound more quickly, thus reducing the average optimality gap compared to the method with the MTZ-based formulation.

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 risk, which is represented by 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 7 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

Table 7 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^r	Dev=10%			Dev=25%			Dev=50%						
		Risk	PoR	SD	Best	Worst	SD	Best	Worst	SD	Best	Worst	SD	
		0	12.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	30.2	0.0	0.0
1	11.1	0.0	0.2	1.8	28.5	0.0	0.0	1.8	0.2	30.2	4.3	0.0	23.2	7.6
5	0.0	4.3	7.5	23.2	0.1	8.8	0.0	44.5	13.7	0.3	10.2	0.0	44.9	16.0
10	0.0	6.1	10.3	33.3	0.0	9.8	0.0	44.5	15.5	0.0	11.3	0.0	55.7	17.9
0	12.0	0.0	0.0	0.0	34.9	0.0	0.0	0.0	0.0	47.4	0.0	0.0	0.0	0.0
0	6.6	0.2	0.9	6.8	11.8	0.9	0.0	8.5	1.8	19.6	2.5	0.0	24.7	4.9
0	0.0	0.5	1.1	7.1	0.0	2.0	0.0	14.9	3.6	0.0	6.8	0.0	34.6	10.2
0	0.0	0.5	1.2	7.1	0.0	2.2	0.0	23.5	4.3	0.0	8.1	0.0	53.1	12.5
0	22.9	0.0	0.0	0.0	53.7	0.0	0.0	0.0	0.0	62.9	0.0	0.0	0.0	0.0
1	16.6	0.2	1.0	6.8	35.3	0.9	0.0	8.5	1.8	41.3	7.0	0.0	27.0	8.6
5	0.0	5.0	7.7	23.2	0.1	9.7	0.0	45.2	13.8	0.2	13.5	0.0	54.7	17.1
10	0.0	6.3	10.0	32.1	0.0	10.9	0.0	45.2	15.4	0.0	13.6	0.0	55.4	17.6
All		6.8	1.9	3.3	11.8	3.8	16.0	19.7	5.8	19.3	6.5	0.0	31.1	9.4

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.

Interestingly enough, the SD values of the cost increase for the robust solutions are relatively small. Hence, the differences in terms of PoR among solutions within an instance class are not particularly large. From a decision-maker's standpoint, this means that using the RO approach usually results in solutions with similar cost increment over the deterministic solution, with relatively low chances of large variability. This is particularly good for planning because, once the appropriate 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 2 shows the plots of the PoR versus the Risk for instances in classes C1, R1, and RC1, considering different deviations (0, 10, and 25%) and uncertainty

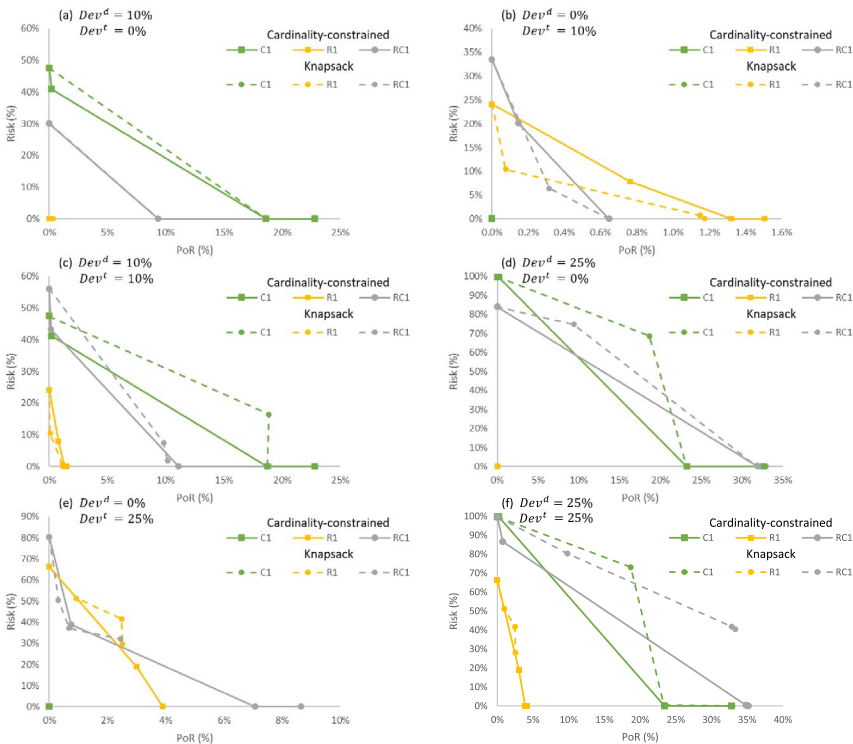


Fig. 2 Trade-off between price of robustness (PoR) and probability of constraint violation (Risk) for the instances in classes C1, R1, and RC1

budgets ($\Gamma \in \{0, 1, 5, 10\}$ and $\Delta \in \{0, 20, 40, 60\}$) for both uncertainty sets. 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 Fig. 2c and f, where an increase of more than 30% in the costs is needed to neutralize the risks. We also note that in some configurations, namely in class C1 in Fig. 2a, c, d, and f; in class R1 in 2(b); and class RC1 in 2(c), 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 0.5%.

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 8 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 resemble those obtained for the cardinality-constrained uncertainty set, when considering the same deviation level (*Dev*), particularly for those under with smaller deviation levels (10% or 25%). This is confirmed in the charts presented in Fig. 3, 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 sets. In particular, there are some configurations where both uncertainty sets yield similar solutions in terms of cost and risk, such as $\Gamma^d = 5$, $\Gamma^t = 0$ and $\Delta^d = 40$, $\Delta^t = 0$ with *Dev* = 10%; $\Gamma^d = 1$, $\Gamma^t = 0$ and $\Delta^d = 20$, $\Delta^t = 0$ with *Dev* = 50%; and $\Gamma^d = 10$, $\Gamma^t = 0$ and $\Delta^d = 60$, $\Delta^t = 0$ with *Dev* = 50%.

Table 8 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	SD	Risk	PoR	SD	Best	Worst	SD	Risk	PoR	SD	Best	Worst	SD	Risk	PoR	SD	Best	Worst	SD	Risk	PoR	SD	Best	Worst	SD			
0	0	12.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	28.0	0.0	0.0	0.0	0.0	0.0	0.0	30.2	0.0	0.0	0.0	0.0	0.0			
20	0	0.0	4.4	0.0	23.2	7.5	0.0	0.0	23.2	4.4	0.0	23.2	0.0	23.2	7.5	21.7	4.4	0.0	23.2	0.0	23.2	4.3	30.2	4.3	0.0	23.2	0.0	23.2	7.5		
40	0	0.0	4.3	0.0	23.2	7.6	0.0	0.0	23.2	8.3	0.0	44.5	0.0	44.5	13.5	0.0	8.3	0.0	44.5	0.0	44.5	8.3	13.3	8.3	0.0	44.9	0.0	44.5	13.5		
60	0	0.0	4.4	0.0	23.2	7.5	0.0	0.0	23.2	9.9	0.0	44.5	0.0	44.5	15.5	0.0	9.9	0.0	44.5	0.0	44.5	10.0	0.0	47.4	0.0	44.5	0.0	44.5	15.7		
0	0	12.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	34.9	0.0	0.0	0.0	0.0	0.0	0.0	46.2	0.2	0.0	0.0	0.0	0.0	0.0		
0	20	5.2	0.1	0.0	1.2	0.2	0.0	0.0	1.2	0.2	0.0	8.5	0.0	8.5	1.2	27.5	0.2	0.0	8.5	0.0	8.5	1.2	46.2	0.2	0.0	0.0	8.5	1.2	1.2		
0	40	0.4	0.4	0.0	6.8	1.1	0.0	0.0	6.8	1.1	0.0	7.1	0.0	7.1	1.1	19.9	0.7	0.0	7.1	0.0	7.1	1.6	40.3	1.1	0.0	17.1	0.0	17.1	2.7		
0	60	0.0	0.4	0.0	7.1	1.1	0.0	0.0	7.1	1.0	0.0	14.2	0.0	14.2	2.4	14.3	1.0	0.0	14.2	0.0	14.2	2.4	33.7	1.5	0.0	17.1	0.0	17.1	3.3		
0	0	22.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	53.7	0.0	0.0	0.0	0.0	0.0	62.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
20	20	7.8	4.5	0.0	24.7	7.6	0.0	0.0	24.7	4.6	0.0	23.2	0.0	23.2	7.5	43.8	4.6	0.0	23.2	0.0	23.2	4.6	62.5	4.6	0.0	23.2	0.0	23.2	7.5		
40	40	0.7	4.8	0.0	23.2	7.5	0.0	0.0	23.2	9.1	0.0	44.5	0.0	44.5	13.4	20.3	9.1	0.0	44.5	0.0	44.5	9.4	49.5	9.4	0.0	44.5	0.0	44.5	13.4		
60	60	0.3	4.8	0.0	23.2	7.5	0.0	0.0	23.2	10.7	0.0	44.5	0.0	44.5	15.4	15.7	10.7	0.0	44.5	0.0	44.5	11.3	32.3	11.3	0.0	44.5	0.0	44.5	15.6		
All		5.1	2.3	0.0	13.0	4.0	0.0	0.0	13.0	4.1	0.0	21.3	0.0	21.3	6.5	23.3	4.1	0.0	21.3	0.0	21.3	4.2	37.4	4.2	0.0	22.3	0.0	22.3	6.7		

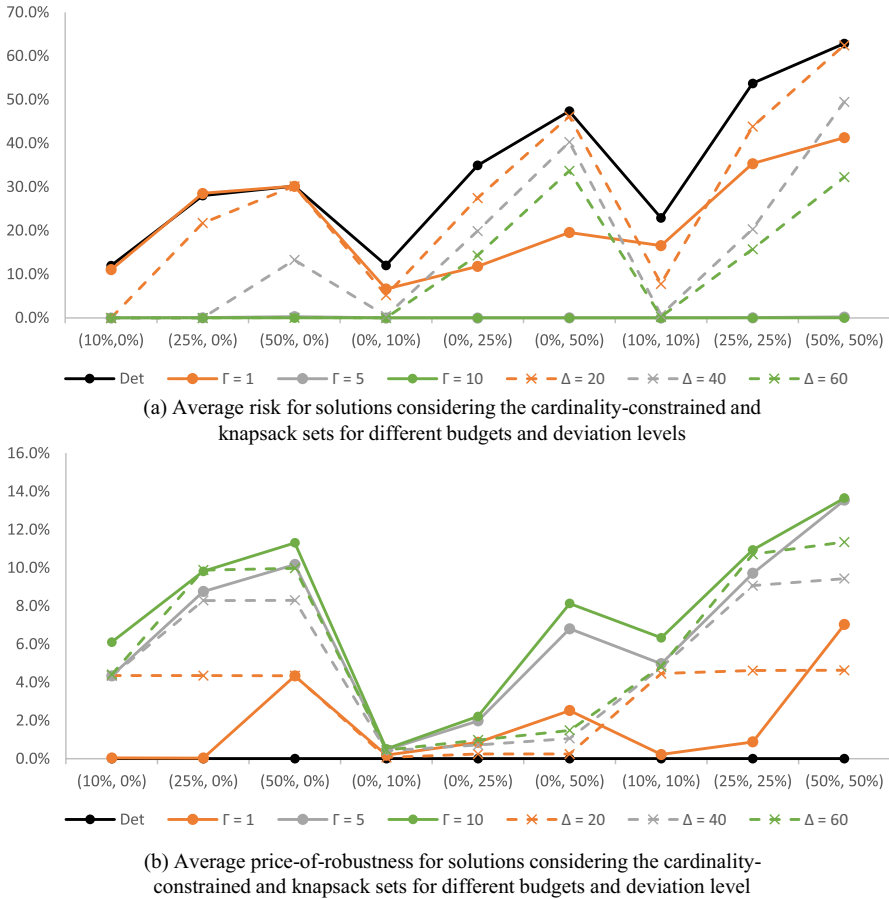


Fig. 3 Average Risk and PoR for different budget values and deviation levels

It is worth noting, however, that several configurations provided unique solutions that could not be achieved by the other uncertainty sets. Some of these solutions offer interesting trade-offs or a strictly superior performance in terms of cost and risk to those offered by the other sets. For example, the solutions using the knapsack uncertainty set with $\Delta^d = 0$ and $\Delta^t = 20$ with $Dev = 10\%$ are on average strictly better than those obtained with the cardinality-constrained set with $\Gamma^d = 0$ and $\Gamma^t = 1$, as they have lower risk and cost. A similar behavior can be observed in the solutions with $Dev = 25\%$ with $\Delta^d = 40$ and $\Delta^t = 0$, which have the same zero risk but a slightly lower cost compared to the solutions obtained with the cardinality-constrained uncertainty set with $\Gamma^d = 10$ and $\Gamma^t = 0$. Conversely, there are some solutions where the cardinality-constrained set presents more competitive solutions, such as instances under demand and travel time uncertainty with $Dev = 50\%$, where solutions with $\Gamma^d = \Gamma^t = 1$ are superior to those obtained with $\Delta^d = \Delta^t = 40$.

A particularity of the knapsack uncertainty set is that the robustness of the solution does not increase as much as the cardinality-constrained uncertainty set for instances with larger deviations. For example, for instances with travel time uncertainties of $Dev = 25\%$ and $Dev = 50\%$, we were unable to obtain null-risk solutions using the chosen budgets for the knapsack uncertainty set. This is because, unlike the cardinality-constrained set, the budget of the knapsack set is independent of the deviation, and once the knapsack is completely filled, the solution may not change even in scenarios with larger deviations. Nevertheless, 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 in real-world systems as estimating the total deviation in a route is easier than the number of arcs/nodes attaining their worst-case value.

5.3.3 Two-knapsack uncertainty set

Table 9 summarizes the PoR and risk of the solutions obtained for the two-knapsack uncertainty set for each combination of budgets (Δ^d and Δ^t) and deviation (Dev). We also present the best, worst, and standard deviation (SD) values for the PoR. The solution behavior for this uncertainty set is similar to the others. The risk decreases and the cost increases as higher budgets of uncertainty are considered. Conversely, the risks and costs tend to increase for scenarios with larger deviations. Regardless, the solutions obtained with this set were unique compared to the other sets in terms of risk and cost, presenting distinct trade-offs not achievable by other uncertainty sets. For example, this is the only uncertainty set able to provide solutions that offer trade-offs with non-zero risks of less than 5% in instances with 10% deviation ($\Delta^d = 0$ and $\Delta^t = 20$; $\Delta^d = 0$ and $\Delta^t = 40$; $\Delta^d = 20$ and $\Delta^t = 20$; and $\Delta^d = 40$ and $\Delta^t = 40$). This type of solution may appeal to decision-makers who accept the relatively low risk of infeasibility in exchange for a small reduction in cost.

In the experiments, the two-knapsack uncertainty set was, by construction, more conservative than the knapsack uncertainty set with the same budget Δ^d and Δ^t . This is because the least conservative scenario for a route in the two-knapsack uncertainty set, where each customer visited in the route is in the same knapsack, is equivalent to the single knapsack uncertainty set with the same budget. Moreover, in the cases where customers from different knapsacks are visited on the route, the total budget in the route increases because the deviation of a node in one knapsack can be considered even if the other knapsack is completely filled.

5.4 Solution characteristics and managerial insights

Finally, we analyze the behavior of robust solutions and provide some managerial insights related to different uncertainty sets. For this purpose, we define and compute the customer-per-route ratio metric that can be used to describe the characteristics of

Table 9 Average probability of constraint violation (Risk) and average price of robustness (PoR) of the solutions obtained for the two-knapsack 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			SD			Best			Worst			SD			Risk			PoR			Best			Worst			SD		
		Risk	PoR	SD	Best	Worst	SD	Risk	PoR	SD	Best	Worst	SD	Risk	PoR	SD	Best	Worst	SD	Risk	PoR	SD	Best	Worst	SD	Risk	PoR	SD	Best	Worst	SD			
0	0	12.0	0.0	0.0	0.0	0.0	28.0	0.0	0.0	0.0	0.0	0.0	30.2	0.0	0.0	0.0	0.0	0.0	30.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
20	0	0.0	4.3	0.0	23.2	7.6	13.0	5.4	0.0	24.7	9.1	29.2	5.4	0.0	24.7	9.1	29.2	5.4	0.0	24.7	9.1	29.2	5.4	0.0	24.7	9.1	29.2	5.4	0.0	24.7	9.1			
40	0	0.0	4.3	0.0	23.2	7.6	0.0	8.6	0.0	47.2	13.9	10.2	9.2	0.0	47.2	13.9	10.2	9.2	0.0	47.2	13.9	10.2	9.2	0.0	47.2	13.9	10.2	9.2	0.0	47.2	13.9			
60	0	0.0	4.3	0.0	23.2	7.6	0.0	9.9	0.0	45.2	15.5	0.0	10.7	0.0	45.2	15.5	0.0	10.7	0.0	45.2	15.5	0.0	10.7	0.0	45.2	15.5	0.0	10.7	0.0	45.2	15.5			
0	0	12.0	0.0	0.0	0.0	0.0	34.9	0.0	0.0	0.0	0.0	0.0	47.4	0.0	0.0	0.0	0.0	0.0	47.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
0	20	4.1	0.2	0.0	2.5	0.5	23.4	0.4	0.0	8.5	1.3	42.4	0.4	0.0	8.5	1.3	42.4	0.4	0.0	8.5	1.3	42.4	0.4	0.0	8.5	1.3	42.4	0.4	0.0	8.5	1.3			
0	40	0.2	0.4	0.0	6.8	1.1	15.9	0.7	0.0	8.5	1.6	36.9	0.9	0.0	8.5	1.6	36.9	0.9	0.0	8.5	1.6	36.9	0.9	0.0	8.5	1.6	36.9	0.9	0.0	8.5	1.6			
0	60	0.0	0.5	0.0	7.1	1.2	8.5	1.1	0.0	14.2	2.4	28.3	1.8	0.0	14.2	2.4	28.3	1.8	0.0	14.2	2.4	28.3	1.8	0.0	14.2	2.4	28.3	1.8	0.0	14.2	2.4			
0	0	22.9	0.0	0.0	0.0	0.0	53.7	0.0	0.0	0.0	0.0	62.9	0.0	0.0	0.0	0.0	62.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
20	20	4.3	4.6	0.0	23.2	7.5	32.0	5.8	0.0	24.7	9.0	58.0	5.8	0.0	24.7	9.0	58.0	5.8	0.0	24.7	9.0	58.0	5.8	0.0	24.7	9.0	58.0	5.8	0.0	24.7	9.0			
40	40	0.2	4.8	0.0	23.2	7.5	16.0	9.3	0.0	46.3	13.7	45.6	9.7	0.0	46.3	13.7	45.6	9.7	0.0	46.3	13.7	45.6	9.7	0.0	46.3	13.7	45.6	9.7	0.0	46.3	13.7			
60	60	0.0	5.1	0.0	23.2	7.8	9.0	11.0	0.0	46.0	15.6	27.0	12.2	0.0	46.0	15.6	27.0	12.2	0.0	46.0	15.6	27.0	12.2	0.0	46.0	15.6	27.0	12.2	0.0	46.0	15.6			
All		4.6	2.4	0.0	13.0	4.0	19.5	4.3	0.0	22.1	6.8	34.9	4.7	0.0	22.1	6.8	34.9	4.7	0.0	22.1	6.8	34.9	4.7	0.0	22.1	6.8	34.9	4.7	0.0	22.1	6.8			

the solutions for different configurations in terms of deviation and budget. The table showing the average value for this parameter, considering each budget configuration and deviation level, is presented in “Appendix 3”.

The customer-per-route ratio is calculated by finding the average number of customers visited in each route for a given solution. It can provide a general understanding of the structure of the routes and help assess the number of vehicles used in a solution. Figure 4 presents three charts showing the average customer-per-route ratio for each budget configuration, considering the three studied uncertainty sets. The blue lines represent results with the cardinality-constrained set (*Card*), gray lines represent results with the single-knapsack uncertainty set (*Sing Knap*), and orange lines represent the results with the two-knapsack set (*Two-Knap*). Each chart represents the average results for each deviation level (10%, 25%, and 50%). The budgets are displayed on horizontal axes in format $[\Gamma^d, \Gamma^t]$ or $[\Delta^d, \Delta^t]$ according to the uncertainty set.

The charts show that as the deviation increases, there is a tendency to reduce the number of customers visited per route. This trend is more pronounced for larger deviations (25% and 50%). With all instances having 25 customers, it also means an increase in the number of vehicles. When comparing the individual behavior of uncertainty sets, we observe that scenarios with larger deviation lead to more intense changes in solutions based on the cardinality-constrained set with budget increases. However, this behavior reflects the tendency of this set to provide zero risk at a higher cost compared to both knapsack sets in instances with small deviation levels, and to be the only set capable of finding risk-free solutions in instances with larger deviation parameters. In several cases, the knapsack sets were able to provide solutions with similar risk levels at lower costs and smaller changes

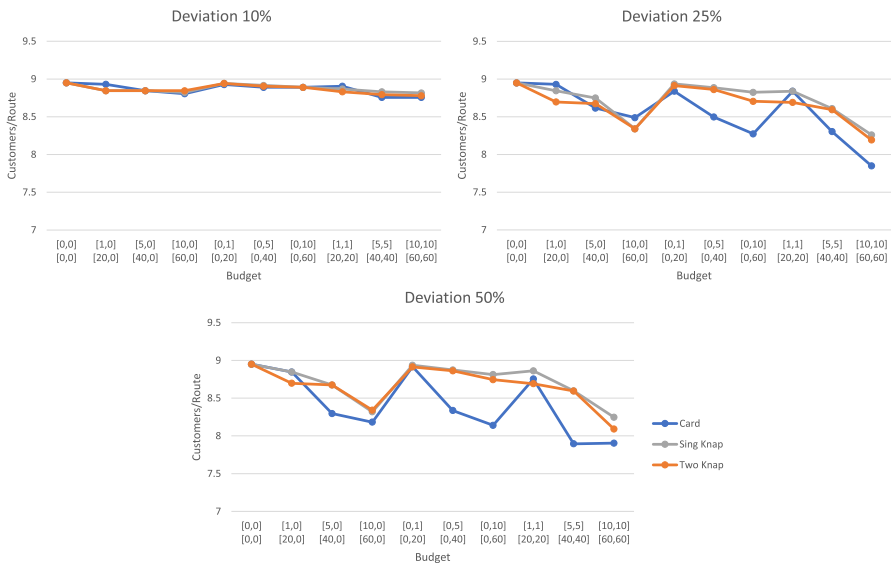


Fig. 4 Average number of customers per route for each budget configuration and uncertainty set

from the deterministic solution in terms of customer-per-route ratio. Therefore, from a decision-maker's perspective, it may be advantageous to evaluate and implement solutions based on a knapsack uncertainty set over a cardinality-constrained set. Additionally, even when solutions provided by a knapsack uncertainty are not strictly superior to those obtained using a cardinality-constrained set they often offer interesting trade-offs better matching the decision maker's strategy.

Figure 4 also illustrates the generally more conservative results of the two-knapsack uncertainty set compared to the single knapsack set, when they are defined using the same budget values. This can be seen from the consistently lower customer-per-route ratio of the two-knapsack uncertainty set in most configurations. The reason for this is that for a given budget Δ^d/Δ^t , the total deviation on a route in the single knapsack set is limited by this budget, while in the two-knapsack set it may be limited by twice that value because each knapsack has its own budget Δ^d/Δ^t . Regarding the difference in behavior patterns between the cardinality-constrained and knapsack uncertainty sets, it is because the budgets for both knapsack sets are fixed amounts that do not depend on the deviations; whereas, with cardinality-constrained sets, the total deviation absorbed depends on both budget size and deviation level.

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 that considers 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.

Computational results with the proposed compact models using benchmark instances from the literature suggest that the MTZ-based formulation yields the best overall performance for both uncertainty sets, even though the CF model exhibits stronger LP relaxations and promotes a better performance of the solver on the deterministic case. An exception to this behavior is the solution of instances under uncertainty exclusively on travel times considering the cardinality-constrained uncertainty set, in which the CF leads to a better performance in terms of number of instances solved to optimality and optimality gaps, thus its use is recommended in this 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 types of uncertainty sets, we observe that the use of a 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. Based on our experiments, 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.

Appendix 1: 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_i^d (demand) or Δ_i^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, time 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:

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

$$\text{s.t. (6), (7), (12), and to} \tag{46}$$

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

$$u_{j\delta_1\delta_2} \geq u_{i(\delta_1 - \hat{\delta}_1)\delta_2} + \bar{d}_j + \hat{d}_j + Q(x_{ij} - 1), (i,j) \in A, j \in S_1^d, j \notin S_2^d, \hat{d}_j \leq \delta_1 \leq \Delta_1^d, 0 \leq \delta_2 \leq \Delta_2^d, \tag{47}$$

$$u_{j\delta_1\delta_2} \geq u_{i\delta_1(\delta_2 - \hat{\delta}_2)} + \bar{d}_j + \hat{d}_j + Q(x_{ij} - 1), (i,j) \in A, j \notin S_1^d, j \in S_2^d, 0 \leq \delta_1 \leq \Delta_1^d, \hat{d}_j \leq \delta_2 \leq \Delta_2^d, \tag{48}$$

$$u_{j\delta_1\delta_2} \geq u_{i(\delta_1 - \hat{\delta}_1)(\delta_2 - \hat{\delta}_2)} + \bar{d}_j + \hat{d}_j + Q(x_{ij} - 1), (i,j) \in A, j \in S_1^d, j \in S_2^d, \hat{d}_j \leq \delta_1 \leq \Delta_1^d, \hat{d}_j \leq \delta_2 \leq \Delta_2^d, \tag{49}$$

$$u_{j\Delta_1^d\delta_2} \geq u_{i(\Delta_1^d - \lambda)\delta_2} + \bar{d}_j + \lambda + Q(x_{ij} - 1), (i,j) \in A, j \in S_1^d, j \notin S_2^d, 0 \leq \lambda \leq \hat{d}_j, 0 \leq \delta_2 \leq \Delta_2^d, \tag{50}$$

$$u_{j\delta_1\Delta_2^d} \geq u_{i\delta_1(\Delta_2^d-\lambda)} + \bar{d}_j + \lambda + Q(x_{ij} - 1), (i, j) \in A, j \notin S_1^d, j \in S_2^d, 0 \leq \lambda \leq \hat{d}_j, 0 \leq \delta_1 \leq \Delta_1^d, \quad (51)$$

$$u_{j\delta_1\Delta_2^d} \geq u_{i(\delta_1-\lambda)(\Delta_2^d-\lambda)} + \bar{d}_j + \lambda + Q(x_{ij} - 1), (i, j) \in A, j \in S_1^d, j \in S_2^d, \lambda \leq \hat{d}_j, \quad (52)$$

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

$$w_{j\delta_1\delta_2} \geq w_{i(\delta_1-\lambda)\delta_2} + \bar{t}_{ij} + s_i + \lambda + b_{n+1}(x_{ij} - 1), \\ (i, j) \in A, (i, j) \in S_1^t, (i, j) \notin S_2^t, 0 \leq \lambda \leq \min\{\hat{t}_{ij}, \delta_1\} \leq \Delta_1^t, 0 \leq \delta_2 \leq \Delta_2^t, \quad (54)$$

$$w_{j\delta_1\delta_2} \geq w_{i\delta_1(\delta_2-\lambda)} + \bar{t}_{ij} + s_i + \lambda + b_{n+1}(x_{ij} - 1), \\ (i, j) \in A, (i, j) \notin S_1^t, (i, j) \in S_2^t, 0 \leq \delta_1 \leq \Delta_1^t, \lambda \leq \min\{\hat{t}_{ij}, \delta_2\} \leq \Delta_2^t, \quad (55)$$

$$w_{j\delta_1\delta_2} \geq w_{i(\delta_1-\lambda)(\delta_2-\lambda)} + \bar{t}_{ij} + s_i + \lambda + b_{n+1}(x_{ij} - 1), \\ (i, j) \in A, (i, j) \in S_1^t, (i, j) \in S_2^t, 0 \leq \lambda \leq \hat{t}_{ij}, \lambda \leq \delta_1 \leq \Delta_1^t, \lambda \leq \delta_2 \leq \Delta_2^t, \quad (56)$$

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

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

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

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 .

Appendix 2: 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:

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

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

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

$$\begin{aligned} \sum_{j:(i,j) \in A} f_{ij\delta_1\delta_2} &\geq \bar{d}_i + \hat{d}_i + \sum_{h:(h,i) \in A} f_{hi(\delta_1 - \hat{d}_i)\delta_2}, \quad i \\ &\in N, i \in S_1^d, i \notin S_2^d, \hat{d}_i \leq \delta_1 \leq \Delta_1^d, 0 \leq \delta_2 \leq \Delta_2^d, \end{aligned} \tag{62}$$

$$\begin{aligned} \sum_{j:(i,j) \in A} f_{ij\delta_1\delta_2} &\geq \bar{d}_i + \hat{d}_i + \sum_{h:(h,i) \in A} f_{hi\delta_1(\delta_2 - \hat{d}_i)}, \quad i \in N, \\ i \notin S_1^d, i \in S_2^d, 0 \leq \delta_1 \leq \Delta_1^d, \hat{d}_i \leq \delta_2 \leq \Delta_2^d, \end{aligned} \tag{63}$$

$$\begin{aligned} \sum_{j:(i,j) \in A} f_{ij\delta_1\delta_2} &\geq \bar{d}_i + \hat{d}_i + \sum_{h:(h,i) \in A} f_{hi(\delta_1 - \hat{d}_i)(\delta_2 - \hat{d}_i)}, \\ i \in N, i \in S_1^d, i \in S_2^d, 0 \leq \delta_1 \leq \Delta_1^d, \hat{d}_i \leq \delta_2 \leq \Delta_2^d, \end{aligned} \tag{64}$$

$$\begin{aligned} \sum_{j:(i,j) \in A} f_{ij\Delta_1^d\delta_2} &\geq \bar{d}_i + \lambda + \sum_{h:(h,i) \in A} f_{hi(\Delta_1^d - \lambda)\delta_2}, \quad i \in N, \\ i \in S_1^d, i \notin S_2^d, 0 \leq \delta_2 \leq \Delta_2^d, 0 \leq \lambda < \hat{d}_i, \end{aligned} \tag{65}$$

$$\begin{aligned} \sum_{j:(i,j) \in A} f_{ij\delta_1\Delta_2^d} &\geq \bar{d}_i + \lambda + \sum_{h:(h,i) \in A} f_{hi\delta_1(\Delta_2^d - \lambda)}, \quad i \in N, \\ i \notin S_1^d, i \in S_2^d, 0 \leq \delta_1 \leq \Delta_1^d, 0 \leq \lambda < \hat{d}_i, \end{aligned} \tag{66}$$

$$\sum_{j:(i,j) \in A} f_{ij\Delta_1^d\Delta_2^d} \geq \bar{d}_i + \lambda + \sum_{h:(h,i) \in A} f_{hi(\Delta_1^d - \lambda)(\Delta_2^d - \lambda)}, \quad i \in N, i \in S_1^d, i \in S_2^d, 0 \leq \lambda < \hat{d}_i, \tag{67}$$

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

$$\sum_{j:(i,j) \in A} g_{ij\delta_1\delta_2} \geq \sum_{\substack{h : (h,i) \in A \\ \lambda \leq \hat{t}_{hi}}} (g_{hi(\delta_1-\lambda)\delta_2} + (\bar{t}_{hi} + \lambda)x_{hi}) + s_i, \quad (69)$$

$$i \in N, (i,j) \in S_1^t, (i,j) \notin S_2^t, 0 \leq \lambda \leq \delta_1 \leq \Delta_1^t, 0 \leq \delta_2 \leq \Delta_2^t,$$

$$\sum_{j:(i,j) \in A} g_{ij\delta_1\delta_2} \geq \sum_{\substack{h : (h,i) \in A \\ \lambda \leq \hat{t}_{hi}}} (g_{hi\delta_1(\delta_2-\lambda)} + (\bar{t}_{hi} + \lambda)x_{hi}) + s_i, \quad (70)$$

$$i \in N, (i,j) \notin S_1^t, (i,j) \in S_2^t, 0 \leq \delta_1 \leq \Delta_1^t, 0 \leq \lambda \leq \delta_2 \leq \Delta_2^t,$$

$$\sum_{j:(i,j) \in A} g_{ij\delta_1\delta_2} \geq \sum_{\substack{h : (h,i) \in A \\ \lambda \leq \hat{t}_{hi}}} (g_{hi(\delta_1-\lambda)(\delta_2-\lambda)} + (\bar{t}_{hi} + \lambda)x_{hi}) + s_i, \quad (71)$$

$$i \in N, (i,j) \in S_1^t, (i,j) \in S_2^t, 0 \leq \lambda \leq \delta_1 \leq \Delta_1^t, 0 \leq \lambda \leq \delta_2 \leq \Delta_2^t,$$

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

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

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

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

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

Appendix 3: Results for customer-per-route ratio

In this section, we present the average results for the customer-per-route ratio discussed in Sect. 5.4. The results are summarized in Table 10, which shows the average ratio for each budget and deviation (*Dev*) configuration for each uncertainty set. The budgets are displayed in the format $[\Gamma^d, \Gamma^t] / [\Delta^d, \Delta^t]$ according to the uncertainty set.

Acknowledgements We are thankful to the Editors and the two anonymous reviewers for carefully reviewing the manuscript and providing valuable suggestions. We also thank Carlos Neves and Marcelo de Freitas Cavalcante, from Federal University of Paraiba (Brazil), for sending us their feedback and pointing out a few typos and inconsistencies in the first version of our manuscript. This work was supported by São Paulo Research Foundation (FAPESP) [Grant Numbers 13/07375-0, 19/22235-6, 19/23596-2, 22/05803-3], Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) [Finance Code 001], the National Council for Scientific and Technological Development

Table 10 Average customer-per-route ratio for each budget and deviation level

Budget	Dev = 10%			Dev = 25%			Dev = 50%		
	Card.	Single Knap.	Two-Knap.	Card.	Single Knap.	Two-Knap.	Card.	Single Knap.	Two-Knap.
	[0,0]/[0,0]	8.9500	8.9500	8.9500	8.9500	8.9500	8.9500	8.9500	8.9500
[1,0]/[20,0]	8.9307	8.8457	8.8457	8.9307	8.8457	8.6971	8.8457	8.8457	8.6971
[5,0]/[40,0]	8.8457	8.8457	8.8457	8.6157	8.7493	8.6748	8.2959	8.6748	8.6748
[10,0]/[60,0]	8.8054	8.8234	8.8457	8.4891	8.3405	8.3405	8.1816	8.3182	8.3405
[0,1]/[0,20]	8.9275	8.9423	8.9423	8.8396	8.9363	8.9139	8.9165	8.9363	8.9139
[0,5]/[0,40]	8.8904	8.9156	8.9052	8.4975	8.8867	8.8620	8.3363	8.8735	8.8620
[0,10]/[0,60]	8.8904	8.8904	8.8904	8.2743	8.8248	8.7057	8.1402	8.8118	8.7448
[1,1]/[20,20]	8.9052	8.8680	8.8309	8.8396	8.8396	8.6911	8.7533	8.8620	8.6911
[5,5]/[40,40]	8.7566	8.8309	8.7938	8.3043	8.6093	8.5945	7.8949	8.5963	8.5945
[10,10]/[60,60]	8.7570	8.8161	8.7789	7.8496	8.2602	8.1932	7.9036	8.2471	8.0909

(CNPq) [Grant Numbers 405702/2021-3, 314079/2023-8], and the Canadian Natural Sciences and Engineering Research Council (NSERC) [Grant Number 2019-00094] and the Fonds de recherche du Québec–Nature et technologie [programme Bourses de doctorat en recherche, dossier 347556].

References

- Agra A, Christiansen M, Figueiredo R et al (2012) Layered formulation for the robust vehicle routing problem with time windows. In: Mahjoub AR, Markakis V, Milis I et al (eds) *Combinatorial optimization*. Springer, Berlin, pp 249–260
- Agra A, Christiansen M, Figueiredo R et al (2013) The robust vehicle routing problem with time windows. *Comput Oper Res* 40(3):856–866
- Bartolini E, Goeke D, Schneider M et al (2021) The robust traveling salesman problem with time windows under knapsack-constrained travel time uncertainty. *Transp Sci* 55(2):371–394
- Bertsimas D, Sim M (2003) Robust discrete optimization and network flows. *Math Program* 98(1):49–71
- Bertsimas D, Sim M (2004) The price of robustness. *Oper Res* 52(1):35–53
- Bianchessi N, Irnich S (2019) Branch-and-cut for the split delivery vehicle routing problem with time windows. *Transp Sci* 53(2):442–462
- Boysen N, Fedtke S, Schwerdfeger S (2021) Last-mile delivery concepts: a survey from an operational research perspective. *OR Spectrum* 43:1–58
- Caunhye AM, Alem D (2023) Practicable robust stochastic optimization under divergence measures with an application to equitable humanitarian response planning. *OR Spectrum* 1–48
- De La Vega J, Munari P, Morabito R (2020) Exact approaches to the robust vehicle routing problem with time windows and multiple deliverymen. *Comput Oper Res* 124:105062
- De La Vega J, Gendreau M, Morabito R et al (2023) An integer L-shaped algorithm for the vehicle routing problem with time windows and stochastic demands. *Eur J Oper Res* 308(2):676–695
- Gavish B (1984) The delivery problem: new cutting planes procedures. In: *TIMS XXVI Conference*, Copenhagen
- Gendreau M, Jabali O, Rei W (2014) Stochastic vehicle routing problems. In: *Vehicle routing: problems, methods, and applications*, 2nd edn. Society for Industrial and Applied Mathematics, pp 213–239
- Gounaris CE, Wiesemann W, Floudas CA (2013) The robust capacitated vehicle routing problem under demand uncertainty. *Oper Res* 61(3):677–693
- Gouveia L (1995) A result on projection for the vehicle routing problem. *Eur J Oper Res* 85(3):610–624
- Hoogendoorn Y, Spliet R (2023) An improved integer L-shaped method for the vehicle routing problem with stochastic demands. *INFORMS J Comput* 35(2):423–439
- Langevin A, Desrochers M, Desrosiers J et al (1993) A two-commodity flow formulation for the traveling salesman and the makespan problems with time windows. *Networks* 23(7):631–640
- Lee C, Lee K, Park S (2012) Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *J Oper Res Soc* 63:1294–1306
- Letchford AN, Salazar-González JJ (2006) Projection results for vehicle routing. *Math Program* 105(2):251–274
- Letchford AN, Salazar-González JJ (2015) Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. *Eur J Oper Res* 244(3):730–738
- Lysgaard J, Letchford A, Eglese R (2004) A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math Program* 100:423–445
- Miller C, Tucker A, Zemlin R (1960) Integer programming formulation of traveling salesman problems. *J Assoc Comput Mach* 7(1):326–9
- Minoux M (2009) On robust maximum flow with polyhedral uncertainty sets. *Optim Lett* 3(3):367–376
- Munari P, Savelsbergh M (2022) Compact formulations for split delivery routing problems. *Transp Sci* 56(4):1022–1043
- Munari P, Moreno A, De La Vega J et al (2019) The robust vehicle routing problem with time windows: compact formulation and branch-price-and-cut method. *Transp Sci* 53(4):1043–1066
- Ordonez F (2010) Robust vehicle routing. In: Hasenbein JJ, Gray P, Greenberg HJ (eds) *Risk and optimization in an uncertain world*. Inform, Catonsville, MD, chap 7, pp 153–178
- Pessoa AA, Poss M, Sadykov R et al (2021) Branch-cut-and-price for the robust capacitated vehicle routing problem with knapsack uncertainty. *Oper Res* 69(3):739–754

- Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper Res* 35(2):254–265
- Subramanyam A, Repoussis PP, Gounaris CE (2020) Robust optimization of a broad class of heterogeneous vehicle routing problems under demand uncertainty. *INFORMS J Comput* 32(3):661–681
- Sungur I, Ordonez F, Dessouky M (2008) A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Trans (Inst Ind Eng)* 40(5):509–523
- Wang A, Subramanyam A, Gounaris C (2021) Robust vehicle routing under uncertainty via branch-price-and-cut. *Optim Eng* 23:1895–1948
- Yu Q, Cheng C, Zhu N (2022) Robust team orienteering problem with decreasing profits. *INFORMS J Comput* 34(6):3215–3233

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.