Rafael Ajudarte de Campos

Aircraft routing under uncertainty via robust optimization

São Carlos - SP, Brazil
2022

Rafael Ajudarte de Campos

Aircraft routing under uncertainty via robust optimization

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Mestre em Engenharia de Produção.

Orientador(a): Prof. Pedro Munari, Dr.

São Carlos – SP, Brasil

2022

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Engenharia de Produção

Folha de Aprovação

Defesa de Dissertação de Mestrado do candidato Rafael Ajudarte de Campos, realizada em 02/09/2022.

Comissão Julgadora:

Prof. Dr. Pedro Augusto Munari Junior (UFSCar)

Prof. Dr. Reinaldo Morabito Neto (UFSCar)

Prof. Dr. Bruno Petrato Bruck (UFPB)

Prof. Dr. Walton Pereira Coutinho (UFPE)

Prof. Dr. Douglas José Alem Junior (Edin)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Engenharia de Produção.

Statement of Authorship

I hereby declare that the dissertation submitted is my own work. All direct or indirect sources used are acknowledged as references. I further declare that I have not submitted this dissertation at any other institution in order to obtain a degree.

Statement of financial support

We are thankful to the *São Paulo Research Foundation* (FAPESP), process number 2019/22235-6, for funding this research. This study was also financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES)* – Finance Code 001.

Acknowledgements

Embora minha dissertação esteja em inglês, eu optei por escrever meus agradecimentos em português pois acredito que é a melhor língua para poder expressar meus sentimentos às pessoas que eu gostaria de agradecer.

Primeiramente, gostaria de agradecer aos meus pais, Júlio e Erliane, por terem sempre me dado todo o suporte necessário para obter esta conquista, além de todas as outras em minha vida.

Queria agradecer também ao meu irmão, Daniel, uma das melhores pessoas que conheço e com quem posso sempre buscar para ter uma conversa interessante.

Sou muito grato também à minha noiva, Beatriz, que me acompanha nesta jornada desde minha graduação e que sempre me apoiou e me estimulou a seguir em frente.

Agradeço também aos meus avós, Pedro e Maria, e meu tio Marco por estarem sempre preocupados comigo e meu irmão e nos apoiando em cada etapa de nossas vidas.

Agradeço especialmente ao meu orientador, Prof. Dr. Pedro Munari, que me orientou e trabalhou comigo no desenvolvimento deste trabalho, provendo estrutura, conselhos e palavras de encorajamento. Sou muito grato por todas as portas que você me abriu.

Agradeço aos membros da banca por todas as revisões e contribuições para melhoria deste trabalho.

Agradeço a todos meus amigos e colegas do Grupo de Pesquisa Operacional (GPO) da UFSCar, em especial Aura e Mateus, que frequentemente religaram meus computadores durante a pandemia.

Sou grato também aos meus amigos, Linneu, Diego, Giovane e Mariana com quem sempre posso contar.

Agradeço, por fim, à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), número de processo 2019/22235-6, e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) pelo financiamento desta pesquisa.

Abstract

We address the robust vehicle routing problem (RVRP), focusing on the development of mathematical models and solution methods to incorporate uncertainties regarding travel times and demand, in traditional and practical variants. We are particularly interested in a practical variant, the aircraft routing problem, motivated by the real case of an on-demand airline company. Features such as heterogeneous fleet, time windows and maintenance requests are incorporated into robust optimization models that allow for the variability of uncertain parameters to be addressed. In particular, a new type of commodity flow model formulation, not yet explored in the robust optimization literature, even in classical variants, was developed for both the traditional RVRP and the aircraft routing problem. Moreover, we propose new compact models and tailored branch-and-cut methods considering different types of uncertainty sets, namely the cardinality constrained set and the single and multiple knapsack sets, using a recent approach based on dynamic programming to obtain the robust counterparts. The developed approaches were implemented and analyzed through computational experiments using instances from the literature as well as real-world data related to aircraft routing.

Keywords: Vehicle Routing Problem, Robust Optimization, Aircraft Routing

List of Figures

List of Tables

Contents

1 Introduction

In general, transportation costs represent between 10 to 20% of a product price (Khooban, 2011). Reducing these costs brings advantages to a company, as this can result in products with lower prices or a greater mark up. The Vehicle Routing Problem (VRP) arises in this context, aiming to reduce transportation costs, increase customer satisfaction, among many other objectives (Irnich; Toth; Vigo, 2014). Although this problem has been studied for over 60 years in the literature (Dantzig; Ramser, 1959; Balinski; Quandt, 1964), its practical relevance and complexity makes it a relevant subject even nowadays (Toth; Vigo, 2014; Laporte; Toth; Vigo, 2013). Many studies usually focus on the development of mathematical models and algorithms that are more efficient and/or closer to real-life situations by considering, for example, new types of constraints (Irnich; Toth; Vigo, 2014) and parameter variability (Sungur; Ordonez; Dessouky, 2008).

The VRP consists of a set of customers that must be serviced by a fleet of vehicles available in a depot. The main objective of this problem is to determine the optimal set of routes, usually minimizing costs, in order to service every customer. In these routes, each vehicle must depart from the depot, service its assigned customers, and then return to the origin. The most traditional variant of the VRP is the capacitated VRP (CVRP) in which the vehicles have a limited capacity (Irnich; Toth; Vigo, 2014) and each costumer's demand uses part of this capacity. A common extension of the CVRP is the VRP with time windows (VRPTW), in which we are not only concerned with capacity of the vehicle but also with customer time windows, the time intervals that restrict the exact time at which a vehicle can visit its assigned customers (Desaulniers; Madsen; Ropke, 2014). If the vehicle arrives earlier than the opening of a customer's time window, it must wait until this time to start the service, whereas the vehicle must not arrive after the closing of the time window.

Another relevant subject in the VRP literature is the existence of uncertainty in parameters, which has gained increased attention in the last years. In most real-life applications, the majority of parameters, such as travel times and demand, are not completely known during the planning phase (Ordonez, 2010). Despite this observation, most works in literature consider the parameters as deterministic values, which is unfortunate as optimal solutions based on these values are likely to become infeasible in practice (Oyola; Arntzen; Woodruff, 2016a; Gendreau; Jabali; Rei, 2016; De La Vega; Munari; Morabito, 2018). There are several approaches in the literature that seek to incorporate these uncertainties into models and solution methods (Ordonez, 2010). In particular, we are interested in the Robust Optimization (RO) approach, which allows us to provide routes that are more realistic and have higher chances of remaining feasible when executed, using knowledge from their maximum deviation (Ben-Tal; Nemirovski, 1999; Bertsimas; Sim, 2004). The main advantage of this approach over other common ones is that it does not require the choice of a probabilistic distribution to model the uncertain parameters, as such requirement is usually hard to be accurately done in practice. To avoid creating overly conservative solutions, where every uncertain parameter would assume its worst case, many strategies have been proposed to define uncertainty sets with controlled robustness, such as limiting the number of parameters attaining their worst case in a route (Bertsimas; Sim, 2004) or limiting the total deviation on it (Gounaris; Wiesemann; Floudas, 2013).

In this dissertation, we initially focus on the robust counterparts of a traditional variant of the VRP, namely the robust VRPTW (RVRPTW), in which we consider uncertainty on demand and travel time parameters. Then, we address robust approaches for a real-life variant motivated by the case of an airline company (Munari; Alvarez, 2019), which is hereafter referred to as the robust aircraft routing problem (RARP). The addressed aircraft routing problem is motivated by the real-life case of a company that offers on-demand air transportation services. These services typically arise in air taxi and fractional ownership contract companies (Yang et al., 2008; Yao et al., 2008; Zwan; Wils; Ghijs, 2011). In these companies, unlike the traditional ones, the customer defines the origin and destination of flights, departure times and the aircraft type that has to be used. The company must ensure that the desired aircraft will be available in the right place on time. Additionally, the company is responsible for the maintenance of the aircraft and for scheduling the appropriate crew members. Therefore, additional requirements should be considered when modelling this system in comparison to the traditional VRP variants, such as: fleet heterogeneity, different types of request (customers' and maintenance's ones) and crew requisites. To the best of our knowledge, there are no RO models fully addressing this application in the literature so far. By reducing this gap, we have the potential of helping the company's decision-making process in a context of travel time uncertainty.

The application of different uncertainty sets in solution methods oftentimes results in solutions with different cost and robustness levels (Subramanyam; Repoussis; Gounaris, 2020). For a decision maker stand-point, having these different characteristics is attractive, since they have more options to pick, and may choose the one that best suits their strategy. Nonetheless, the use of different uncertainty sets has been little explored in the context of travel time variability. While there are works that study different uncertainty sets for VRP under variability in demand (Gounaris; Wiesemann; Floudas, 2013; Subramanyam; Repoussis; Gounaris, 2020), all RVRPTW work under travel time variability have considered only the cardinality-constrained uncertainty set so far (Lee; Lee; Park, 2012; Agra et al., 2012; Munari et al., 2019). Recently, Bartolini et al. (2021) proposed a column generation algorithm that considers uncertainty on travel times with a knapsack uncertainty set for the robust traveling salesman problem with time windows, a variant of the RVRPTW with a single vehicle. Yet, there are still no compact formulations for the RVRPTW under uncertainty on times.

Thus, the central objective of this work is to create new RO models and solution methods, for traditional and practical variants, considering different types of uncertainty sets. It is worth noting that we found no compact RO model in the VRPTW literature that uses an uncertainty set other than the cardinality uncertainty set proposed by Bertsimas and Sim (2004). Therefore the main contributions of this work are:

- We propose a new compact commodity flow formulation for the deterministic and robust VRP with time windows (VRPTW). We are not aware of any compact formulation for this variant in which both load and time propagation are modeled based on commodity flow constraints, which are known for having stronger linear relaxation than those using MTZ-based constraints;

- We extend both the literature and the developed formulations to consider different uncertainty sets, namely the single and multiple knapsack uncertainty sets, which were not yet implemented in a context of VRPTW under uncertainty of travel times;
- We adapt the developed approaches to address the RARP, which is motivated by a real-life case of an on-demand air transportation company.

The remainder of this document is structured as follows. Chapter 2 presents the fundamental theoretical aspects for this work and the literature review. In Chapter 3, we present the new RO models we developed for the RVRPTW with different uncertainty sets and their computational studies. In Chapter 4, we study and develop models for a on-demand air transportation company. Finally, in Chapter 5, we present some concluding remarks.

2 Foundations and Literature Review

In this chapter, we review the most relevant models found in literature for the Robust Capacitated Vehicle Routing Problem (RCVRP) and the Robust Vehicle Routing Problem with Time Windows (RVRPTW). We describe both the traditional way to obtain a robust counterpart from a deterministic formulation, namely using the dualization scheme, which is widely used in the literature for VRP variants (Bertsimas; Sim, 2003; Agra et al., 2012; Gounaris; Wiesemann; Floudas, 2013); and a recently introduced way that is based on the linearization of recursive equations, proposed by Munari et al. (2019). Before that, to better understand the robust models, we will first present different uncertainty sets that can be used to represent the uncertain parameters.

The remainder of this chapter is structured as follows. In Section 2.1, we briefly present two-index vehicle flow formulations for the traditional CVRP and the VRPTW, as they are used to derive some robust formulations presented in the following sections. In Section 2.2, we review the concept of RO that is adopted in this work and present different uncertainty sets existent in literature. Section 2.3 describes RO models for variants of the VRP in literature. Lastly, other relevant works on RO for VRP are summarized in Section 2.4.

2.1 Deterministic formulations of traditional VRPs

We present the traditional two-index vehicle flow formulation for the CVRP based on Miller-Tucker-Zemlin (MTZ) constraints (Miller; Tucker; Zemlin, 1960). This particular formulation serves as the basis for the application of a variety of RO approaches in literature (Gounaris; Wiesemann; Floudas, 2013; Agra et al., 2013; Munari et al., 2019) and is also used for the development of models in this work. The relevant sets and parameters for this model are:

- N: set of nodes;
- N^*: set of customer nodes;
- A: set of arcs;
- K: set of vehicles;
- n: number of customer nodes;
- d_i: demand from customer i, $i = 1, \ldots, n$. We assume that $d_i \leq Q$, in order to make customer service feasible;
- Q: maximum load a vehicle can carry;
- c_{ij}: traveling costs between nodes i and j.

The problem can be represented by a network in which nodes numbered from 1 to n correspond to customers. If it is possible to transport products from a node i to another node

j, then these nodes are connected by an arc denoted as $(i, j) \in A$. To simplify the model and without any loss of generality, the depot is represented by two nodes, 0 and $n + 1$, where the former represents the departing depot and the latter represents the arrival depot. This modeling technique aims to simplify the formulation and does not require a physical separation between the starting and ending points of vehicles. The decision variables used in the model are the following, for all $(i, j) \in A$ and $j = 0, 1, \ldots, n$:

- $x_{ij} = \begin{cases} 1, & \text{if nodes } i \text{ and } j \text{ are visited consecutively;} \\ 0, & \text{otherwise.} \end{cases}$
- u_j: the total load in the vehicle that visits node j, right after it serves this node;

Using the defined parameters and decision variables, we have the following model for the problem:

$$\min \quad \sum_{i \in N} \sum_{i \in N} c_{ij} x_{ij} \tag{2.1.1}$$

$$\text{s.t.} \quad \sum_{\substack{j \in N \\ j \neq i}} x_{ij} = 1, \qquad i \in N^*, \tag{2.1.2}$$

$$\sum_{\substack{i \in N \\ i \neq h}} x_{ih} = \sum_{\substack{j \in N \\ j \neq h}} x_{hj}, \qquad h \in N^*, \tag{2.1.3}$$

$$\sum_{\substack{j \in N \\ j \neq 0}} x_{0j} \leq |K|, \tag{2.1.4}$$

$$u_j \geq u_i + d_j - Q(1 - x_{ij}), \quad i, j \in N, i \neq j, i < n + 1, j > 0, \tag{2.1.5}$$

$$u_j \leq Q, \qquad j \in N, \tag{2.1.6}$$

$$u_j \geq 0, \qquad j \in N. \tag{2.1.7}$$

$$x_{ij} \in \{0, 1\}, \qquad i, j \in N. \tag{2.1.8}$$

The objective function (2.1.1) seeks to minimize the total transportation cost, which corresponds to the sum of the travel costs of all arcs used by the vehicles. Constraints (2.1.2) ensure that each node i is visited by a single vehicle. Constraints (2.1.3) ensure that there can be a vehicle departing from node i, only if there is a vehicle arriving to this node. Constraints (2.1.4) ensure that at most $|K|$ vehicles leave from the depart depot. Constraints (2.1.5) compute the load being carried on a vehicle at each node and forbid subtours in the vehicle flow. To ensure that the capacity of each vehicle is not exceeded, we use constraints (2.1.6). Finally, constraints (2.1.7)-(2.1.8) guarantee the domain of the decision variables. It is worth noting that many other models were established to address and extend this problem, either to incorporate new types of practical characteristics or to obtain a formulation with improved computational performance (Letchford; Salazar-González, 2006; Letchford; Salazar-González, 2015; Desaulniers; Madsen; Ropke, 2014). Among the most common characteristics added are the time windows constraints, addressed in the remainder of this section.

The VRPTW extends the CVRP by allowing customers to specify time windows in which their service can start, and thus it can be more consistent with reality (Desaulniers; Madsen; Ropke, 2014). It presents intra-route constraints that require, in addition to the requirements considered by the CVRP, that each customer i must be visited within a time window $[a_i, b_i]$. The following set of additional parameters are required to represent the problem:

- a_i: earliest time it is possible to start the service on node i;
- b_i: latest time it is possible to start the service on node i;
- t_{ij}: travelling time between the nodes i and j;
- s_i: service time in node i.

It is important to note that for the arrival and departure depots, the opening times are usually $a_0 = a_{n+1} = 0$. In addition, when $a_i = 0$, for all $i \in N$, the problem is known as the VRP with deadlines (Lee; Lee; Park, 2012).

In compact models, one of the most common strategies for adding constraints related to time windows is to use an approach similar to the MTZ constraints (Miller; Tucker; Zemlin, 1960). To adapt the model (2.1.1)–(2.1.7) so that it considers time windows, we further define the following decision variables:

$$w_i : \text{ earliest time it is possible to start the service on node } i.$$

Thus, the following constraints can be added to model (2.1.1)–(2.1.7):

$$w_j \geq w_i + t_{ij} + s_i - M(1 - x_{ij}), \quad i, j \in N, i \neq j, i < n+1, j > 0, \tag{2.1.9}$$

$$a_i \leq w_i \leq b_i, \quad i \in N. \tag{2.1.10}$$

Constraints (2.1.9) guarantee that the service time on any node j will have a lower bound composed by the arrival time at the previous node i plus the service time in i and the travel time between i and j. A big-M parameter is used to inactivate this constraint if the arc (i, j) is not used. Constraints (2.1.10) ensure that every customer i is not served after b_i. If the vehicle arrives before the lower bound of the time window, it must wait until a_i to start the service. These constraints also guarantee the non-negative domain of the variable w_i since every a_i is greater than or equal to zero.

2.2 Incorporating uncertainty via Robust Optimization

Robust Optimization (RO) is an approach for modeling and solving mathematical programming problems in which parameters are uncertain during the planning phase and are only known during the implementation phase (Oyola; Arntzen; Woodruff, 2016b). There are other approaches that also incorporate uncertainty, such as stochastic programming with chance-constraints, the stochastic programming with recourse and the scenario-based formulations with recourse. The first uses the cumulative probabilistic distribution from the uncertain parameter, such as demand or travel time, to determine the value that should be used for this parameter to ensure the feasibility of solution in $\alpha\%$ of times. The second allows a recourse action, e.g. the vehicle returns to the warehouse to restock in case the demand is greater than expected, every time the solution would become infeasible, and evaluates the total expected costs of such actions. Both stochastic programming approaches require the choice of probability distributions to model the uncertain parameters, which is something that can be complex in real-life situations. The recourse approach may also use scenario-based formulations, in which the decision-maker generates a finite set of scenarios, often using the underlying probability of the parameters, and solves a first-stage problem that generates routes that are feasible for all scenarios simultaneously. Then, in the second-stage, the recourse updates the routes to meet each realization. The main challenge of this approach is the scenario-generation process, as usually real-world systems do not have all theoretical requirements for using the available methods.

One perspective of robust optimization seeks for protecting solutions against the worst-case variation of the input data. Yet, "worst-case" is problem-dependent. One very popular way to perceive worst-case is the realization of uncertain parameters that deteriorates the objective function the most. This approach then focus on providing solutions that are feasible for any possible realization of the input data in a worst-case perspective. Unlike the stochastic programming approaches, knowledge over the probabilistic distribution of the uncertain parameter is not always necessary. While some RO approaches such as the Distributionally RO may require some statistical data on the uncertain parameters, such as mean and variance (Rahimian; Mehrotra, 2019), most standard approaches do not need this information. In fact, several works in the RO literature require only an estimate of the worst-case values of the uncertain parameters (Agra et al., 2012; Munari et al., 2019; Subramanyam; Repoussis; Gounaris, 2020).

A simple strategy to introduce RO paradigms into a solution method would be by solving the problem assuming that all parameters attain their worst-case value simultaneously. This would result in a solution that accounts for any possible data realization. From the perspective of applications in VRP problems, opting for this strategy usually means getting a solution that is feasible for any data realization. However, although this solution is considered "safe" from becoming infeasible, its costs may be prohibitive. From the decision maker's point of view, it may be preferable to choose a solution that can balance cost and "safety". To create more balanced solutions, authors have proposed different uncertainty sets to represent the uncertain parameters realization whose size can be controlled by the decision maker, thus possibly finding a solution better aligned to their strategy. For example, one of the most traditional uncertainty sets studied in literature is the cardinality constrained set (Bertsimas; Sim, 2004), in which the decision maker chooses the number of parameters that can simultaneously attain their worst-case values. Decreasing the budget size resulting in a cheaper, albeit less-safe, solution. Conversely, taking higher uncertainty budgets ensures more conservative solutions. Other uncertainty sets were also studied in the RO literature, and their particularities influence on how the problems are modeled. Thus, it is important to understand these sets before formulating models, which is why we introduce them in the next topic.

2.2.1 Uncertainty sets

2.2.1.1 Cardinality constrained set

The cardinality constrained set works with a budget, usually represented by Γ, which can be understood as the threshold of the total scaled variation of the uncertain parameters. Particularly, if Γ is integer we can interpret it as the number of parameters that can simultaneously attain their worst-case value in a route (Bertsimas; Sim, 2004). As in most works on RVRP, we assume that the worst-case value of the demand and travel time parameters occur when they assume their highest possible value, since this impairs the feasibility of the route. Then, a solution is considered robust feasible for a determined budget Γ, if it is feasible when up to Γ parameters assume their worst-case value. When we are addressing uncertainty on demand, that means the solution must be feasible when the demand of any Γ customers assume their largest value. Likewise, if the uncertain parameter is the travel time between nodes, the solution is only robust if any combination of Γ arcs attain their worst-case time, while the other arcs assume their expected value.

Regarding the uncertainty on travel times, we assume that the travel time value for each arc $(i, j) \in A$, t_{ij}, ranges from its expected value $\bar{t}_{ij}$ up to $\bar{t}_{ij} + \hat{t}_{ij}$, where $\hat{t}_{ij}$ is the maximum deviation. Let γ^t_{ij}, $0 \leq \gamma^t_{ij} \leq 1$ be a random variable that represents a normalized scale deviation for travel time. Hence, the travel time can be modelled with the following expression:

$$t_{ij} = \bar{t}_{ij} + \gamma^t_{ij}\hat{t}_{ij}, \qquad\qquad 0 \leq \gamma^t_{ij} \leq 1.$$

The size of the uncertainty set is controlled by limiting the sum of the variables γ^t_{ij} to be less than a budget Γ^t. With this information, the travel time can be modeled with the following polyhedral representation (Bertsimas; Sim, 2004):

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

The uncertainty set representing the demand can be modeled with a similar strategy. In this representation, the demand d_i ranges from $\bar{d}_i$ to $\bar{d}_i + \hat{d}_i$, where $\bar{d}_i$ represents the nominal value and $\hat{d}_i$ is the maximum deviation. For a given route, this set can be written in function of γ^d_i and Γ^d as follows:

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

It is worth noting that, for an integer budget value (Γ^d), despite γ^d_i being a continuous variable, and thus having infinite numbers of possible combinations (respect. γ^t_{ij}), the worst-case scenario, i.e., the one that worsens the objective function the most for a determined budget, happens when some specific nodes/arcs assume $\gamma^d_i = 1$ (respect. $\gamma^t_{ij} = 1$), while the others assume 0 (Bertsimas; Sim, 2004). If the budget Γ is not integer, then the worst-case scenario can be found by making $\lfloor\Gamma\rfloor$ parameters attaining their worst-case value and a single node/arc assume $\gamma^d_i = 1 - \lfloor\Gamma\rfloor$ (respect. $\gamma^t_{ij} = 1 - \lfloor\Gamma\rfloor$).

2.2.1.2 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 in a route, considering one or many knapsacks (Subramanyam; Repoussis; Gounaris, 2020). Each knapsack l involves a subset of nodes or arcs, depending on which parameter we are analyzing, with its own budget of deviation Δ_l. Usually, authors in the RVRP literature (Gounaris; Wiesemann; Floudas, 2013; Subramanyam; Repoussis; Gounaris, 2020) model these knapsacks by relating the realizations to geographical regions. For example, one may divide the nodes/arcs in 4 different quadrants (NE, SE, NW, SW) and set a budget for each quadrant based on its characteristics. For instance a quadrant, with higher variability in customer's demand may have higher budget than others. Additionally, while usually there are no limitations on how the knapsacks are modeled, authors usually build them as disjoint sets, meaning that their nodes/arcs do not overlap.

Let L be the set of knapsacks and S_d^l the set of nodes in each knapsack l. We define Δ_l^d as the budget for demand deviation in knapsack $l \in L$ and Δ_l^t as the budget for travel times in knapsack $l \in L$ with S_l^t defined accordingly. Then, we can model this set for demand, U_L^d, and travel time, U_L^t, using the following expressions:

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

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

The demand for each node i, d_i, in set U_L^d ranges from its expected value $\bar{d}_i$ up to its worst-case value $\bar{d}_i + \hat{d}_i$. The sum of all realized deviations of nodes, i.e., the difference between the realized demand d_i and its expected value $\bar{d}_i$, in a given knapsack l is limited by Δ_l^d. The same is done for the travel time t_{ij}, whose realization ranges from $\bar{t}_{ij}$ and $\bar{t}_{ij} + \hat{t}_{ij}$, and the sum of all deviations is limited by Δ_l^t.

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

To the best of our knowledge, there is no paper considering the multiple knapsack uncertainty set for uncertain travel times so far in the literature. The works that use this uncertainty set all consider uncertainty on demand only. Namely, one of the uncertainty sets tested by Gounaris, Wiesemann and Floudas (2013), which proposed compact formulations the RCVRP under demand uncertainty with general polyhedral sets representation, was the multiple knapsack uncertainty set. Pessoa et al. (2020) tackled the same problem by proposing a branch-and-cut-and-price algorithm. This set, among others, was also studied by Subramanyam, Repoussis and Gounaris (2020) who proposed robust versions for node and arc exchange neighborhoods that are commonly used in local search and then incorporate them into metaheuristic algorithms for the heterogeneous VRP under demand uncertainty. It is worth noting that it is not trivial to extend these methods to consider uncertainty on travel times because they were designed based on the assumption that, if a knapsack is completely filled, it does not matter where in the route the deviation was accounted for. However, we cannot make this assumption in problems with uncertainty on travel times, as travel time deviations can be absorbed by the customer's opening time. Indeed, even after a long deviation in a given arc, the vehicle may still have to wait for the exact time the window opens. Therefore, we cannot make the same assumption as in the case with uncertain demand.

2.2.1.3 Other uncertainty sets in literature

In this subsection, we present uncertainty sets that have been explored for RO problems in the literature, but are not considered in our approaches. Note that all the previously stated uncertainty sets and some that will be presented in this topic, namely the convex set proposed by Soyster (1973) and the factor models uncertainty sets, are subsets of a general polyhedral uncertainty set. As presented by Gounaris, Wiesemann and Floudas (2013), the general representation of these sets for uncertainty on demand is given by:

$$U_W^d = \{d \in \mathbb{R}^{|N|} \mid Wd \leq h, W \in \mathbb{R}^{r \times |N|}, h \in \mathbb{R}^r\}.$$

This essentially means the uncertainty set can be represented as a polyhedron, and thus with linear equations. In this general statement, we have r constraints limiting the total value the demand can achieve, with a weight matrix (W) of size $r \times |N|$ and an upper bound parameter (h) for each constraint.

To our knowledge, Soyster (1973) was the first to consider a worst-case approach explicitly in a model. In this work, the author proposed a convex set that basically defines that every uncertain parameter assumes its worst-case value. The uncertainty set for demand is represented by:

$$U_C^d = \{d \in \mathbb{R}^{|N|} \mid d_i = \bar{d}_i + \hat{d}_i, i \in N\}.$$

This is a simple approach to incorporate uncertainty into problems, although it usually results in an excessively conservative solution, by protecting it from a scenario in which every uncertain parameter assumes its worst-case value, which in practice is highly unlikely, resulting in a high cost .

Ben-Tal and Nemirovski (1999) proposed the *ellipsoidal* uncertainty set. Unlike the one proposed by Soyster (1973), this set is able to be adjusted in order to control the robustness of solutions. We start by defining *cov* as the matrix specified by the decision maker that represents the co-variance between the demand of two nodes. With this matrix, we can represent the set as follows:

$$U_E^d = \{d \in \mathbb{R}^{|N|} \mid d_i = \bar{d}_i + cov_i^{\frac{1}{2}}\xi, i \in N, \xi \in \Xi\},$$

where $\Xi = \{\xi \in \mathbb{R}^{|N|} | \xi^T\xi \leq 1\}$. The parameters represented by this set have its value ranging inside an ellipsoidal representation centered in the expected value of the parameter. Despite being less conservative than the convex set proposed by Soyster (1973), this set suffers from non-linearity and hence results in approaches less computationally tractable (Subramanyam; Repoussis; Gounaris, 2020).

Another uncertainty set found in literature is the *factor models* set (Ceria; Stubbs, 2006; Gounaris; Wiesemann; Floudas, 2013; Subramanyam; Repoussis; Gounaris, 2020). For demand, this specific set can be represented by:

$$U_F^d = \{d \in \mathbb{R}^{|N|} \mid d_i = \bar{d}_i + \Psi\xi, i \in N, \xi \in \Xi\},$$

where $\Xi = \{\xi \in [-1,1]^F : |e^T\xi| \leq \beta F\}$, and parameters $\Psi \in \mathbb{R}^{|N|xF}$, $F \in \mathbb{R}^+$ and $\beta \in [0,1]$ are chosen by the decision maker and $e \in \mathbb{R}^F$ is a vector of ones. This set represents the deviation as a linear combination of independent factors $\xi_1, \ldots, \xi_F$ within a factor loading matrix, Ψ, that correlates these factors for the demand realization. To prevent every factor assuming its worst-case value, the constraint $|e^T\xi| \leq \beta F$ is used. The parameter β is a chosen limit for how many factors assume its maximum value. For instance, $\beta = 0$ means there will be the same number of factors with value greater than zero than factors that present negative values.

A final possible uncertainty set is the discrete set, composed by a convex hull of finite possible realizations of demands (Charris; Prins; Santos, 2015). Essentially, the decision maker creates a set of D possible realization, stipulating the demand for each node in each realization. With this information, it is possible to represent the uncertainty set as follows:

$$U_D^d = conv\{d^{(j)} | j = 1, \dots, D\},$$

where *conv* represents the convex hull of the possible realizations. One way to model this set is to use historical demand/time data as the realizations. It is important, however, to be careful with possible outliers and overfitting of the solution (Subramanyam; Repoussis; Gounaris, 2020).

Finally, it is worth noting that all sets described in this subsection can be trivially adapted to represent travel time uncertainty.

2.3 RO formulations for traditional VRPs

The use of RO to incorporate uncertainty into VRP variants is relatively new. Sungur, Ordonez and Dessouky (2008) were the first to use this paradigm to consider demand uncertainty in the CVRP. They proposed compact models based on convex hull, box, and ellipsoidal uncertainty sets. These, however, simply replaced the nominal demand of the deterministic formulation by an augmented modified demand, which resulted in overly conservative solutions when compared to the stochastic methods with chance-constrained or recourse. Lee, Lee and Park (2012) used the RO with bounded uncertainty to control the robustness of solution for the VRP with deadlines under demand and travel time uncertainties. The authors proposed a three-index formulation with MTZ constraints following the cardinality constrained uncertainty set. Due to the long computational times required to solve the model, the authors also proposed a branch-and-price algorithm to solve the problem. Even so, they were unable to solve several instances with 25 customers to optimality.

Agra et al. (2012) proposed the first RO formulation for the RVRPTW with uncertainty on travel times, considering the cardinality constrained uncertainty set. They proposed a model based on layered-graphs for a problem with heterogeneous fleet, but no capacity constraints, which was applied in a context of maritime transportation. The proposed formulation was able to solve the majority of the studied small scale instances (10 to 20 customer nodes), but failed to find feasible solutions for some instances with 20 customers. This problem was later revisited by Agra et al. (2013) who proposed two more efficient robust approaches, one extends the resource inequalities formulation by employing adjustable robust optimization and the other generalizes a path inequality formulation to consider uncertainty on travel time.

Gounaris, Wiesemann and Floudas (2013) proposed robust counterparts of several traditional formulations for the CVRP considering uncertainty on demand. 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 exemplify 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 with larger instances. Gounaris et al. (2016) proposed an adaptive memory programming metaheuristic, which presented high quality solutions, for the same class of problems using the cardinality constrained uncertainty set. Recently, Munari et al. (2019) proposed a new robust counterpart model based on the linearization of recursive equations for the RVRPTW under uncertainty on both demand and travel times considering the cardinality constrained uncertainty set. The proposed model considerably outperformed other formulations based on traditional methods (Agra et al., 2012) when using a general-purpose integer programming solver. Using this model, the authors solved the majority of the instances with 25 customers to optimality and found feasible solutions for all instances, while with other formulations no instance was solved to optimality and feasible solutions were obtained for less than 60% of the instances. In the same paper, the authors also proposed a branch-price-and-cut method based on a set partitioning formulation of the problem that was able to solve most of benchmark instances, with up to 100 customers each, to optimality.

In the remainder of this section, we present some compact models proposed for the RVRPTW in the literature. Firstly, we describe the traditional approach to create a robust counterpart model, namely the dualization scheme proposed by Bertsimas and Sim (2004), and present a few compact models that rely on this technique. Then, we present a more recent approach to obtain robust counterparts proposed by Munari et al. (2019), based on the linearization of recursive equations, which was used to derive the compact model that has currently the best overall performance on benchmark instances. It is worth noting that each of these models consider the cardinality constrained uncertainty set and, with the exception discussed in Subsection 2.3.1.3, which was designed for a generic polyhedral uncertainty set, it is not trivial to adapt them for other uncertainty sets, as their modeling strategies depend on some particularities of the cardinality constrained set.

2.3.1 Dualization-based models

We first describe how the dualization scheme can be used to obtain robust counterparts and then present some RO formulations of traditional VRPs that are obtained using this technique. This approach was firstly proposed by Bertsimas and Sim (2003) and was designed for the cardinality constrained set. It is based on the dualization of the maximization subproblem that represents the worst-case scenario for the uncertain parameters in a given constraint of the model. Before presenting this approach specifically for the VRP variants, it is important to understand how it is used on a general integer linear optimization problem.

2.3.1.1 Dualization scheme for a generic problem

We start with a general integer problem for a set of n variables and m constraints:

$$\min \quad c^T x \tag{2.3.1}$$

$$\text{s.t.} \quad Ax \leq b \tag{2.3.2}$$

$$l \leq x \leq u \tag{2.3.3}$$

$$x_i \in \mathbb{Z}, \qquad i = 1, \dots, k. \tag{2.3.4}$$

Parameter c represents a cost vector of size n; l and u are, respectively, vectors that represent the lower and upper bound for each variable; A is a coefficient matrix $m \times n$, and b is a vector of size m.

We consider that some coefficients of matrix A are uncertain, something like demand or travel time parameters on the VRP. Let K_i be the set of uncertain coefficients in the ith row of A. Using the cardinality constrained set, we assume that each entry a_{ij} such that $j \in K_i$ belongs to the interval $[\bar{a}_{ij}, \bar{a}_{ij} + \hat{a}_{ij}]$, where $\bar{a}_{ij}$ is the nominal value of the parameter and $\hat{a}_{ij}$ is its maximum deviation.

Let parameter Γ^i_a be the budget of the number of deviations in the uncertain parameters of each constraint i (row i of matrix A). We define γ^i_j as the primitive random variable that represents the fraction of the deviation with respect to the nominal value, i.e. $a_{ij} = \bar{a}_{ij} + \hat{a}_{ij}\gamma^i_j$. Let $S^i_a \subset K_i$, such that $|S^i_a| \leq \Gamma^i_a$, be the subset of coefficients for which $\gamma^i_j > 0$. Since we want to optimize the worst-case scenario, model (2.3.1)-(2.3.4) can be rewritten as follows:

$$\min \quad c'x \tag{2.3.5}$$

$$\text{s.t.} \quad \sum_j \bar{a}_{ij}x_j + \max_{S^i_a \in K_i, |S^i_a| \leq \Gamma^i_a} \left\{ \sum_{j \in S^i_a} \gamma^i_j \hat{a}_{ij}|x_j| \right\} \leq b_i, \quad i = 1, \ldots, k, \tag{2.3.6}$$

$$l \leq x \leq u, \tag{2.3.7}$$

$$x_i \in \mathbb{Z}, \qquad i = 1, \ldots, k. \tag{2.3.8}$$

Since the resulting model has now maximization subproblems in the definition of its constraints, it is no longer linear. However, Bertsimas and Sim (2004) were able to transform those subproblems into linear constraints by the dualization strategy. To better exemplify how it was made, we first rewrite the subproblem of the ith constraint as follows:

$$\max \quad \sum_j \gamma^i_j \hat{a}_{ij}|x_j|, \tag{2.3.9}$$

$$\text{s.t.} \quad \sum_j \gamma^i_j \leq \Gamma^i_a, \tag{2.3.10}$$

$$0 \leq \gamma^i_j \leq 1, \quad j. \tag{2.3.11}$$

Since this problem is feasible and bounded, by strong duality, its dual is also feasible and bounded and their objective values coincide. The dual model is as follows:

$$\min \quad \sum_j p_{ij} + \Gamma^i_a z_i, \tag{2.3.12}$$

$$\text{s.t.} \quad p_{ij} + z_i \geq \hat{a}_{ij}|x_j|, \tag{2.3.13}$$

$$p_{ij}, z_i \geq 0, \qquad j, \tag{2.3.14}$$

where variable z_i refers to constrains (2.3.10) from the primal problem, while p_{ij} refers to constraints (2.3.11). It is possible to plug the dual subproblems into model (2.3.5)-(2.3.8), resulting in the following model:

$$\min \quad c'x \tag{2.3.15}$$

$$\text{s.t.} \quad \sum_j \bar{a}_{ij}x_j + \min\left\{\Gamma^i_a z_i + \sum_j p_{ij}\right\} \leq b_i, \quad i = 1, \ldots, k, \tag{2.3.16}$$

$$z_i + p_{ij} \geq \hat{a}_{ij} y_j, \qquad i, j \tag{2.3.17}$$

$$z_i, p_{ij}, y_j \geq 0, \qquad i, j \tag{2.3.18}$$

$$-y_j \leq x_j \leq y_j, \qquad j \tag{2.3.19}$$

$$l_j \leq x_j \leq u_j, \qquad j \tag{2.3.20}$$

$$x_i \in \mathbb{Z}, \qquad i = 1, \ldots, k. \tag{2.3.21}$$

If there is a feasible solution (z_i, p_{ij}) for the minimization subproblem that satisfies (2.3.16), then the optimal solution (z_i^*, p_{ij}^*) of the corresponding subproblem also satisfies this constraint. Thus, we can drop the “min” expression in constraints (2.3.16), resulting in the following model:

$$\min \qquad c'x \tag{2.3.22}$$

$$\text{s.t.} \quad \sum_j \bar{a}_{ij} x_j + \Gamma_a^i z_i + \sum_j p_{ij} \leq b_i, \quad i = 1, \ldots, k, \tag{2.3.23}$$

$$z_i + p_{ij} \geq \hat{a}_{ij} y_j, \qquad i, j \tag{2.3.24}$$

$$z_i, p_{ij}, y_j \geq 0, \qquad i, j \tag{2.3.25}$$

$$-y_j \leq x_j \leq y_j, \qquad j \tag{2.3.26}$$

$$l_j \leq x_j \leq u_j, \qquad j \tag{2.3.27}$$

$$x_i \in \mathbb{Z}, \qquad i = 1, \ldots, k. \tag{2.3.28}$$

This approach defines one uncertainty set for each constraint of the problem. Indeed, each subproblem has its own budget of uncertainty and they are independent from each other. Hence, the subproblem concerns only the deviations of a single constraint. If in the deterministic formulation, the uncertain parameters are not all in the same constraint but their realizations must belong to a common uncertainty set, we need to reformulate the model in order to have all these parameters in a single constraint. For instance, both the standard formulations for the CVRP and VRPTW presented in Section 2.1 need to be changed for this approach. Indeed, each customer demand appears in a different constraint of those models and, hence, it is not possible to straightforwardly impose that they belong to the same uncertainty set. In the following subsection we exemplify how some authors adapted the traditional formulations in order to use the dualization approach.

2.3.1.2 RVRPTW formulation based on the layered formulation

Agra et al. (2012) is an example of a RO formulation for the VRPTW that uses the dualization approach to incorporate uncertainty into the compact model. To use this approach, the authors reformulated the problem using a layered formulation. They considered uncertainty in travel times only, based on a maritime transportation context with no capacity constraints, and hence the authors did not consider uncertainty on demand. The proposed layered formulation is based on the creation of 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 that the vehicle is at after visiting $l-1$ nodes on a path from the origin. For each layer, there is a set of possible arcs, α, that the vehicle can take, based on the feasibility of the time windows' constraints and the nodes previously visited on the route. Figure 1 was extracted from Agra et al. (2012) and visually exemplifies the layered representation.

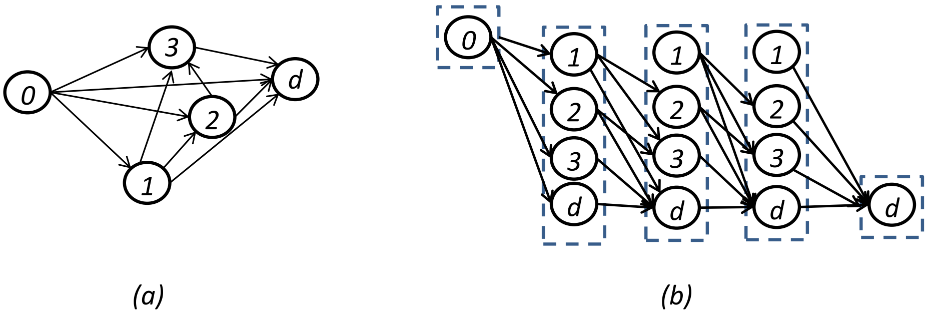


Figure 1 – (a) Network for a commodity k; (b) its representation in layers. Source: (Agra et al., 2012)

Before discussing how uncertainty in the parameters is taken into account in the model, it is important to understand the decision variables used on the deterministic problem:

$$x_{ij}^{k} = \begin{cases} 1, & \text{if vehicle } k \in K \text{ visits node } i \in N \text{ and } j \in N \text{ consecutively,} \\ 0, & \text{otherwise;} \end{cases}$$

$$z_{ij}^{kl} = \begin{cases} 1, & \text{if vehicle } k \in K \text{ visits node } i \in N \text{ on the position } l \in L \text{ on the way to node } j \in N, \\ 0, & \text{otherwise.} \end{cases}$$

For the robust formulation, the travel time t_{ij} between nodes i and j can range from its average $\bar{t}_{ij}$ up to the value $\bar{t}_{ij} + \hat{t}_{ij}$, where $\hat{t}_{ij}$ is the maximum possible variation. In the model, the constraints associated with the uncertain parameter are:

$$\sum_{(i,j)\in A:(j,i,l_1)\in\alpha} z_{ij}^{kl_1} a_i + \sum_{l=l_1,\ldots,l_2-1} \sum_{(i,j)\in A:(j,i,l)\in\alpha} z_{ij}^{kl} t_{ij}^{k} \leq \sum_{(i,j)\in A:(j,i,l_2)\in\alpha} z_{ji}^{kl_2} b_j, \qquad 1 \leq l_1 \leq l_2 \leq L, k \in K, \quad (2.3.29)$$

These constraints compute the accumulated time in a route in the left hand side and ensure this time does not exceed the closing time window. Notice that, differently from the two-index vehicle flow formulation, constraints (2.3.29) of the layer formulation involve all the travel times of the arcs that are traversed in a route that belongs to a feasible solution of the model.

To ensure the feasibility of constraints (2.3.29) for all possible realizations inside the uncertainty set, the worst-case scenario of the uncertain parameters within the uncertainty set must be analyzed. Given the type of the constraint, it is possible to rewrite it as follows:

$$\sum_{j:(j,i,l_1)\in\alpha} z_{ij}^{kl_1} a_i + \sum_{l=l_1,\ldots,l_2-1} \sum_{(i,j)\in A} z_{ij}^{kl} \bar{t}_{ij} + \max_{S^t \subset A, |S^t| \leq \Gamma^t} \sum_{l=l_1,\ldots,l_2-1} \sum_{(i,j)\in A} z_{ij}^{kl} \gamma_{ij} \hat{t}_{ij} \leq \sum_{j:(j,i,l_2)\in\alpha} z_{ji}^{kl_2} b_j, \qquad 1 \leq l_1 \leq l_2 \leq L, k \in K, \quad (2.3.30)$$

where γ_{ij} represents the fraction of the maximum deviation of the time parameter considered in the worst-case scenario and $S^t \subset A$ is the subset of arcs (i,j) for which $\gamma_{ij} > 0$, such that $|S^t| \leq \Gamma^t$. Thus, for each l_1, l_2 and $k \in K$ such that $1 \leq l_1 \leq l_2 \leq L$, the subproblem defined by the internal maximization involving the sum of a product of variables in this constraint can be rewritten as:

$$\max \sum_{l=l_1,\ldots,l_2-1} \sum_{(i,j)\in A} z_{ij}^{kl} \gamma_{ij} \hat{t}_{ij} \qquad (2.3.31)$$

$$\text{s.t.} \sum_{(i,j)\in A} \gamma_{ij} \leq \Gamma^t, \tag{2.3.32}$$

$$0 \leq \gamma_{ij} \leq 1, \qquad (i,j) \in A. \qquad . \tag{2.3.33}$$

The objective function (2.3.31) aims to find the maximum total deviation. Constraints (2.3.32) enforce that the total deviation cannot exceed the budget Γ^t. Constraints (2.3.33) guarantee that the deviation in each arc is non-negative and limited by 100% of the maximum deviation.

Using the dualization scheme, this subproblem can be replaced by its dual problem given as follows, since both result in the same optimal value.

$$\min \qquad \Gamma^t v^{kl_1l_2} + \sum_{(i,j)\in A} u_{ij}^{kl_1l_2}, \tag{2.3.34}$$

$$\text{s.t. } u_{ij}^{kl_1l_2} + v^{kl_1l_2} \geq \hat{t}_{ij} + \sum_{l_1 \leq l < l_2} z_{ij}^{kl}, \quad (i,j) \in A. \tag{2.3.35}$$

$$v^{kl_1l_2} \geq 0, \tag{2.3.36}$$

$$u_{ij}^{kl_1l_2} \geq 0, \tag{2.3.37}$$

Variable $v^{kl_1l_2}$ is the dual variable associated with constraints (2.3.32), while the dual variable $u_{ij}^{kl_1l_2}$ is associated to (2.3.33). We can now replace the maximization subproblem in (2.3.30) with the objective function (2.3.34) from the dual subproblem and the additional constraints (2.3.35)–(2.3.37). The resulting robust counterpart is as follows:

$$\min \sum_{k\in K} \sum_{(i,j)\in A} c_{ij}^k x_{ij}^k, \tag{2.3.38}$$

$$\text{s.t.} \sum_{k\in K} \sum_{\substack{j\in N:\\(i,j)\in A}} x_{ij}^k = 1, \qquad i = 1\ldots, n, \tag{2.3.39}$$

$$\sum_{j:(j,i,l-1)\in\alpha} z_{ji}^{kl-1} - \sum_{j:(j,i,l)\in\alpha} z_{ji}^{kl} = \begin{cases} -1, & \text{if } i = 0, \\ 1, & \text{if } i = d \text{ and } l = L, \\ 0, & \text{otherwise.} \end{cases}$$
$$1 \leq l \leq L, i \in N, k \in K, \tag{2.3.40}$$

$$\sum_{j:(j,i,l)\in\alpha} z_{ji}^{kl} = x_{ij}^k, \qquad (i,j) \in A, k \in K, \tag{2.3.41}$$

$$\sum_{j:(j,i,l_1)\in\alpha} z_{ij}^{kl_1} a_i + \sum_{(i,j)\in A} \bar{t}_{ij} \sum_{l=l_1,\ldots,l_2-1} z_{ij}^{kl} + \Gamma^t v^{kl_1l_2} + \sum_{(i,j)\in A} u_{ij}^{kl_1l_2} \leq \sum_{j:(j,i,l_2)\in\alpha} z_{ji}^{kl_2} b_j,$$
$$1 \leq l_1 \leq l_2 \leq L, k \in K, \tag{2.3.42}$$

$$u_{ij}^{kl_1l_2} + v^{kl_1l_2} \geq \hat{t}_{ij} + \sum_{l_1 \leq l < l_2} z_{ij}^{kl}, \qquad (i,j) \in A, l_1 \leq l < l_2 < L, \tag{2.3.43}$$

$$v^{kl_1l_2} \geq 0 \qquad 1 \leq l_1 < l_2 \leq L, k \in K, \tag{2.3.44}$$

$$u_{ij}^{kl_1l_2} \geq 0 \qquad 1 \leq l_1 < l_2 \leq L, k \in K, (i,j) \in A, \tag{2.3.45}$$

$$x_{ij}^k \in \{0,1\}, \qquad k \in K, (i,j) \in A, \tag{2.3.46}$$

$$z_{ij}^{kl} \in \{0,1\}, \qquad k \in K, (i,j,l) \in \alpha. \tag{2.3.47}$$

The objective function (2.3.38) consists of minimizing the total travel costs. Constraints (2.3.39) ensure that every customer node is visited exactly once. Constraints (2.3.40) guarantee the flow in the layers, which comes from the condition of the vehicle k departing only once from the node i, making the difference between the number of arcs arriving at the starting node 0 and the number of arcs departing from it equal to -1; the vehicle k must also depart from every customer node that it visits, thus making the difference equal to 0; finally, the arriving depot node, $n+1$, should be visited only once and the vehicle will not depart from it, resulting in a difference of 1. Constraints (2.3.41) relate the variables z_{ij}^{kl} and x_{ij}^{k}. Constraints (2.3.42) and (2.3.43) ensure that customer time windows are respected for any possible realization of the uncertain travel times. Constraints (2.3.44) and (2.3.45) set the domain of the continuous variables, and constraints (2.3.46) and (2.3.47) set the domain for the binary variables.

Agra et al. (2012) used the described formulation to solve relatively small-scale instances with 10 to 20 cargoes and 1 to 5 vehicles. They were all solved to optimality when using an additional approach that reduces the maximum number of layers, with average computational times of 225.29 seconds to solve the model, plus 107.64 seconds to run the layer-reduction algorithm. The computational time of the reduction algorithm considerably increased depending on the number of vehicles in the problem, ranging from 6.09 to 334 seconds, but still tends to be more efficient than not using it. Notably, although the robust counterpart of the problem can be computationally tractable, it is considerably more difficult to solve than the deterministic formulation, and can take more than 10 times the time to solve, denoting a limited practicability (Chen; Sim; Xiong, 2020) of the method. In fact, Munari et al. (2019) tested the same formulation in instances from Solomon's benchmark with 25 customers in order to compare it with their own formulation. They noted that the layered formulation was not able to prove optimality for any instance and found feasible solutions or proved infeasibility for only 59.48% of them within the time limit of 3600 seconds.

2.3.1.3 RCVRP formulation using MTZ constraints

The formulations proposed by Gounaris, Wiesemann and Floudas (2013) for the RCVRP use the dualization approach as well, but consider general polyhedral uncertainty sets. In this subsection, we present their formulation based on MTZ constraints, considering specifically the cardinality constrained uncertainty set, which is a special case of the polyhedral uncertainty set.

Starting from the standard deterministic two-index model with MTZ constraints for CVRP (Toth; Vigo, 2002), presented in Section 2.1, it is possible to replace the product load decision variable u_i with $\sum_{l \in N} v_{il} d_l$. This allows all demand parameters to be in the same constraint, making it possible to apply the dualization approach. The proposed model, then, results in:

$$\min \quad \sum_{(i,j)\in A} c_{ij} x_{ij}, \tag{2.3.48}$$

$$\text{s.t.} \quad \sum_{\substack{j=1 \\ j\neq i}}^{n+1} x_{ij} = 1, \qquad i \in N^*, \tag{2.3.49}$$

$$\sum_{\substack{i=0 \\ i\neq h}}^{n} x_{ih} = \sum_{\substack{j=1 \\ j\neq h}}^{n+1} x_{hj}, \qquad h \in N^*, \tag{2.3.50}$$

$$\sum_{l=1}^{n}(v_{jl}-v_{il})d_l+Q(1-x_{ij})\geq d_j,\quad i,j\in N^*,i\neq j \tag{2.3.51}$$

$$\sum_{l=1}^{n}(v_{il})d_l\geq d_i,\quad i\in N^*, \tag{2.3.52}$$

$$\textstyle\sum_{l=1}^{n}(v_{il})d_l\leq Q,\quad i\in N, \tag{2.3.53}$$

$$v_{il}\geq 0,\quad (i,l)\in A, \tag{2.3.54}$$

$$x_{ij}\in\{0,1\},\quad (i,j)\in A, \tag{2.3.55}$$

In this model, d_l is an uncertain parameter that belongs to the set $[\bar{d}_l,\bar{d}_l+\hat{d}_l]$ where $\bar{d}_l$ is the expected value of d_l, while $\hat{d}_l$ is the maximum variation that this demand can have. Then, following Bertsimas and Sim (2004) steps, we define the primitive random variable γ_i^d that represents the fraction of the maximum variation that the demand of the node presents. The sum of all these variables must be limited by the budget Γ^d. Then, constraints (2.3.51) can be rewritten for the robust problem as:

$$\sum_{l=1}^{n}(v_{jl}-v_{il})\bar{d}_l+\max_{\sum_{l\in N^*}\gamma_l^d\leq\Gamma^d}(\sum_{l=1}^{n}((v_{jl}-v_{il})(\hat{d}_l\gamma_l^d))-(\bar{d}_j+\hat{d}_j\gamma_j^d))+Q(1-x_{ij})\geq 0,i,j\in N^*,i\neq j, \tag{2.3.56}$$

where N^* is the set of all customer nodes. Then, the maximization subproblem can be represented by:

$$\max\sum_{l=1}^{n}(v_{jl}-v_{il})(\hat{d}_l\gamma_l^d)-(\hat{d}_j\gamma_j^d) \tag{2.3.57}$$

$$0\leq\gamma_l^d\leq 1,\quad l\in N^*, \tag{2.3.58}$$

$$\sum_{l\in N^*}\gamma_l^d\leq\Gamma^d. \tag{2.3.59}$$

The dual of this subproblem is:

$$\min\sum_{l=1}^{n}\alpha_{lk}+\Gamma^d\mu_k \tag{2.3.60}$$

$$\alpha_{lk}+\mu_k\geq\hat{d}_l(v_{jl}-v_{il}-1),\quad i,j,l\in N^*,i\neq j,l=j,k\in K, \tag{2.3.61}$$

$$\alpha_{lk}+\mu_k\leq\hat{d}_l(v_{jl}-v_{il}),\quad i,j,l\in N^*,i\neq j,l\neq j,k\in K. \tag{2.3.62}$$

Finally, the robust counterpart is obtained by replacing constraints (2.3.51) with:

$$\bar{d}_j+\sum_{l=1}^{n}\alpha_{lk}+\Gamma^d\mu_k-\sum_{l=1}^{n}(v_{jl}-v_{il})\bar{d}_l\leq Q(1-x_{ij}),\quad i,j\in N^*,i\neq j,k\in K, \tag{2.3.63}$$

$$\alpha_{lk}+\mu_k\geq\hat{d}_l(v_{jl}-v_{il}-1),\quad i,j,l\in N^*,i\neq j,l=j,k\in K, \tag{2.3.64}$$

$$\alpha_{lk}+\mu_k\geq\hat{d}_l(v_{jl}-v_{il}),\quad i,j,l\in N^*,i\neq j,l\neq j,k\in K. \tag{2.3.65}$$

Similar steps need to be taken to constraints (2.3.52) and (2.3.53) – both constraints have the uncertain parameter on them, and for this reason we also need their robust counterpart. Constraints (2.3.52) are then represented by:

$$\sum_{l=1}^{n}(v_{il})\bar{d}_l+\max(\sum_{l=1}^{n}(v_{il}\hat{d}_l)\gamma_l^d-(\hat{d}_i)\gamma_i^d)-\bar{d}_i\geq 0,i\in N^*.$$

After dualizing the inner subproblem, we obtain:

$$\sum_{l=1}^{n}(v_{il})\bar{d}_l + \sum_{l=1}^{n}\alpha'_{lk} + \Gamma^d\mu'_k - \bar{d}_i \geq 0, \quad i \in N^*, i \neq j, \tag{2.3.66}$$

$$\alpha'_{lk} + \mu'_k \geq \hat{d}_l(v_{il} - 1), \quad i,j,l \in N^*, i \neq j, l = j, k \in K, \tag{2.3.67}$$

$$\alpha'_{lk} + \mu'_k \geq \hat{d}_l v_{il}, \quad i,j,l \in N^*, i \neq j, l \neq j, k \in K, \tag{2.3.68}$$

Note that we needed to create additional sets of variables which represent the dual variables from this subproblem, α'_{lk} and μ'_k, for these constraints.

Lastly, we also transformed the robust counterpart of constraints (2.3.53):

$$\sum_{l=1}^{n}(v_{il})\bar{d}_l + \max\sum_{l=1}^{n}(v_{il}\hat{d}_l)\gamma_l^d \leq Q, i \in N,$$

into linear constraints by applying the same dualization approach, resulting in the following set of constraints:

$$\sum_{l=1}^{n}\alpha''_{lk} + \Gamma^d\mu''_k + \sum_{l=1}^{n}(v_{il})d_l \leq Q, \quad i \in N^*, k \in K, \tag{2.3.69}$$

$$\alpha''_{lk} + \mu''_k \geq \hat{d}_l v_{il}, \quad i,j,l \in N^*, i \neq j, k \in K, \tag{2.3.70}$$

The new variables α''_{lk} and μ''_k are also dual variables from the maximization sub-problem of the original robust constraint. Thus, model (2.3.71)-(2.3.85) is obtained, which represents the full robust counterpart of the problem.

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{2.3.71}$$

$$\text{s.t.} \sum_{\substack{j=1\\ j\neq i}}^{n+1} x_{ij} = 1, \quad i \in N^*, \tag{2.3.72}$$

$$\sum_{\substack{i=0\\ i\neq h}}^{n} x_{ih} = \sum_{\substack{j=1\\ j\neq h}}^{n+1} x_{hj}, \quad h \in N^*, \tag{2.3.73}$$

$$\bar{d}_j + \sum_{l=1}^{n}\alpha_{lk} + \Gamma^d\mu_k - \sum_{l=1}^{n}(v_{jl} - v_{il})\bar{d}_l \leq Q(1 - x_{ij}), \quad i,j \in N^*, i \neq j, k \in K, \tag{2.3.74}$$

$$\alpha_{lk} + \mu_k \geq \hat{d}_l(v_{jl} - v_{il} - 1), \quad i,j,l \in N^*, i \neq j, l = j, k \in K \tag{2.3.75}$$

$$\alpha_{lk} + \mu_k \geq \hat{d}_l(v_{jl} - v_{il}), \quad i,j,l \in N^*, i \neq j, l \neq j, k \in K \tag{2.3.76}$$

$$\sum_{l=1}^{n}(v_{il})\bar{d}_l + \sum_{l=1}^{n}\alpha'_{lk} + \Gamma^d\mu'_k - \bar{d}_i \geq 0, \quad i \in N^*, i \neq j, \tag{2.3.77}$$

$$\alpha'_{lk} + \mu'_k \geq \hat{d}_l(v_{il} - 1), \quad i,j,l \in N^*, i \neq j, l = j, k \in K, \tag{2.3.78}$$

$$\alpha'_{lk} + \mu'_k \geq \hat{d}_l v_{il}, \quad i,j,l \in N^*, i \neq j, l \neq j, k \in K, \tag{2.3.79}$$

$$\sum_{l=1}^{n}\alpha''_{lk} + \Gamma^d\mu''_k + \sum_{l=1}^{n}(v_{il})d_l \leq Q, \quad i \in N^*, k \in K, \tag{2.3.80}$$

$$\alpha''_{lk} + \mu''_k \geq \hat{d}_l v_{il}, \quad i,j,l \in N^*, i \neq j, k \in K, \tag{2.3.81}$$

$$x_{ij} \in \{0,1\}, \quad (i,j) \in A, \tag{2.3.82}$$

$$v_{ij} \geq 0, \qquad (i,j) \in A, \quad (2.3.83)$$

$$\alpha_{ik} \geq 0, \qquad i \in N, k \in K, \quad (2.3.84)$$

$$\mu_k \geq 0, \qquad k \in K. \quad (2.3.85)$$

Model (2.3.71)–(2.3.85) only protects against uncertainties on demand. The incorporation of travel time uncertainty, however, is not trivial because this method assumes that the worst-case scenario in a route can be identified by simply taking the combination of γ_l^d that achieve the highest absolute deviation in a route. While this assumption works fine for deviations in demand, it is not true for deviations on travel time, as higher absolute deviations may be absorbed by the waiting time at some customers and, thus, taking the worst-case value of another arc might result in later arrival times in subsequent nodes in the route.

Gounaris, Wiesemann and Floudas (2013) also developed other models with the same approach but different base formulations, some with better and other with worse results than the one presented. We have chosen to present the formulation based on MTZ mainly because it is a more traditional formulation. The highlighted models in their paper were the so-called two-index models and the vehicle assignment models, both solved by tailored branch-and-cut algorithms using robust rounded capacity inequalities cuts. These cuts were developed by the authors as an extension of the rounded capacity inequalities widely used in the deterministic VRP formulations (Poggi; Uchoa, 2014). They are stated as follows:

$$\sum_{i \in N \setminus S} \sum_{j \in S} x_{ij} \geq \left\lceil \frac{1}{Q} \max_{d \in U^d} \sum_{i \in S} d_i \right\rceil, \qquad S \subset N^*,$$

where S represents every possible subset of nodes. The main difference between these cuts and the deterministic ones is that we must consider the maximum realization inside the polyhedral set instead of using the sum of demands. Since considering every possible infeasible set would make the problem prohibitively large, the authors dynamically introduced these cuts in a tailored branch-and-cut algorithm.

Moreover, this algorithm frequently checks for violations on these constraints and thus is important to have an efficient way to evaluate the right-hand sides of these constraints. Gounaris, Wiesemann and Floudas (2013) presented analytical solutions for the multiple knapsack and the factor model uncertainty sets. For example, for the multiple knapsack uncertainty set, with L knapsacks, the maximum realization of the demand for a specific subset of nodes S can be written as:

$$\max_{d \in U^d} \sum_{i \in S} d_i = \sum_{i \in S} \bar{d}_i + \sum_{l=1}^{L} \min\Big(\Delta_l^d, \sum_{i \in S \cap S_d^l} \hat{d}_i\Big),$$

in which S_d^l represents the subsets of nodes that are in a given knapsack l. This method allows to check violations in each subset S in time $\mathcal{O}(|S|)$. In the authors' studies, even the formulations that already incorporate uncertainty through a polynomial number of constraints had their performance improved when those cuts were used.

2.3.2 Formulation based on the linearization of recursive equations

A recent alternative for the dualization scheme was proposed by Munari et al. (2019), who developed an approach 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. This approach is one of the basis of this work.

This formulation is based on the cardinality constrained uncertainty set and assumes that the budget of uncertainty Γ^d is integer and defined as the maximum number of parameters attaining their worst-case values simultaneously in the demand vector. Given a predefined route $r = (v_0, v_1, \ldots, v_h)$, the variable $u_{v_j\gamma}$ represents the largest load required to serve all nodes in the route r up to node v_j, when the demands of $\gamma \leq \Gamma^d$ customers attain their worst-case values simultaneously. The worst-case value of a v_j node is the highest possible demand for this node. Let $\bar{d}_{v_j}$ be the nominal demand of the v_j node of the route and $\hat{d}_{v_j}$ the maximum variation of this node. Then, the value of $u_{v_j\gamma}$ can be computed using the following recursive equations:

$$u_{v_j\gamma} = \begin{cases} d_{v_0}, & \text{if } j = 0, \\ u_{v_{j-1}\gamma} + \bar{d}_{v_j}, & \text{if } \gamma = 0, \\ \max\{u_{v_{j-1}\gamma} + \bar{d}_{v_j}, u_{v_{j-1}(\gamma-1)} + \bar{d}_{v_j} + \hat{d}_{v_j}\}, & \text{otherwise,} \end{cases} \tag{2.3.86}$$

for all $j = 0, 1, \ldots, h$ e $0 \leq \gamma \leq \Gamma^d$. The first line of this expression defines that for the first node of the route, v_0, the total load must simply be its demand (typically zero, since it is the depot). The second line determines that, when no parameter reaches its worst-case value, the total load in a given node v_j is the load in the previous node plus the demand of the node v_j. These first two lines work as boundary conditions. Finally, the third line selects, for each γ and v_j node, the option that gives the largest load: taking the worst γ cases from the nodes prior to v_j and add only the nominal demand of v_j; or taking the previous $\gamma - 1$ worst-case values and add the demand for the worst-case realization of v_j. Then, the maximum load is found in the variable $u_{v_h\Gamma^d}$, where $v_h = n + 1$ is the last node of the route. It is also important to verify if this value respects the vehicle capacity Q to guarantee the robust feasibility of the route.

Munari et al. (2019) converted these recursive equations into constraints for the compact two-index vehicle flow model with MTZ constraints. The constraints associated with load propagation and capacity satisfaction become:

$$u_{j\gamma} \geq u_{i\gamma} + \bar{d}_j x_{ij} - Q(1 - x_{ij}), \qquad (i,j) \in A, \gamma = 0, \ldots, \Gamma^d, \tag{2.3.87}$$

$$u_{j\gamma} \geq u_{i(\gamma-1)} + (\bar{d}_j + \hat{d}_j) x_{ij} - Q(1 - x_{ij}), \qquad (i,j) \in A, \gamma = 1, \ldots, \Gamma^d, \tag{2.3.88}$$

$$\bar{d}_j \leq u_{j\gamma} \leq Q, \qquad j \in N, \gamma = 0, \ldots, \Gamma^d. \tag{2.3.89}$$

Constraints (2.3.87) and (2.3.88) work similarly to the recursive equations (2.3.86), checking for each γ whether the load up to a certain node j ($u_{j\gamma}$) is greater if considering the γ worst cases of the previous nodes, without the node j, or taking the $\gamma - 1$ worst-case values and adding the worst-case value of j. Constraints (2.3.89) guarantee the feasibility of the route, in a given robustness, by limiting the total load to the vehicle capacity. The lower bound in (2.3.89) aims to better restrict the problem seeking a better linear relaxation.

It is possible to apply the same strategy for the travel time. It is important to pay attention to the additional condition of respecting the time window's opening time at node i (a_i) when computing the service's starting time (w_i). Moreover, the upper bound, in this case the closing time of node i (b_i), that acts similarly to the vehicle capacity still exists for this parameter.

Thus, the new time flow variable is defined as $w_{v_j\gamma}$, which represents the service's starting time when considering γ worst-cases values in a route. With this information, the recursive equations for time can be written as follows:

$$w_{v_j\gamma} = \begin{cases} a_{v_0}, & \text{if } j = 0, \\ \max\{a_{v_j}, w_{v_{j-1}\gamma} + s_{v_{j-1}} + \bar{t}_{v_{j-1}v_j}\}, & \text{if } \gamma = 0, \\ \max\{a_{v_j}, w_{v_{j-1}\gamma} + s_{v_{j-1}} + \bar{t}_{v_{j-1}v_j}, & \\ \qquad w_{v_{j-1}(\gamma-1)} + s_{v_{j-1}} + \bar{t}_{v_{j-1}v_j} + \hat{t}_{v_{j-1}v_j}\}, & \text{otherwise.} \end{cases} \tag{2.3.90}$$

for all $j = 0, 1, \ldots, h$ e $0 \leq \gamma \leq \Gamma^t$. Similarly to the demand, the first two lines of the recursive equations are boundary conditions, which are used for the first node in the route and for the calculation in the deterministic case, respectively. Notably, in this second line, a new element was added, namely the opening time windows. When computing the times, it is fundamental to always verify if the arrival time in a node is greater than its opening time. If it is not, the vehicle must wait for a_{vj} before starting the service.

The third line checks if the worst-case scenario for the arrival time is given by verifying which is higher: the opening time of the time window; taking the γ worst-case values before, without using the last arc traveled; or using the previous $\gamma - 1$ worst-case values and considering the worst-case value of the last arc. Thus, it is possible to linearize these equations using the following constraints:

$$w_{j\gamma} \geq w_{i\gamma} + (s_i + \bar{t}_{ij})x_{ij} - M_{ij}(1 - x_{ij}), \qquad (i,j) \in A, \gamma = 0, \ldots, \Gamma^t, \tag{2.3.91}$$

$$w_{j\gamma} \geq w_{i(\gamma-1)} + (s_i + \bar{t}_{ij} + \hat{t}_{ij})x_{ij} - M_{ij}(1 - x_{ij}), \qquad (i,j) \in A, \gamma = 1, \ldots, \Gamma^t, \tag{2.3.92}$$

$$a_i \leq w_{i\gamma} \leq b_i, \qquad i \in N, \gamma = 0, \ldots, \Gamma^t. \tag{2.3.93}$$

Constraints (2.3.91) and (2.3.92) act similarly to the recursive function checking, for each γ up to Γ^t and for each node $j \in N$, if the service begins later when taking the worst γ cases from the previous arcs, without the worst-case value from arc (i,j), or taking the worst $\gamma - 1$ cases from previous arcs and using the worst-case value of arc (i,j). The satisfaction of time windows is ensured by constraints (2.3.93).

The model introduced by Munari et al. (2019) is an extension of the two-index vehicle flow model presented in Section 2.1. In addition to the vehicle flow variables x_{ij} already defined, we have the following additional variables:

- $u_{i\gamma}$: the load in the vehicle after serving node i, considering γ worst-case realizations of the demands;
- $w_{i\gamma}$: the earliest time at which the vehicle can start the service at the node i, considering γ worst-case realizations of the travel times;

The robust counterpart obtained by Munari et al. (2019) using the linearization of recursive equations is given as follows:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij}, \tag{2.3.94}$$

$$\text{s.t.} \sum_{\substack{i=1 \\ j\neq i}}^{n+1} x_{ij} = 1, \qquad j \in N^*, \tag{2.3.95}$$

$$\sum_{\substack{i=0\\i\neq h}}^{n} x_{ih} = \sum_{\substack{j=1\\j\neq h}}^{n+1} x_{hj}, \qquad h \in N^*, \tag{2.3.96}$$

$$u_{j\gamma} \geq u_{i\gamma} + \bar{d}_j x_{ij} - Q(1 - x_{ij}), \qquad (i,j) \in A, \gamma \leq \Gamma^d, \tag{2.3.97}$$

$$u_{j\gamma} \geq u_{i\gamma-1} + (\bar{d}_j + \hat{d}_j) x_{ij} - Q(1 - x_{ij}), \qquad (i,j) \in A, 1 \leq \gamma \leq \Gamma^d, \tag{2.3.98}$$

$$d_j \leq u_{j\gamma} \leq Q, \qquad i \in N, \gamma \leq \Gamma^d, \tag{2.3.99}$$

$$w_{j\gamma} \geq w_{i\gamma} + (s_i + \bar{t}_{ij}) x_{ij} - M(1 - x_{ij}), \qquad (i,j) \in A, \gamma \leq \Gamma^t, \tag{2.3.100}$$

$$w_{j\gamma} \geq w_{i\gamma-1} + (s_i + \bar{t}_{ij} + \hat{t}_{ij}) x_{ij} - M(1 - x_{ij}), \qquad (i,j) \in A, 1 \leq \gamma \leq \Gamma^t, \tag{2.3.101}$$

$$a_i \leq w_{i\gamma} \leq b_i, \qquad i \in N, \gamma \leq \Gamma^t, \tag{2.3.102}$$

$$w_{i\gamma} \geq 0, \qquad i \in N, 0 \leq \gamma \leq \Gamma^t. \tag{2.3.103}$$

$$x_{ij} \in \{0, 1\}, \qquad i, j \in N. \tag{2.3.104}$$

The objective function (2.3.94) consists of minimizing the total traveling cost. Constraints (2.3.95) ensure that every customer is visited only once, while (2.3.96) establishes the vehicle flow by enforcing that every costumer is visited by one vehicle and only one vehicle must depart from it. Constraints (2.3.97) and (2.3.98) ensure the load flow, forbidding subtours, and compute the worst-case scenario of the load by choosing the greatest between using the γ worst previous cases and adding the nominal demand (2.3.97) to it and using the $\gamma - 1$ worst previous cases and adding the worst-case value from the current node (2.3.98). The vehicle capacity is imposed by the constraints (2.3.99). Constraints (2.3.100) and (2.3.101) operate similarly as the constraints (2.3.97) and (2.3.98), respectively, and (2.3.102) ensure that the time windows are respected. Finally, constraints (2.3.103) and (2.3.104) set the domain of the variables.

By having fewer constraints and variables, this formulation performs better than the ones using the dualization approach on benchmarking instances (Munari et al., 2019). Therefore, it may be preferable to use this approach, as long as it is possible to model the problem with a similar dynamic programming framework.

2.4 Other RO formulations in literature

In this section, we briefly present other interesting works in the literature that use RO to develop models for VRP variants under uncertainty. Most of them also proposed different exact and heuristic algorithms to solve the addressed problems, since the compact models were not able to provide optimal solutions for the considered instances in practical computational times. Moreover, the majority of works use the cardinality constrained uncertainty set in their formulation. This data is also summarized in Table 1, which presents the variant addressed, the main solution methods, uncertainty sets and parameters considered by each work.

- Solyali, Cordeau and Laporte (2012) developed a compact MIP formulation for the robust Inventory Routing Problem (IRP) under demand uncertainty. The authors used the dualization approach, proposed by Bertsimas and Sim (2004), to create the robust counterpart. To solve the problem, the authors proposed two robust MIP formulations and implemented them within a branch-and-cut algorithm. One of the models follows a more traditional approach in which all the decision variables must be chosen before the

Table 1 – Summary of works in the RVRP literature

Work	**Variant solved**	**Main Solution methods**	**Uncertainty set**	**Uncertain parameters**
Agra et al. (2012)	VRPTW	Compact Formulation	Cardinality Constrained	Travel Time
Solyali, Cordeau e Laporte (2012)	Inventory Routing Problem	Branch-and-Cut	Cardinality Constrained	Demand
Lee, Lee e Park (2012)	VRP with deadlines	Dantzig-Wolfe Decomposition	Cardinality Constrained	Demand and Travel Time
Gounaris et al. (2013)	CVRP	Branch-and-Cut	Knapsack and Factor Model	Demand
Agra et al. (2013)	VRPTW	Branch-and-Cut and Column-and-roll-generation	Cardinality Constrained	Travel Time
Tajik et al. (2014)	Pollution routing problem	Compact Formulation	Cardinality Constrained	Service and Travel Time, CO2 emission and fuel
Hu et al. (2018)	VRPTW	Column-and-roll-generation	Cardinality Constrained	Demand and Travel Time
De La Vega, Munari e Morabito (2018)	VRPTW w/ multiple deliverymen	Heuristic	Cardinality Constrained	Demand
Li e Chung (2019)	CVRP and SDVRP for disaster relief	Metaheuristic	Cardinality Constrained	Demand and Travel Time
Rahbari et al. (2019)	CVRP w/ perishable goods	Branch-and-Cut	Cardinality Constrained	Travel Time and freshness-life
Munari et al. (2019)	VRPTW	Compact formulation and Branch-price-and-cut	Cardinality Constrained	Demand and Travel Time
Subramanyam et al. (2020)	HVRP	Metaheuristic	Many	Demand
Bartolini et al. (2020)	TSPTW	Column-generation	Knapsack	Travel Time
Pessoa et al. (2021)	CVRP	Branch-cut-and-price	Knapsack	Demand

uncertainty realization, while the other allows that part of the variables can be selected after the realization, in the called adjustable robust formulation (Ben-Tal et al., 2004). Both formulations presented solutions immunized to uncertainty, with the adjustable one being more computationally efficient albeit a little more expensive. Both formulations were able to solve to optimality literature instances with up to 30 customers and seven periods within 2 hours.

- Lee, Lee and Park (2012) tackled the robust vehicle routing problem with deadlines under travel time and demand uncertainty. This particular variant is similar to the RVRPTW, but imposes only the closing time of time windows. Thus, vehicles can start its service immediately after arriving in the customer, given the deadlines are respected. The authors used the Bertsimas and Sim (2004) uncertainty set to represent the variable parameters. They applied the Dantzig-Wolfe decomposition approach and the uncertainty was encapsulated in the column generation subproblem. The authors tested their algorithm in adapted instances from VRPTW and CVRP literature (Solomon, 1987; Augerat, 1995). The instances were of relatively small size, with 25 customers for the ones originated from the VRPTW and up to 40 nodes for the ones adapted from the CVRP. The proposed algorithm delivered safer, albeit more expensive, solutions for all instances when compared to the deterministic solution. However, it was only able to solve slightly more than half of the instances from Solomon (1987) and only two from Augerat (1995) to optimality.

- Agra et al. (2013) studied the VRPTW under uncertainty on travel time in the same maritime transportation context as Agra et al. (2012). The authors proposed two formulations for the problem. The first formulation extends the resource inequalities formulation by employing adjustable robust optimization techniques. In adjustable RO, some decision variables are allowed to adapt themselves as uncertain parameters vary in uncertainty sets. The second formulation implicitly considers uncertainty in a path inequalities formulation. Both formulations presented similar computational performance and were faster than the layered formulation of Agra et al. (2012). Similarly to the model proposed by Agra et al. (2012), while these new formulations were computationally tractable, they were considerably more difficult to solve than the deterministic problem.

- Tajik et al. (2014) addressed the pollution routing problem with pickup and delivery under uncertainty on service time, travel time, fuel consumption and CO_2 emission cost. Pollution routing differs from the traditional problems by not focusing only on costs, but also on the fuel consumption, whose reduction may lesser the environmental and human health impacts of the company. The authors developed an MILP formulation to solve the problem and analysed the impact of using this technique over considering deterministic demand. Notably, the objective function value of the robust models suffered less deviation than the deterministic one.

- Hu et al. (2018) proposed a robust optimization model for the VRPTW with demand and travel time uncertainty. They created, and used, an uncertainty set similar to the cardinality constrained set but with the budget (Γ) size being dependent on the number of nodes in the route. Essentially, Γ will be a percentage, chosen by the decision maker, of the route's size. The authors developed the model by adapting the resource inequalities formulation from Agra et al. (2013). To deal with bigger instances, they also designed a two-stage algorithm based on a modified adaptive variable neighborhood search heuristic.

- De La Vega, Munari and Morabito (2018) proposed a compact formulation for the VRPTW with multiple deliverymen under demand uncertainty, a variant in which the service time in a node depends on the number of deliverymen assigned in the vehicle. To incorporate variability, the authors proposed a compact formulation using the traditional dualization approach Bertsimas and Sim (2004). Due to the long computational times required to solve this formulation, the authors proposed a RO extension of the Solomon's heuristic I1 to obtain feasible solutions.

- Li and Chung (2019) worked with the RCVRP and the robust split delivery VRP under demand and travel time uncertainty in a context of disaster relief routing. This problem often arises in the aftermath of disastrous events, when delivering critical supplies and/or services to the affected population in need. In this context, the focus is shifted from costs/profit to the welfare of the victims. This is usually reflected on the objective function by considering the time it takes to service the affected victims. The authors proposed RO models considering a polyhedral uncertainty set with different objective functions, such as summation of arrival times and the latest arrival time, and analysed their impact on solution. They also proposed a two-stage heuristic method that combines insertion and tabu search algorithms.

- Rahbari et al. (2019) developed two bi-objective formulations for the RCVRP with cross docking in a context of perishable goods distribution. The first objective of these models is minimizing the total delivery costs and the second objective is maximizing the total weighted freshness of the delivered products. In one of the models the authors considered uncertainty on travel times, while in the other they considered uncertainty on freshness-life of product, both were developed by using an approach based on Gounaris, Wiesemann and Floudas (2013). The results show that uncertainty in the travel time considerably deteriorates objective function whereas uncertainty on the freshness-life has little impact on distribution costs and highly increases the freshness of the delivered items.

3 New Formulations for the VRPTW and RVRPTW

Before addressing the practical aircraft routing problem, we first study a classical problem to design the robust methods which will be later extended to tackle real-world instances. Thus, in this chapter, we present a novel formulation for the traditional VRPTW based on commodity flow and RO models which resort to the linearization technique of recursive equations proposed by Munari et al. (2019) which will later serve as basis for our case study. In Section 3.1, we present a new deterministic formulation for the VRPTW based on the single commodity flow (CF) formulation. Then, we derive the robust counterpart of this formulation to obtain a new RO model for the RVRPTW in Section 3.2, using the cardinality constrained uncertainty set. Then, in Section 3.3, we show the models developed for the single knapsack uncertainty set and in Section 3.4 we present the models for the multiple knapsack uncertainty set. In Section 3.5 we explain the tailored B&C algorithm designed for this problem. Finally, in Section 3.6, we present the results of computational experiments using benchmark instances from the literature.

3.1 CF formulation for the VRPTW

In this section, we describe the model for the (deterministic) VRPTW that inspired our new formulation for the RVRPTW under time and demand uncertainty. This formulation is based on the single commodity flow model for the CVRP (Gouveia, 1995; Letchford; Salazar-González, 2015) which is an extension of the model originally proposed by Gavish and Graves (1978) for the Traveling Salesman Problem (TSP). This formulation is known for having stronger linear relaxation than the formulation based on MTZ constraints.

While there are formulations based on commodity flow constraints for the CVRP in literature (Gouveia, 1995) and variations of it with stronger linear relaxations (Letchford; Salazar-González, 2006; Letchford; Salazar-González, 2015), we did not find any compact formulation for the VRPTW based on commodity flow variables in the literature. There is, however, one for the TSP with Time Windows (Langevin et al., 1993) and other for the Split Delivery VRPTW (Bianchessi; Irnich, 2019), both in a deterministic context. Thus, this section presents the adaptation developed on the formulation for the CVRP in order to introduce time windows and time flow constraints.

Starting with the CF formulation of the CVRP, we use the same notation as in Chapter 2 and thus $N = \{0, \dots, n+1\}$ is the set of nodes, where nodes $1, \dots, n$ represent customers, while nodes 0 and $n+1$ represent, respectively, the departure and arrival depots. Set A comprises the arcs between nodes (a complete graph is assumed). As parameters, we have the demand d_i for each node $i \in N$, with $d_0 = d_{n+1} = 0$; the capacity of vehicles, Q; and the travel cost c_{ij} for each arc $(i,j) \in A$. The decision variables of this model are given by the binary variable x_{ij} which indicates whether arc $(i,j) \in A$ is traversed; and the continuous variable f_{ij} that represents the load of the vehicle that traverses this arc. The model is then given by:

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij}, && && (3.1.1)\\
\text{s.t.} \quad & \sum_{\substack{i=1\\ i\neq j}}^{n+1} x_{ij} = 1, && j \in N^*, && (3.1.2)\\
& \sum_{\substack{i=0\\ i\neq h}}^{n} x_{ih} = \sum_{\substack{j=1\\ j\neq h}}^{n+1} x_{hj}, && h \in N^*, && (3.1.3)\\
& \sum_{h=0}^{n} f_{hi} - \sum_{j=1}^{n+1} f_{ij} = d_i, && i \in N^*, && (3.1.4)\\
& f_{ij} \leq Qx_{ij}, && (i,j) \in A, && (3.1.5)\\
& x_{ij} \in \{0,1\}, f_{i,j} \geq 0, && (i,j) \in A. && (3.1.6)
\end{aligned}
$$

The objective function (3.1.1) consists of minimizing the total traveling costs. Constraints (3.1.2) ensure that each customer is visited only once, whereas (3.1.3) guarantee the correct vehicle flow through the nodes. Constraints (3.1.4) impose the correct load flow, ensuring that the load in the vehicle that leaves a node i is equal to the difference between the load when the vehicle arrived at this node and the node demand d_i. These constraints also forbid subtours. Constraints (3.1.5) prevent the vehicle from exceeding its capacity in any arc. Finally, (3.1.6) define the domain of variables x_{ij} and f_{ij}.

It is possible to strengthen the model by replacing constraints (3.1.5) with tighter ones (Gavish, 1984), such as:

$$d_j x_{ij} \leq f_{ij} \leq (Q - d_i)x_{ij}, \quad (i,j) \in A. \qquad (3.1.7)$$

Note that these constraints have the same function as those of the original model, but the upper and lower bounds are more restricted for each node. The lower bound of (3.1.7) is guaranteed because when traveling to any node j from any node i, the vehicle must have at least the quantity needed to supply j, while the upper bound is secured as the maximum quantity that can be taken in arc (i,j) is the capacity of the vehicle minus the demand of node i.

To adapt this model to the VRPTW, we use the same logic as the original by treating time as a second commodity. Thus, we introduce the continuous variable g_{ij} that represents the elapsed time of a route (it can be seen as the “load” of this second commodity) when the vehicle leaves node i towards node j. Thus, g_{ij} corresponds to the time the vehicle finishes serving customer i and is available to travel to customer j if arc (i,j) is taken. Recall that the VRPTW require the following additional parameters, as defined in Chapter 2: a_i and b_i, which represent the opening and closing time of the node's time windows, respectively; the service time at node i, s_i; and the travel time when traversing arc (i,j), t_{ij}. The model does not allow the vehicle to arrive at the customer after the closing time (b_i) and the vehicle must wait until the opening time (a_i) to start its services, if it arrives earlier.

With these definitions, it is possible to develop the model for the VRPTW based on commodity flow variables. To our knowledge, there was no compact formulation explicitly defined for the VRPTW in which time propagation are modeled using this type of constraints. To introduce them, we add the following constraints to model (3.1.1)-(3.1.6):

$$\sum_{j=1}^{n+1} g_{ij} \geq s_i + \sum_{h=0}^{n}(g_{hi} + t_{hi}x_{hi}), \quad i \in N^*, \tag{3.1.8}$$

$$(a_i + s_i)x_{ij} \leq g_{ij} \leq (b_i + s_i)x_{ij}, \quad (i,j) \in A. \tag{3.1.9}$$

Constraints (3.1.8) act similarly to (3.1.4) and impose the correct flow of time trough the visited nodes in a route. Constraints (3.1.9) ensure that time windows are met. They also guarantee that the variable g_{ij} is non-negative if arc (i,j) is traversed. At the lower bound, g_{ij} must assume at least the value $(s_i + a_i)$, since the earliest possible time to finish the service on a node is when the vehicle starts its service immediately after the customer opens. At the upper bound, the maximum value that g_{ij} can assume is, by definition, the moment in which the vehicle starts its service at the exact time that the time window closes. Furthermore, if an arc (i,j) is not traversed, i.e $x_{ij} = 0$, constraints (3.1.9) ensure that g_{ij} is zero. Thus, (3.1.9) also impose that the elapsed time is computed correctly in constraints (3.1.8), since the only variable g_{ij} allowed to take a non-zero value is the one where $x_{ij} = 1$, resulting in a constraint equivalent to (2.1.9) (when $x_{ij} = 1$) and turning-off any other constraint considering that uses arc $(i,l), l \neq j$. It is worth noting that constraints (3.1.7) can be used instead of (3.1.5) in the formulation of the VRPTW as well.

There are two notable differences between constraints (3.1.4) and (3.1.8). The first one is that the order of the summations is reversed because, while in the product loading the amount carried by the vehicle decreases after each visit (delivery), the time increases. For example, suppose we have the route $R = \{0, i, n+1\}$ in which $d_i = 50$, $Q = 100$, $t_{0i} = t_{in+1} = 60$, $s_i = 10$, $a_i = 30$ and $b_i = 100$. Then, the load variables f_{0i}, which represent the total product load in the vehicle when it arrives in node i, and f_{in+1}, defined as the product load in the vehicle when it departs from customer i, would take, respectively, the values 10 (or any value between d_i=10 and Q=200) and 0 (or any value 10 units lower than f_{0i}). This was imposed by constraints (3.1.4), because:

$$\sum_{h=0}^{n} f_{hi} - \sum_{j=1}^{n+1} f_{ij} = f_{0i} - f_{in+1} = d_i = 10,$$

$$f_{0,i} = f_{in+1} + 10.$$

Conversely, for the time variables, the arrival time in node i, g_{0i}, will be smaller than the arrival time in the depot g_{in+1}. Hence, g_{0i} will assume a value of at least 60, whereas g_{in+1} needs to be at least 130 due to constraints (3.1.8), according to the following equations:

$$g_{0i} \geq s_0 + \sum_{h=0}^{n}(g_{hi} + t_{hi}x_{hi}) = 0 + 0 + 60x_{0i} = 60,$$

$$g_{in+1} \geq s_i + \sum_{h=0}^{n}(g_{hn+1} + t_{hn+1}x_{hn+1}) = g_{0i} + s_i + 60x_{in+1} = 60 + 10 + 60 = 130.$$

The second difference between constraints (3.1.4) and (3.1.8) is the use of an inequality instead of an equation because if the vehicle arrives before the opening of the time window, an equality in (3.1.8) would make the solution infeasible. The following example shows why an equality would not work in such case. Suppose in a solution that the minimum arrival time in node

j after visiting node i is 60 minutes (g_{ij}), due to constraints (3.1.8); however, the lower bound of the time windows for node j, enforced by constraints (3.1.9), is 90 minutes (i.e., $a_i > g_{ij}$). If (3.1.8) were equations, they would impose g_{ij} to be 60 minutes, and the solution would be infeasible as the variable's lower bound is 90 because of constraints (3.1.9). It is worth noting that the use of inequalities does not affect the solution in terms of the vehicle flow.

3.2 CF formulation based on the cardinality constrained uncertainty set

In this section, we propose a new compact model for the RVRPTW considering the traditional cardinality constrained uncertainty set, based on the CF formulation presented in the previous section.

As observed in the literature review presented in Chapter 2, the cardinality constrained set has been widely used in different contexts, such as maritime transportation (Agra et al., 2012; Agra et al., 2013), urban delivery routing (De La Vega; Munari; Morabito, 2018), disaster relief routing (Li; Chung, 2019), and on-demand air transportation (Campos; Alvarez; Munari, 2019). Currently, the best compact model for the RVRPTW is the one proposed by Munari et al. (2019), which was obtained by the linearization of recursive equations. Hence, the same approach is used to derive the robust counterpart of the CF formulation. This formulation has a stronger linear relaxation than the one proposed by Munari et al. (2019) and hence might show better computational times in general-purpose ILP solvers as well as in tailored branch-and-cut methods.

To obtain the robust counterpart of the CF model, the first change is the incorporation of the index γ into the variables f_{ij} and g_{ij}. As in the MTZ-based formulation of Munari et al. (2019), this index represents the number of parameters that attains their worst-case values simultaneously in a route for demand ($\gamma^d \leq \Gamma^d$) and travel time ($\gamma^t \leq \Gamma^t$). In this way, we have the variables $f_{ij\gamma}$ and $g_{ij\gamma}$ that represents the load and elapsed time, respectively, for the vehicle that traverses arc (i, j) considering γ worst-case realizations. With these variables we can redefine the load and time flow constraints of the CF model. For instance, constraints (3.1.4) are replaced with:

$$\sum_{h=0}^{n} f_{hi\gamma} \geq \bar{d}_i + \sum_{j=1}^{n+1} f_{ij\gamma}, \qquad i \in N^*, \gamma = 0, \ldots, \Gamma^d, \tag{3.2.1}$$

$$\sum_{h=0}^{n} f_{hi\gamma} \geq \bar{d}_i + \hat{d}_i + \sum_{j=1}^{n+1} f_{ij(\gamma-1)}, \quad i \in N^*, \gamma = 1, \ldots, \Gamma^d. \tag{3.2.2}$$

These constraints guarantee the worst-case load in the vehicle, according to the following cases: γ worst-case realizations previously occurred and then we consider only the nominal demand of node i, as computed in the right-hand side of (3.2.1); or $\gamma - 1$ worst-case realizations previously occurred and we consider both the nominal demand of node i and its maximum deviation, as calculated in the right-hand side of (3.2.2).

It is important to emphasize that, on the deterministic case (that is, for $\gamma = \Gamma^d = 0$), constraints (3.2.2) are not defined and (3.2.1) becomes equivalent to (3.1.4). It is also worth noting that it was necessary to replace the equality from (3.1.4) with an inequality, since the problem would become infeasible by forcing two different equalities. This is without any loss of optimality, since $f_{hi\gamma}$ can still assume the lowest possible value, to keep the solution feasible (according to the robust perspective).

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 uncertainties in demands and travel times:

$$\min \quad \sum_{(i,j)\in A} c_{ij}x_{ij}, \tag{3.2.3}$$

$$\text{s.t} \quad \sum_{\substack{i=1\\ i\neq j}}^{n+1} x_{ij} = 1, \qquad j \in N^*, \tag{3.2.4}$$

$$\sum_{\substack{i=0\\ i\neq h}}^{n} x_{ih} = \sum_{\substack{j=1\\ j\neq h}}^{n+1} x_{hj}, \qquad h \in N^*, \tag{3.2.5}$$

$$\sum_{h=0}^{n} f_{hi\gamma} \geq \bar{d}_i + \sum_{j=1}^{n+1} f_{ij\gamma}, \qquad i \in N^*, \gamma \leq \Gamma^d, \tag{3.2.6}$$

$$\sum_{h=0}^{n} f_{hi\gamma} \geq \bar{d}_i + \hat{d}_i + \sum_{j=1}^{n+1} f_{ij(\gamma-1)}, \qquad i \in N^*, 0 < \gamma \leq \Gamma^d, \tag{3.2.7}$$

$$\bar{d}_j x_{ij} \leq f_{ij\gamma} \leq (Q - \bar{d}_i)x_{ij}, \qquad (i,j) \in A, \gamma \leq \Gamma^d, \tag{3.2.8}$$

$$\sum_{j=1}^{n+1} g_{ij\gamma} \geq s_i + \sum_{h=0}^{n} (g_{hi\gamma} + \bar{t}_{hi}x_{hi}), \qquad i \in N^*, \gamma \leq \Gamma^t, \tag{3.2.9}$$

$$\sum_{j=1}^{n+1} g_{ij\gamma} \geq s_i + \sum_{h=0}^{n} (g_{hi\gamma-1} + (\bar{t}_{hi} + \hat{t}_{hi})x_{hi}), \qquad i \in N^*, 0 < \gamma \leq \Gamma^t, \tag{3.2.10}$$

$$(s_i + a_i)x_{ij} \leq g_{ij\gamma} \leq (b_i + s_i)x_{ij}, \qquad (i,j) \in A, \gamma \leq \Gamma^t, \tag{3.2.11}$$

$$x_{ij} \in \{0,1\}, \qquad (i,j) \in A, \tag{3.2.12}$$

$$g_{i\gamma} \geq 0, \qquad i \in N, 0 \leq \gamma \leq \Gamma^t. \tag{3.2.13}$$

Objective function (3.2.3) consists of minimizing the total traveling costs. Constraints (3.2.4) and (3.2.5) guarantee the correct vehicle flow. Load constraints (3.2.6) and (3.2.7) guarantee the correct load propagation, and constraints (3.2.8) enforce the vehicles' capacity. Constraints (3.2.9) and (3.2.10) act similarly but for the time propagation. Constraints (3.2.11) ensure that the time windows are respected that is, the vehicle must wait the time window open before it can start the delivery service on the node and it cannot start the service after the time window is closed. Finally, constraints (3.2.12) and (3.2.13) ensure the domain of the variables.

Thanks to the capacity and time windows constraints (3.2.8) and (3.2.11), only one variable related to product load ($f_{ij\gamma}$) and time propagation ($g_{ij\gamma}$) is allowed to have a non-null value for each i and γ, specifically the one related to arc (i,j) where $x_{ij} = 1$. Thus, suppose that in an optimal solution, we have $x_{i_1j_1} = 1$ and $x_{j_1k_1} = 1$. Then, time constraints (3.2.9) and (3.2.10) related to node j_1, for any $\gamma > 0$, can be simply represented as follows:

$$g_{j_1k_1\gamma} \geq g_{i_1j_1\gamma} + \bar{t}_{i_1j_1} + s_{j_1}, \tag{3.2.14}$$

$$g_{j_1k_1\gamma} \geq g_{i_1j_1\gamma-1} + \bar{t}_{i_1j_1} + \hat{t}_{i_1j_1} + s_{j_1}. \tag{3.2.15}$$

From the MTZ-based model discussed in Section 2.3.2, the time constraints related to node j_1 in path (i_1, j_1, k_1) should be represented by the following constraints:

$$w_{j_1\gamma} \geq w_{i_1\gamma} + \bar{t}_{i_1j_1} + s_{j_1}, \tag{3.2.16}$$

$$w_{j_1\gamma-1} \geq w_{i_1\gamma} + \bar{t}_{i_1j_1} + \hat{t}_{i_1j_1} + s_{j_1}. \tag{3.2.17}$$

Given that variables $g_{j_1k_1\gamma}$ and w_{j_1} both represent the departure times from node j_1, then it is possible to see that constraints (3.2.14)-(3.2.15) are equivalent to (3.2.16)-(3.2.17). A similar conclusion can be drawn for the load constraints. Thus, since both models have equivalent load and time constraints, and turn-off variables and constraints for arcs not used, we can conclude that model (3.2.3)-(3.2.13) is equivalent to the one proposed by Munari et al. (2019). Finally, it is possible to use this model for the RCVRP by simply dropping constraints (3.2.9)-(3.2.11)

3.3 Formulations based on the single knapsack uncertainty set

In this section, we extend the MTZ-based and the CF formulations for the RVRPTW considering the single knapsack uncertainty set, a variant of the knapsack uncertainty set that consider that there is a single knapsack encompassing all nodes/arcs. In this set, the decision maker limits the sum of deviation value by a value Δ for all routes, instead the maximum number of parameters attaining their worst-case value simultaneously.

3.3.1 MTZ-based formulation for the RVRPTW

This model adapts the formulation described in Subsection 2.3.2 and adapts it for the single knapsack uncertainty set. The vehicle flow variables are the same as the ones in model (2.3.94)-(2.3.104), while the load and time variables have a slightly different meaning:

- $u_{i\delta}$: the load that the vehicle carried up to node i, considering up to a total deviation of δ over the demand's nominal value of all nodes visited until this node;
- $w_{i\delta}$: the earliest time that a vehicle can start the service at the node i, considering up to a total deviation δ over the travel time's nominal value of all arcs traversed until this node.

As we can see, now the index δ represents the summation of load/time units over the nominal value considered until the current node instead of the number of parameters attaining their worst-case values simultaneously. Note that these changes bring some limitations when compared to the dualization scheme, namely the budget and deviations must be integer, since the deviation is computed in the index, and the problem should have relatively small budgets, because this size directly affects the number of decision variables in the problem. Later in this section we will discuss some workarounds for these problems.

To help understanding these variables we created Figure 2, which presents an example to study the behavior from product load variable $u_{i\delta}$. In this figure, we present a route $R = (r_0, r_1, \ldots, r_4, r_{n+1})$ where each node i in this route has a nominal demand $\bar{d}_i$ and a maximum deviation $\hat{d}_i$. In this example, we define the budget of uncertainty for demand, Δ^d, as 30 product units. Then, for the given route, we compute the value of variable $u_{i\delta}$ and show the value inside a box that represents the node r_i. Under each box there is a container representing their knapsack and the total deviation already considered up to node i, represented by the blue boxes. Note that $u_{i\delta}$ accounts for the worst-case vehicle load when it arrives in node i, while delta accounts for the total accumulated deviation in this load. If in a particular case the budget's size is greater than the total deviation of all nodes in a route, we simply need to consider the worst-case value of all nodes in it, similarly to the approach used by (Soyster, 1973).

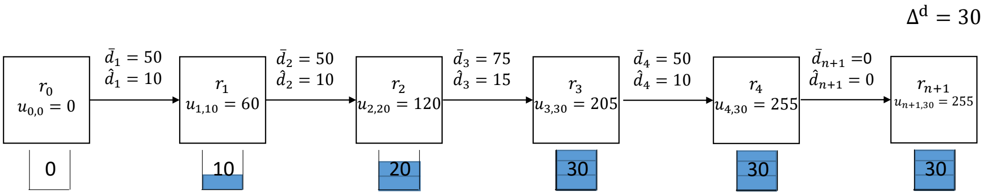


Figure 2 – Example of calculation of load variables for the single knapsack uncertainty set.

In this example, we can compute the total load in a node, considering the RO approach, by simply adding the maximum deviation possible until the knapsack is filled to the total nominal demand in the route. Thus, we can compute the value of each variable $u_{i\delta}$, by the following formula:

$$u_{i\delta} = \sum_{j=0}^{i} \bar{d}_j + min(\delta, \sum_{j=0}^{i} \hat{d}_j). \tag{3.3.1}$$

In this particular example we start the route departing from depot r_0, with a total load of zero. When we visit the first customer of the route, the total load is filled by 60 units, 50 from the nominal demand ($\bar{d}_1$) and 10 units of deviation ($\hat{d}_1$). Since we considered 10 units of deviation in our variables, we fill this value in the knapsack and set the variable $u_{1,10}$ to 60, meaning this is the load when we allow up to 10 units of deviation over the nominal demand. Note that, by our definition of (3.3.1), if we use a δ greater than 10 we would still have a total load of 60 in node 1 (for example, $u_{1,30} = 60$), since there is not another source of deviation other than this first node. On the other hand, if δ is smaller than 10, we would simply need to fill the knapsack as much as possible (up to δ) and the total load would be simply the total nominal demand plus δ. Thus, in this first node $\delta = 10$ is the first point in which we achieve the maximum load possible ($u_{1,10} = 60$) and that's why it is presented in the figure.

The second customer node of this example has a similar behavior as the first. Its deviation, $\hat{d}_2 = 10$, can be inserted completely into the knapsack and thus, after taking into account for the nominal demand, a total load in the route would be 120, and the knapsack would have a total of 20 units. This time the maximum load is achieved when $\delta = 20$ ($u_{2,20} = 120$), any value higher than that would provide the same load while any δ lower than that would result in a load where we simply fill the knapsack by adding δ to the nominal demand of the nodes in the route up to the analyzed node. In the third customer of this example we finally face the situation in which not all deviation from the node ($\hat{d}_3 = 15$) fits in the knapsack. In this case, we need to simply fill the knapsack until it is completely full. In this example, that means consider 10 units of deviation (in order to completely fill the knapsack), and not consider the excess (5 units). Thus, after adding the nominal demand, we get a total load of $u_{3,\Delta^d} = u_{3,30} = 205$. Finally, since the knapsack is already full, we only need to consider the nominal value of the last customer, resulting in a total load of $u_{4,\Delta^d} = u_{4,30} = 255$ and its whole deviation ($\hat{d}_4 = 10$) is ignored. Thus, once the knapsack is completely filled we only need to consider the deterministic demand from the remaining customers in the route.

To compute the value of the load variables in the model, we used a dynamic-programming approach akin to the one used in the previous uncertainty set. Thus, for a given route $R = (r_0, r_1, \ldots, r_n)$, the recursive equations designed to compute these variable are the following:

$$u_{r_j\delta} = \begin{cases} \bar{d}_{r_0}, & \text{if } j = 0, \\ u_{r_{j-1}\delta} + \bar{d}_{r_j}, & \text{if } \delta < \hat{d}_{r_j}, \\ \max\{u_{r_{j-1}\delta} + \bar{d}_{r_j}, u_{r_{j-1}(\delta - \hat{d}_{r_j})} + \bar{d}_{r_j} + \hat{d}_{r_j}\}, & \text{if } \hat{d}_{r_j} \leq \delta \leq \Delta^d, \\ \max\{u_{r_{j-1}\Delta^d} + \bar{d}_{r_j}, u_{r_{j-1}(\Delta^d - \lambda)} + \bar{d}_{r_j} + \lambda\}, & \text{if } \lambda \leq \hat{d}_{r_j}, \lambda \leq \Delta^d. \end{cases} \tag{3.3.2}$$

The first two lines are boundary conditions of the problem. The first one is used in the first node in the route, the depot, and states that the total load in that node is its own demand, usually zero. The second one is used in the deterministic case and computes the total load as the sum of the nominal demands only. This second line is also used to compute the values of the load δ when it is smaller than the deviation of the node, since we only need to check if u_{j,Δ^d} is feasible, the total deviation in those can be partially ignored. The third line is used when we try to fill the knapsack with the full deviation from node r_j, we choose the highest between not using the deviation in that particular node and inserting its maximum deviation. To do this, we only need to take the load from the previous node, considering a total deviation of $\delta - \hat{d}_{r_j}$. Finally, the last line is used to fill the remaining space of the knapsack, when the deviation in a node is greater than the remaining space, by choosing between not considering deviation in that node or taking a value λ that fills it. The index λ in this last type of constraint was created to measure the amount of deviation inserted in the knapsack when it is less than the maximum deviation of the evaluated node. Note that in the case where $\delta = \Delta^d$, the third and fourth line will be equivalent if $\lambda = \hat{d}_{r_j}$. So we can further simplify the recursive equations by reducing the domain of δ in the third line to $\hat{d}_{r_j} \leq \delta < \Delta^d$.

The travel time is trickier to compute due to the opening time windows. If we fill the knapsack as quickly as possible like in the demand, we might choose a deviation that is overshadowed by the waiting time (regarding the opening time of a customer time window), whereas considering the same deviation value in a later point could turn the solution infeasible. Thus, the dynamic programming equations are different from the demand, for the same route $R = (r_0, r_1, \ldots, r_n)$, because we are not allowed to apply the same simplifications as in the demand. Thus, to compute variable $w_{r_j\delta}$, which accounts for the worst-case vehicle arrival time in node r_j, considering a total deviation of $\delta \leq \Delta^t$ time units, the following expression is used:

$$w_{r_j\delta} = \begin{cases} a_{r_0}, & \text{if } j = 0, \\ \max\{a_{r_j}, w_{r_{j-1}\delta} + \bar{t}_{r_{j-1}r_j} + s_{r_{j-1}}\}, & \text{if } \delta = 0, \\ \max\{a_{r_j}, w_{r_{j-1}\delta} + \bar{t}_{r_{j-1}r_j} + s_{r_{j-1}}, & \\ \qquad w_{r_{j-1}(\delta-\lambda)} + \bar{t}_{r_{j-1}r_j} + s_{r_{j-1}} + \lambda\}, & \text{if } \lambda \leq \hat{t}_{r_{j-1}r_j}, \lambda \leq \delta \leq \Delta^t. \end{cases} \tag{3.3.3}$$

Similarly to the equations for demand, the first two lines are boundary conditions. The first line is used in the first node in the route and sets the starting time of the vehicle, while the second one computes the deterministic elapsed time in the route. Note that for the elapsed time we must verify if the vehicle arrives before the opening time windows, and if it does, it must wait until it is time to start the service. Finally, the third line tries to compute the highest elapsed time when considering a deviation of δ. We need to check if it is better not to consider deviation in that particular arc or taking some level of deviation (λ) that ranges from 1 unit up to the deviation of the arc or δ, whichever is higher. We also need to confirm if the total elapsed time respects the opening time windows. Note that these recursive equations are used to compute the worst-case arrival time. In order to check the feasibility of the route we would need to check at the end of each iteration of the dynamic programming algorithm if $w_{r_j\delta} \leq b_{r_j}$ for each $r_j \in R$ and $\delta \leq \Delta^t$. In the compact model. these verifications are introduced as the upper-bound of the time windows constraints.

With this new interpretation, we were able to develop the following model for this uncertainty set:

$$\min \quad \sum_{(i,j)\in A} c_{ij}x_{ij}, \tag{3.3.4}$$

$$\text{s.t.} \quad \sum_{\substack{i=1\\ i\neq j}}^{n+1} x_{ij} = 1, \qquad j \in N^*, \tag{3.3.5}$$

$$\sum_{\substack{i=0\\ i\neq h}}^{n} x_{ih} = \sum_{\substack{j=1\\ j\neq h}}^{n+1} x_{hj}, \qquad h \in N^*, \tag{3.3.6}$$

$$u_{j\delta} \geq u_{i\delta} + \bar{d}_j + M(x_{ij}-1), \qquad i,j \in N, \delta \leq \Delta^d, \tag{3.3.7}$$

$$u_{j\delta} \geq u_{i(\delta-\hat{d}_j)} + \bar{d}_j + \hat{d}_j + M(x_{ij}-1), \qquad i,j \in N, \hat{d}_j \leq \delta \leq \Delta^d, \tag{3.3.8}$$

$$u_{j\Delta^d} \geq u_{i(\Delta^d-\lambda)} + \bar{d}_j + \lambda + M(x_{ij}-1), \qquad i,j \in N, \lambda < \hat{d}_j, \tag{3.3.9}$$

$$u_{j\Delta^d} \leq Q, \qquad j \in N, \tag{3.3.10}$$

$$w_{j\delta} \geq w_{i(\delta-\lambda)} + \bar{t}_{ij} + \lambda + s_i + M(x_{ij}-1), \qquad i,j \in N, \delta \leq \Delta^t, \lambda \leq \hat{t}_{ij}, \lambda \leq \delta, \tag{3.3.11}$$

$$a_j \leq w_{j\delta} \leq b_j, \qquad j \in N, \delta \leq \Delta^t, \tag{3.3.12}$$

$$x_{ij} \in \{0,1\}, \qquad i,j \in N, \tag{3.3.13}$$

$$u_{i\delta} \geq 0, \qquad i \in N, 0 \leq \delta \leq \Delta^d, \tag{3.3.14}$$

$$w_{i\delta} \geq 0, \qquad i \in N, 0 \leq \delta \leq \Delta^t. \tag{3.3.15}$$

The objective function (3.3.4) is the same as in the other formulations presented in this work and thus it seeks to minimize the total travelling cost. Constraints (3.3.5) and (3.3.6) ensure, respectively, that every customer node is visited only once and when a customer is visited one, and only one, vehicle must depart from it. Constraints (3.3.7)-(3.3.9) tries to find the highest load possible in node j considering a budget $\delta \leq \Delta^d$, similarly to the recursive equations (3.3.2). Constraints (3.3.7) represent the case in which we only add the nominal demand of the node into the vehicle's load and consider the total deviation of δ units in the previous nodes of the route, it is also used to find the deterministic load (when $\delta = 0$). In Constraints (3.3.8) we try to fit the whole deviation from the node into the knapsack of size λ, if possible, similarly to the third line from recursive equations (3.3.2). Finally, constraints (3.3.9) try to fill the knapsack of size Δ^d using a similar approach as the fourth line from recursive equations (3.3.3), checking different sizes (λ) of deviation, unit by unit, to be included into the variable up to $\hat{d}_j$. The model must attain the highest load between the three options. Constraints (3.3.10) ensure that vehicle capacity is respected. Notably, we just need to verify for $u_{j\Delta^d}$, since $u_{j\delta} \leq u_{j\Delta^d}, \delta < \Delta^d$, if a solution is feasible for $u_{j\Delta^d}$, it will be feasible for any $\delta \leq \Delta^d$. If we consider only these constraints and the ones related to the variable's domain (3.3.13 and 3.3.14), we have a formulation for the RCVRP.

Constraints (3.3.11) are used to compute the elapsed time following a strategy close to recursive equations (3.3.3). They compute the arrival time in node j considering a total deviation of δ in the route. We evaluate the quantity of time deviation λ that should be considered from that particular node, while the remaining $\delta - \lambda$ units of deviation happened in previous nodes from the route, and choose the λ that results in the latest arrival time. Constraints (3.3.12) ensure the time windows are respected. It is interesting to note that, due to the time windows' opening time, we were unable to reduce the number of constraint as we did on demand for in constraints (3.3.7)-(3.3.9). That happens because the maximum deviation in a route R for the demand is basically the minimum between knapsack size (Δ^d) and the sum of maximum deviation from each node ($\sum_{j \in R} \hat{d}_j$). However, due to the time windows, time must be evaluated on every single node, since some deviations might be overshadowed by the opening time windows if we simply fill the knapsack with the first deviation as we do in demand. Finally, the domain of each variable is found in constraints (3.3.13), (3.3.14) and (3.3.15).

3.3.2 CF formulation for the RVRPTW

This model was developed by extending the deterministic CF model (3.1.1)-(3.1.9) to consider uncertainty on demand and travel time using the approach used in the previous formulation. The sets, parameters and binary variables are the same as the ones in model (3.3.4)-(3.3.12), and the new load and time variables are:

- $f_{ij\delta}$: the load of products carried between nodes i and j, with a deviation of δ over the demand's nominal value already considered;
- $g_{ij\delta}$: the arrival time in node j if arc (i,j) is taken, with a deviation of δ over the travel time's nominal value already considered;

Now, it is possible to adapt the model (3.2.3)-(3.2.13) for the new uncertainty set. The new model is as follows:

$$\min \quad \sum_{(i,j)\in A} c_{ij}x_{ij}, \tag{3.3.16}$$

$$\text{s.t.} \quad \sum_{\substack{i=1 \\ i\neq j}}^{n+1} x_{ij} = 1, \qquad j \in N^*, \tag{3.3.17}$$

$$\sum_{\substack{i=0 \\ i\neq h}}^{n} x_{ih} = \sum_{\substack{j=1 \\ j\neq h}}^{n+1} x_{hj}, \qquad h \in N^*, \tag{3.3.18}$$

$$\sum_{h\in N} f_{hi\delta} - \sum_{j\in N} f_{ij\delta} \geq \bar{d}_i, \qquad i \in N, \delta \leq \Delta^d, \tag{3.3.19}$$

$$\sum_{h \in N} f_{hi\delta} - \sum_{j \in N} f_{ij\delta-\hat{d}_i} \geq \bar{d}_i + \hat{d}_i, \qquad i \in N, \hat{d}_i \leq \delta \leq \Delta^d, \tag{3.3.20}$$

$$\sum_{h \in N} f_{hi\Delta^d} - \sum_{j \in N} f_{ij\Delta^d-\lambda} \geq \bar{d}_i + \lambda, \qquad i \in N, \lambda < \hat{d}_i, \tag{3.3.21}$$

$$f_{ij\delta} \leq Qx_{ij}, \qquad i,j \in N, \delta \leq \Delta^d, \tag{3.3.22}$$

$$\sum_{j \in N} g_{ij\delta} \geq \sum_{\substack{h \in N \\ \lambda \leq \hat{t}_{hi}}} (g_{hi\delta-\lambda} + (\bar{t}_{hi} + \lambda)x_{hi}) + s_i, \quad i \in N, \delta \leq \Delta^t, \lambda \leq \delta, \tag{3.3.23}$$

$$(a_j + s_j)x_{ij} \leq g_{ij\delta} \leq (b_j + s_j)x_{ij}, \qquad i,j \in N, \delta \leq \Delta^t, \tag{3.3.24}$$

$$x_{ij} \in \{0,1\}, \qquad (i,j) \in A, \tag{3.3.25}$$

$$f_{ij\delta} \geq 0, \qquad (i,j) \in A, 0 \leq \delta \leq \Delta^d, \tag{3.3.26}$$

$$g_{ij\delta} \geq 0, \qquad (i,j) \in A, 0 \leq \delta \leq \Delta^t. \tag{3.3.27}$$

As expected, this model works similarly to (3.2.3)-(3.2.13) but using the single knapsack uncertainty set and the same approach done in (3.3.4)-(3.3.15). The objective function (3.3.16), again, consists of minimizing the total travel costs. Constraints (3.3.17) and (3.3.18) ensure that each costumer is visited only once and only one vehicle can depart from it. Constraints (3.3.19)-(3.3.21) works similarly to (3.3.7)-(3.3.9), forbidding subtours and computing the demand's worst-case scenario, but using the commodity flow variable. Constraints (3.3.22) are capacity constraints. Constraints (3.3.23) act the same way as (3.3.11) to compute the time, while the time windows constraints are given by (3.3.24). The domain of the variables is defined in (3.3.25)–(3.3.27).

As an additional note, we mention a few difficulties that the models based on single knapsack uncertainty sets might face and their possible workarounds. The first one is the large number of extra variables aaded to the model when Δ grows larger. A first point is made that the value for Δ in real contexts is usually not large. Pessoa et al. (2020), for instance, considered 100 minutes of deviation as an extreme case. This would be considered a highly unstable environment, since the delay would take roughly 20% of a worker's time, it is more likely that the company's time data is inaccurate than that the worker is late this much. A possible workaround is to change the order of magnitude of δ, for instance, instead of using δ as an 1 minute variation, one can use it as 5 minutes for each unity, this way it is possible to reduce the problem's size to some extent, since this particular dynamic programming has complexity $O(n\delta_2)$. On the other hand, the capacity Q has no impact in the computational performance, since it is only called at the end of the algorithm to check if the solution is feasible (i.e., $u_{r_{n+1}\Delta^d} \leq Q$), a $O(1)$ process. However, it is expected that instances with larger capacities have larger knapsacks, which will make the problem hard to solve.

If this adaptation is applied, it is important to change the other parameters accordingly, such as dividing Δ and multiplying λ on constraints (3.3.21) and (3.3.23) by the chosen constant, 5 in our example. For better understanding, we present how the constraints (3.3.23) and (3.3.24) would look if we change the meaning of δ to 5 time units instead of one:

$$\sum_{j \in N} g_{ij\delta} \geq \sum_{\substack{h \in N \\ \lambda \leq \hat{t}_{hi}}} (g_{hi\delta-\lambda} + (\bar{t}_{hi} + 5\lambda)x_{hi}) + s_i, \quad i \in N, \delta \leq \frac{\Delta^t}{5}, \lambda \leq \delta, \tag{3.3.28}$$

$$(a_j + s_j)x_{ij} \leq g_{ij\delta} \leq (b_j + s_j)x_{ij}, \qquad i,j \in N, \delta \leq \frac{\Delta^t}{5}. \tag{3.3.29}$$

These changes can reduce the model to a more tractable size. In a practical sense, using a bigger variation for each δ such as this might be better to understand the solution than a more fractional solution. A second limitation of these formulations is the impossibility of using non-integer units for δ. This can be easily solved by the previous strategy, multiplying it with a constant that turns that value into an integer, with the negative side of possibly worsening the solution times.

3.4 Formulations based on the multiple knapsack uncertainty set

In this section, we present models for the multiple knapsack uncertainty set, the generalization of the single knapsack set. To model this problem, we need a set of knapsacks (S), quadrants in which the nodes (or arcs) are divided with their own individual budget ($\Delta_s, s \in S$), and a set that indicates the nodes in each knapsack ($\mathcal{S}^s, s \in S$). This uncertainty set can be more appropriate in environments where it is possible to identify a shared limitation of the deviation in demand/travel times for $|S|$ sets of nodes/arcs. For example, in an instance where the customers are split in a set of $|S|$ clusters, it might make sense to limit the travel time deviation for each cluster individually, as part of the delay in that route can be related to the travel time required to reach the cluster from the depot.

3.4.1 MTZ-based formulation for the RVRPTW

This model is a extension of (3.3.4)-(3.3.15). To adapt it to the multiple knapsack uncertainty set, we needed to change the load and time variables. We will exemplify the model with a 2-knapsack problem, but it can be easily extended to the k-knapsack problem. The sets are the same as (3.3.4)-(3.3.15) together with S and $\mathcal{S}^s$. The parameters Δ^d and Δ^t are changed to Δ^d_s and Δ^t_s, so we can incorporate the deviation limit for each knapsack. The variables of these models are x_{ij}, which assumes the value of 1 if, and only if, arc (i, j) is traversed. The load and time variables are similar to the ones in the single knapsack formulation, but with an index δ^s for each knapsack. Thus, for the 2-knapsack problem we have:

- $u_{i\delta_1\delta_2}$: the 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;
- $w_{i\delta_1\delta_2}$: the earliest time which is possible to start the service at the node i, with a total deviation of δ_1 over the travel time's nominal value for the first knapsack and δ_2 for the second one;

Figure 3 exemplifies the behavior of the product load variable $u_{i\delta_1\delta_2}$. In this figure, we present a route $R = (r_0, r_1, \ldots, r_4, r_{n+1})$ where each node i in this route has a nominal demand $\bar{d}_i$ and a maximum deviation $\hat{d}_i$. In this example, we define the budget of uncertainty for demand in two different knapsacks, a blue one (with budget Δ^d_1=20), which limits the deviations from customers inside a set S_1, and a red one (with budget Δ^d_2=15), which is related to a customers' subset S_2. In this example, the subset S_1 is composed by the odd-numbered customers ($S_1 = (r_1, r_3)$) while subset S_2 contains the even-numbered customers ($S_2 = (r_2, r_4)$). The blue knapsack is only filled by the deviation from the nodes in S_1, painted in blue in the figure, while the red knapsack considers the demand deviation from nodes in S_2, painted in red. Thus, we can interpret r_1 and r_3 as belonging to a quadrant and r_2 and r_4 being inside another quadrant. Then, for the given route, we compute the value of variables $u_{i\delta_1\delta_2}$ and show the value inside a box that represents the node i. Under each box there are two containers representing the knapsacks, the left one, filled in blue, represents the knapsack from the quadrant that contains the odd customers while the right one, filled in red, does the same but for the even-numbered customers.

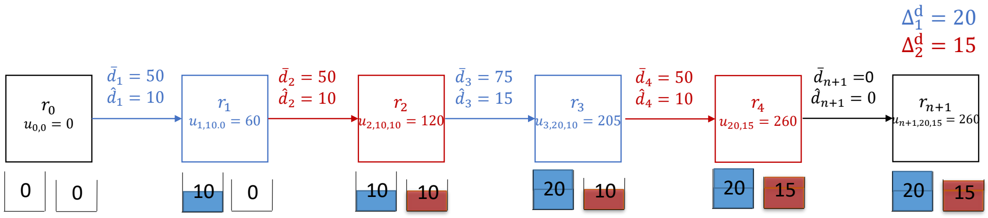


Figure 3 – Example of calculation of load variables for the multiple knapsack uncertainty set.

In this example, we start at the depot r_0 with empty load and knapsacks. Then, at the first customer node in the route, we are able to insert all their deviation ($\hat{d}_1 = 10$) into their respective knapsack (the blue one), thus the total load in that node will be $\bar{d}_1 + \hat{d}_1 = 50 + 10 = 60$. This way, we can represent the total load with the variable $u_{1,10,0} = 60$. Note that any value of δ_1 greater than 10 and any δ_2 will have the same total load of 60, since there is not another source of deviation that could increase the total load in the route up to that node. The second customer node in the route, r_2, has a similar behavior but its deviation is inserted into the second knapsack. Thus, by adding their nominal demand ($\bar{d}_2 = 50$) plus the deviation ($\hat{d}_2 = 10$) into the total load, we result in $u_{2,10,10} = 120$. Now, in r_3, we note that we are unable to insert the whole customer deviation ($\hat{d}_3 = 15$), as there is an excess of 5 units, thus the complete load, after considering the nominal demand ($\bar{d}_3 = 75$), is 205, which can be represented by the variable $u_{3,20,10}$. Note that although there was free space in the second knapsack to insert this additional deviation, we are not allowed to do so because it can only be filled by nodes in its quadrant (in this case, the red one). Finally, with the deviation of the last customer ($\hat{d}_4 = 10$) we completely fill the second knapsack, and some excess of deviation is generated (5 units). Thus, after adding the nominal demand ($\bar{d}_4 = 50$) the final load in the route is 260 units. The load variable associated to this final load in the destination depot, r_{n+1}, is $u_{n+1,\Delta_1^d,\Delta_2^d} = u_{n+1,20,15} = 260$. Thus, with all this information the resulting model is given by:

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}, \tag{3.4.1}$$

$$\text{s.t.} \sum_{\substack{i=1 \\ i\neq j}}^{n+1} x_{ij} = 1, \qquad j \in N^*, \tag{3.4.2}$$

$$\sum_{\substack{i=0 \\ i\neq h}}^{n} x_{ih} = \sum_{\substack{j=1 \\ j\neq h}}^{n+1} x_{hj}, \qquad h \in N^*, \tag{3.4.3}$$

$$u_{j\delta_1\delta_2} \geq u_{i\delta_1\delta_2} + \bar{d}_j + M(x_{ij} - 1), \qquad i, j \in N, \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \tag{3.4.4}$$

$$u_{j\delta_1\delta_2} \geq u_{i\delta_1-\hat{d}_j\delta_2} + \bar{d}_j + \hat{d}_j + M(x_{ij}-1),$$
$$i,j \in N, j \in S_1, j \notin S_2, \hat{d}_j \leq \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \quad (3.4.5)$$

$$u_{j\delta_1\delta_2} \geq u_{i\delta_1\delta_2-\hat{d}_j} + \bar{d}_j + \hat{d}_j + M(x_{ij}-1),$$
$$i,j \in N, j \notin S_1, j \in S_2, \delta_1 \leq \Delta_1^d, \hat{d}_j \leq \delta_2 \leq \Delta_2^d, \quad (3.4.6)$$

$$u_{j\delta_1\delta_2} \geq u_{i\delta_1-\hat{d}_j\delta_2-\hat{d}_j} + \bar{d}_j + \hat{d}_j + M(x_{ij}-1),$$
$$i,j \in N, j \in S_1, j \in S_2, \hat{d}_j \leq \delta_1 \leq \Delta_1^d, \hat{d}_j \leq \delta_2 \leq \Delta_2^d, \quad (3.4.7)$$

$$u_{j\Delta_1^d\delta_2} \geq u_{i\Delta_1^d-\lambda\delta_2} + \bar{d}_j + \lambda + M(x_{ij}-1),$$
$$i,j \in N, \lambda \leq \hat{d}_j, \delta_2 \leq \Delta_2^d, j \in S_1, j \notin S_2, \quad (3.4.8)$$

$$u_{j\delta_1\Delta_2^d} \geq u_{i\delta_1\Delta_2^d-\lambda} + \bar{d}_j + \lambda + M(x_{ij}-1),$$
$$i,j \in N, \lambda \leq \hat{d}_j, \delta_1 \leq \Delta_1^d, j \notin S_1, j \in S_2, \quad (3.4.9)$$

$$u_{j\delta_1\Delta_2^d} \geq u_{i\delta_1-\lambda\Delta_2^d-\lambda} + \bar{d}_j + \lambda + M(x_{ij}-1),$$
$$i,j \in N, \lambda \leq \hat{d}_j, j \in S_1, j \in S_2, \quad (3.4.10)$$

$$u_{j\Delta_1^d\Delta_2^d} \leq Q, \qquad j \in N, \quad (3.4.11)$$

$$w_{j\delta_1\delta_2} \geq w_{i\delta_1-\lambda\delta_2} + \bar{t}_{ij} + s_i + \lambda + M(x_{ij}-1),$$
$$i,j \in N, \lambda \leq \hat{t}_{ij}, \lambda \leq \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t, j \in S_1, j \notin S_2, \quad (3.4.12)$$

$$w_{j\delta_1\delta_2} \geq w_{i\delta_1\delta_2-\lambda} + \bar{t}_{ij} + s_i + \lambda + M(x_{ij}-1),$$
$$i,j \in N, \lambda \leq \hat{t}_{ij}, \delta_1 \leq \Delta_1^t, \lambda \leq \delta_2 \leq \Delta_2^t, j \notin S_1, j \in S_2, \quad (3.4.13)$$

$$w_{j\delta_1\delta_2} \geq w_{i\delta_1-\lambda\delta_2-\lambda} + \bar{t}_{ij} + s_i + \lambda + M(x_{ij}-1),$$
$$i,j \in N, \lambda \leq \hat{t}_{ij}, \lambda \leq \delta_1 \leq \Delta_1^t, \lambda \leq \delta_2 \leq \Delta_2^t, j \in S_1, j \in S_2, \quad (3.4.14)$$

$$a_j \leq w_{j\delta_1\delta_2} \leq b_j, \qquad i,j \in N, \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t, \quad (3.4.15)$$

$$x_{ij} \in \{0,1\}, \qquad i,j \in N, \quad (3.4.16)$$

$$u_{i\delta_1\delta_2} \geq 0, \qquad i \in N\delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \quad (3.4.17)$$

$$w_{i\delta_1\delta_2} \geq 0, \qquad i \in N, \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t. \quad (3.4.18)$$

Similarly to the previous models, the objective function (3.4.1) minimizes the total travelling costs. Constraints (3.4.2) and (3.4.3) ensure that every costumer is visited and that only one vehicle departs from it. Constraints (3.4.4)-(3.4.10) are responsible for computing the demand for the worst-case scenario and forbid subtours. The load with a set δ_1 and δ_2 in each node will be the greatest between considering deviation in the node, computed by (3.4.5)-(3.4.10), or not, computed by (3.4.4) by just adding the nominal demand.

Constraints (3.4.5)-(3.4.7) work similarly to (3.3.8), where the algorithm tries to insert the maximum demand in the node, if there is enough space in the knapsack, to compute the worst-case demand. If the node belongs only to knapsack 1, only δ_1 is affected as we have in (3.4.5). Similarly, if the node is in the knapsack 2, only δ_2 is affected and constraints (3.4.6) are used. Finally, if the node belongs to both knapsacks, the line (3.4.7) is used. If the node does not belong to any knapsack, we do not consider any deviation and simply use constraints (3.4.4). Constraints (3.4.8), (3.4.9) and (3.4.10) are similar to (3.4.5), (3.4.6) and (3.4.7), respectively, but for when the deviation in the node is greater than the remaining space in the knapsack, which will be filled. The vehicle's capacity is ensured by constraints (3.4.11).

Constraints (3.4.12)-(3.4.14) are similar to (3.4.8)-(3.4.10), but for time and, due to how the opening time windows require to be always verified or otherwise we might consider a deviation that will be overshadowed by it, for all combination of $\delta_1 \leq \Delta_1^t$ and $\delta_2 \leq \Delta_2^t$. The time windows' constraints are found in (3.4.15) and the domain of variables is defined in (3.4.16)-(3.4.18). Note that we can easily extend this model for k knapsacks, by adding indices up to δ^k, although the number of constraints grows as a function of k. In Appendix A.1, we present a model for the 3-knapsack uncertainty set as an example.

3.4.2 CF formulation for the RVRPTW

Similarly to how was done for the model with MTZ constraints, we extend formulation (3.3.16)-(3.3.27) to consider multiple knapsacks. As in the last topic, we exemplify the model with a 2-knapsack uncertainty set. This formulation can also be extended for the k-knapsack uncertainty set, and an example for the model using the 3-knapsack uncertainty set is found in Appendix A.2.

This formulation uses exactly the same sets and parameters as (3.4.1)-(3.4.16). This model also uses the same binary variables x_{ij}, which represent if arc (i,j) is taken, while the load and time variables are changed to:

- $f_{ij\delta_1\delta_2}$: the load of products carried in the vehicle while traversing arc (i,j), with a deviation of δ_1 over the demand's nominal value already considered for the first knapsack and δ_2 for the second one;

- $g_{ij\delta_1\delta_2}$: the arrival time in node j if arc (i,j) is taken considering a deviation of δ_1 over the travel time's nominal value already considered for the first knapsack and δ_2 for the second one;

The resulting model is given by:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij}, \tag{3.4.19}$$

$$\text{s.t.} \sum_{\substack{i=1\\ i\neq j}}^{n+1} x_{ij} = 1, \qquad j \in N^*, \tag{3.4.20}$$

$$\sum_{\substack{i=0\\ i\neq h}}^{n} x_{ih} = \sum_{\substack{j=1\\ j\neq h}}^{n+1} x_{hj}, \qquad h \in N^*, \tag{3.4.21}$$

$$\sum_{h\in N} f_{hi\delta_1\delta_2} - \sum_{j\in N} f_{ij\delta_1\delta_2} \geq \bar{d}_i, \qquad i \in N, \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \tag{3.4.22}$$

$$\sum_{h\in N} f_{hi\delta_1\delta_2} - \sum_{j\in N} f_{ij\delta_1-\hat{d}_i\delta_2} \geq \bar{d}_i + \hat{d}_i,$$
$$i \in N, i \in S_1, i \notin S_2, \hat{d}_i \leq \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \tag{3.4.23}$$

$$\sum_{h\in N} f_{hi\delta_1\delta_2} - \sum_{j\in N} f_{ij\delta_1\delta_2-\hat{d}_i} \geq \bar{d}_i + \hat{d}_i,$$
$$i \in N, i \notin S_1, i \in S_2, \delta_1 \leq \Delta_1^d, \hat{d}_i \leq \delta_2 \leq \Delta_2^d, \tag{3.4.24}$$

$$\sum_{h \in N} f_{hi\delta_1\delta_2} - \sum_{j \in N} f_{ij\delta_1 - \hat{d}_i \delta_2 - \hat{d}_i} \geq \bar{d}_i + \hat{d}_i,$$
$$i \in N, i \in S_1, i \in S_2, \delta_1 \leq \Delta_1^d, \hat{d}_i \leq \delta_2 \leq \Delta_2^d, \tag{3.4.25}$$

$$\sum_{h \in N} f_{hi\Delta_1^d\delta_2} - \sum_{j \in N} f_{ij\Delta_1^d - \lambda\delta_2} \geq \bar{d}_i + \lambda,$$
$$i \in N, i \in S_1, i \notin S_2, \delta_2 \leq \Delta_2^d, \lambda < \hat{d}_i, \tag{3.4.26}$$

$$\sum_{h \in N} f_{hi\delta_1\Delta_2^d} - \sum_{j \in N} f_{ij\delta 1\Delta_2^d - \lambda} \geq \bar{d}_i + \lambda,$$
$$i \in N, i \notin S_1, i \in S_2, \delta_1 \leq \Delta_1^d, \lambda < \hat{d}_i, \tag{3.4.27}$$

$$\sum_{h \in N} f_{hi\Delta_1^d\Delta_2^d} - \sum_{j \in N} f_{ij\Delta_1^d - \lambda\Delta_2^d - \lambda} \geq \bar{d}_i + \lambda,$$
$$i \in N, i \in S_1, i \in S_2, \lambda < \hat{d}_i, \tag{3.4.28}$$

$$f_{ij\delta_1\delta_2} \leq Qx_{ij}, \qquad i, j \in N, \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \tag{3.4.29}$$

$$\sum_{j \in N} g_{ij\delta_1\delta_2} \geq \sum_{\substack{h \in N \\ \lambda \leq \hat{t}_{hi}}} (g_{hi\delta_1 - \lambda\delta_2} + (\bar{t}_{hi} + \lambda)x_{hi}) + s_i,$$
$$i \in N, i \in S_1, i \notin S_2, \lambda \leq \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t, \tag{3.4.30}$$

$$\sum_{j \in N} g_{ij\delta_1\delta_2} \geq \sum_{\substack{h \in N \\ \lambda \leq \hat{t}_{hi}}} (g_{hi\delta_1\delta_2 - \lambda} + (\bar{t}_{hi} + \lambda)x_{hi}) + s_i,$$
$$i \in N, i \notin S_1, i \in S_2, \delta_1 \leq \Delta_1^t, \lambda \leq \delta_2 \leq \Delta_2^t, \tag{3.4.31}$$

$$\sum_{j \in N} g_{ij\delta_1\delta_2} \geq \sum_{\substack{h \in N \\ \lambda \leq \hat{t}_{hi}}} (g_{hi\delta_1 - \lambda\delta_2 - \lambda} + (\bar{t}_{hi} + \lambda)x_{hi}) + s_i,$$
$$i \in N, i \in S_1, i \in S_2, \lambda \leq \delta_1 \leq \Delta_1^t, \lambda \leq \delta_2 \leq \Delta_2^t, \tag{3.4.32}$$

$$(a_j + s_j)x_{ij} \leq g_{ij\delta_1\delta_2} \leq (b_j + s_j)x_{ij}, \qquad i, j \in N, \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t \tag{3.4.33}$$

$$x_{ij} \in \{0, 1\}, \qquad (i, j) \in A, \tag{3.4.34}$$

$$f_{ij\delta_1\delta_2} \geq 0, \qquad (i, j) \in A, \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \tag{3.4.35}$$

$$g_{ij\delta_1\delta_2} \geq 0, \qquad (i, j) \in A, \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t. \tag{3.4.36}$$

The objective function (3.4.19) and the vehicle flow constraints (3.4.20)-(3.4.21) are exactly the same from previous models. Constraints (3.4.22)-(3.4.28) compute the worst total capacity load in the same fashion as constraints (3.4.4)-(3.4.10) from the MTZ model. Constraints (3.4.22) compute for the case when we do not consider the demand deviation in node i, while constraints (3.4.23)-(3.4.25) are activated when the whole demand deviation fits in the knapsack in which i is part of. The only difference among (3.4.23)-(3.4.25) is the choice of which knapsack is filled, this is defined by the sets S_1 and S_2. If i belongs to subset S_1 only, constraints (3.4.23) are activated; if it only belongs to S_2, then constraints (3.4.24) are activated; finally, if i is in both sets, then (3.4.25) become active. Constraints (3.4.26)-(3.4.28) are also used to compute the total capacity load, but for the case the total deviation of node i does not fit into the knapsack, thus we fill it the maximum possible only. Capacity constraints are given by (3.4.29).

Similarly to (3.4.12)-(3.4.14), constraints (3.4.30)-(3.4.32) compute the total elapsed time. Like the demand constraints, each one of these are activated based on which knapsack arc (i, j) belongs to. The first constraint set is activated if the arc belongs to the first knapsack only; the second is activated if the arc belongs to the second knapsack; and the third constraint set is activated if it belongs to both knapsacks. The time-windows are enforced in constraints (3.4.33). Finally, the domain of variables is found in (3.4.34)-(3.4.36).

3.5 Branch-and-Cut

In addition to the compact formulations, we designed a tailored branch-and-cut (B&C) algorithm so as to obtain a better computational performance when solving the studied models. The cut separation was done using a combination of strategies: a dynamic programming based algorithm, a heuristic algorithm created for the RCVRP (Gounaris; Wiesemann; Floudas, 2013) and the CVRPSEP package (Lysgaard; Letchford; Eglese, 2004). The latter is a well known library developed for generating valid inequalities for the deterministic CVRP. We use this library to generate rounded capacity cuts only, which are stated as follows for a given set of nodes V_S:

$$\sum_{i \in N \setminus V_S} \sum_{j \in V_S} x_{ij} \geq \left\lceil \frac{1}{Q} \sum_{j \in V_S} \bar{d}_j \right\rceil, \quad V_S \subset N. \tag{3.5.1}$$

These cuts ensure that the number of vehicles entering set V_S guarantee enough capacity to service all nodes in V_S. Since enumerating every possible set S would demand a long computational time, the CVRPSEP uses an efficient heuristic algorithm to generate a limited number of sets V_S and relevant cuts.

Other than that, for instances with uncertainty on demand we also use the Robust Rounded Capacity Inequality (RCI) proposed by Gounaris, Wiesemann and Floudas (2013). These cuts are a robust extension of the capacity cut previously explained, represented by the following inequalities:

$$\sum_{i \in N \setminus V_S} \sum_{j \in V_S} x_{ij} \geq \left\lceil \frac{1}{Q} * \max_{d \in U^d} \sum_{j \in V_S} d_j \right\rceil, \quad V_S \subset C. \tag{3.5.2}$$

Note that these are almost identical to the deterministic counterpart, the main difference being that, in the robust one, we consider the maximum possible demand inside the uncertainty set instead of the nominal demand. Similarly to the deterministic capacity cuts, enumerating every possible RCI cut would be impractical and therefore we implemented a heuristic algorithm to dynamically insert these cuts as needed, similar to the one proposed by Gounaris, Wiesemann and Floudas (2013). We start with a solution and a randomly generated set of customers V_S, then we interactively perturb this set by inserting or removing a node from it. Every time we want to add/remove a node from set V_S, we analyze every possible 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 RCI 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 while they are in the list. We use this list to avoid cycles. We stop the algorithm when we do not improve the difference between the right-hand side and the left-hand side 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 it requires this value to be frequently checked. To assist this calculation, an auxiliary data structure should be created in order to minimize the number of steps calculated.

For the cardinality constrained uncertainty set, we defined an auxiliary array ($\mathcal{Q}$) containing the demand deviation of all nodes in V_S in descending order. We then compute the maximum demand of this set D_{V_S} by computing the following formula

$$D_{V_S} = \sum_{j \in S} \bar{d}_j + \sum_{\gamma=0}^{\Gamma^d} \mathcal{Q}_\gamma \tag{3.5.3}$$

Then, for each iteration when we check the possibility of insertion/removal of a given node j, we take the maximum 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 expressions:

$$RS_j = D_{V_S} + \bar{d}_j + \max(\hat{d}_j - \mathcal{Q}_{\Gamma^d}, 0), \quad j \notin V_S; \tag{3.5.4}$$

$$RS_j = D_{V_S} - \bar{d}_j - \max(\hat{d}_j - \mathcal{Q}_{\Gamma^d+1}, 0), \quad j \in V_S. \tag{3.5.5}$$

After choosing which node will be inserted/removed from set V_S, we update the D_{V_S} with the right side value (RS_j) from that node and insert/remove its deviation from the auxiliary structure. Note that we do not need to compute D_{V_S} again with the equation 3.5.3, 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, Wiesemann and Floudas (2013), which calculate the maximum demand deviation in this particular uncertainty set in $\mathcal{O}(|V_S|)$. Let S be the set of knapsacks and S^s the subset of nodes inside each knapsack $s \in S$. The right-hand side of the RCI inequality is given by:

$$\sum_{i \in V_S} \bar{d}_i + \sum_{s \in S} \min(\Delta_s^d, \sum_{\substack{i \in V_S \\ i \in S^s}} \hat{d}_i). \tag{3.5.6}$$

Since this is a heuristic algorithm, it might miss a violated constraint. Thus, we also check an integer solution, using a dynamic programming algorithm based on the recursive equations (2.3.86). Whenever the RCI is unable to find a new violated cut, this algorithm checks if the solution is feasible and insert additional feasibility cuts if needed. If a given route $r = (v_0, v_1, \ldots, v_h, \ldots, v_n)$ is infeasible in node v_h the following inequality is inserted into the problem:

$$\sum_{0 < i \leq |h|} y_{v_{i-1}v_i} < |h|. \tag{3.5.7}$$

For the variants with uncertainty on time, we used a different approach. We initialize the model with only the deterministic time flow constraints and dynamically insert the robust constraints (3.2.9) and (3.2.10) whenever they are violated. To efficiently check the feasibility of a solution, we constructed an algorithm based on dynamic programming. This algorithm uses recursive equations similar to (2.3.90) and (3.3.3) to compute the maximum possible time and identify if every time windows is respected in the route.

3.6 Computational results

In this section, we present the computational results of the proposed RO models using benchmark instances from the literature. We analyze the impact of robustness using the cardinality constrained uncertainty set and compare the computational performance of the proposed model against the state-of-the-art compact model for the RVRPTW. We also analyze the performance of the designed B&C algorithm and compared the results when using it with both models. We also studied the performance with the proposed compact formulations and B&C algorithm for the single-knapsack uncertainty set and analyzed the impacts of using this set regarding robustness and computational efficiency. All computational experiments were run in a computer with processor Intel Core i7-8700 CPU @ 3.60 MHz and 16GB of RAM, using the solver IBM CPLEX Optimization Studio v.12.10. We imposed a time limit of 3600 seconds in the experiments.

3.6.1 Instances description

In our computational experiments we use all the instances with 25 customers from the benchmark proposed by Solomon (1987). Solomon's benchmark is composed by six different sets of instances: C1, C2, R1, R2, RC1, RC2. Classes C1 and C2 are defined by having the nodes in clusters, R1 and R2 nodes are randomly distributed and classes RC1 and RC2 are a mix of random and clustered structures. Moreover, problem sets C1, R1 and RC1 differ from C2, R2 and RC2 by having shorter capacities and scheduling horizons, which results in only a few customers per route.

We then adapted these instances to include uncertainty in demands and travel times using a similar approach to Munari et al. (2019). For the *cardinality constrained* uncertainty set, we created new parameters for every instance representing the budget of uncertainty for demand (Γ^d) and travel time (Γ^t). These parameters can assume the values of 0, 1, 5 and 10, in which 0 is the deterministic case. The new instances were divided in three groups based on these parameters: instances in which there is only uncertainty on demand ($\Gamma^d > 0, \Gamma^t = 0$), instances that only consider uncertainty on travel time ($\Gamma^d = 0, \Gamma^t > 0$) and instances in which we consider uncertainty on both parameters ($\Gamma^d > 0, \Gamma^t > 0$). In this last group, the budgets for both parameters are the same, i.e., $\Gamma^d = \Gamma^t$. To generate the *single knapsack* uncertainty set instances, a similar strategy was employed, but we assume $\Delta^q = \{0, 20, 40, 60\}$ and $\Delta^t = \{0, 20, 40, 60\}$.

We also defined the nominal value and maximum deviation for each uncertain parameter. We used the original demands and travel times as their nominal values and the product of the nominal value with a chosen constant to create the maximum deviation. We considered the maximum percentage deviations for demand (Dev^d) and travel time (Dev^t) to be 10%, 25% and 50% of the nominal value, truncated on the first decimal place. Particularly, if the budget of uncertainty for a specific parameter in the instance is zero, the maximum deviation for that parameter (Dev^d or Dev^t) will also be zero. If the budget for both parameters is not zero, Dev^d and Dev^t will have the same value. Thus, by having every possible combination of Γ^d (or Δ^d), Γ^t (or Δ^t), and deviations, each original deterministic instance originated 27 new ones. We present more complete tables for the computational results of the experiments with the RO models in Appendix B.1 and B.2.

3.6.2 Computational performance of the compact formulations

In this subsection, we compare the performance of the commodity flow formulation and the MTZ-based formulation in a general-purpose MIP solver, considering both the cardinality constrained and the knapsack uncertainty sets.

3.6.2.1 Cardinality constrained uncertainty set

We first compare the computational performance of the models across the different set of instances studied. To do so, we solved the same instances for all proposed formulations and compared their results. Table 2 summarizes the results for all combinations of Γ^d and Γ^t for both the commodity flow formulation (CF) and the MTZ-based formulation (MTZ) of Munari et al. (2019) using CPLEX default parameters. In this table, we present the average objective value (*Obj*) and the average computational time in seconds (*T*) for the solutions obtained with the linear programming (LP) relaxation and the MIP models. Furthermore, to assist the analysis of the LP relaxation solutions, we also present the *quality of the LP relaxation* (*QLR*), which is the percentage that the value of the solution of the LP relaxation is in relation to the integer solution value. The closer this parameter is to 100%, the stronger the LP relaxation is. For the MIP models, the table also shows the number of instances that were solved to optimality (*Opt*) and the average optimality gap (*Gap*), given by the average of the relative differences between the lower bound and the objective value of the best integer solution obtained by CPLEX. This last value is larger than zero if the solver was unable to prove optimality within the time limit of 3600 seconds. It is worth mentioning that we have 56 instances for the deterministic case (when $\Gamma^d = \Gamma^t = 0$) and 168 instances for the other combinations of Γ^d and Γ^t.

Table 2 – Average objective values and computational times of the solutions obtained from the MTZ and CF formulations based on the uncertainty budgets's size.

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

Regarding the LP relaxation of the formulations, we observe that in general CPLEX was considerably faster with the MTZ model, whereas the CF model resulted in significantly stronger linear relaxation. On average, the value of the linear relaxation of the CF model is 77.7% of the integer solution against only 51.9% from the MTZ formulation. This behavior was expected, since the CF formulations in literature are known for having tighter LP relaxations than those that use constraints with big-M parameters (Letchford; Salazar-González, 2015).

For the mixed-integer programming formulations, using the MTZ model resulted in the shortest solution times on average and more instances were solved to optimality. A notable exception regarding MTZ model's lead happens on the deterministic instances ($\Gamma^d = \Gamma^t = 0$), as the CF formulation had a significantly better performance. The CF model also outperformed the MTZ model in terms of computational times and number of optimal solutions in instances where the uncertainty budgets (Γ^d and Γ^t) are equal to or less than 1. Furthermore, the average optimality *gaps* in the majority of combinations of Γ^d and Γ^t. That means, when the MTZ model does not solve an instance to optimality, the *gaps* are considerably larger than the CF formulation, which usually is closer to prove optimality of solutions in this situation, which is explained by the stronger linear relaxation of the CF model.

We also identified the traditional behavior found in the majority of works from RO literature (Ordonez, 2010; Agra et al., 2013; Munari et al., 2019) where increasing the budget's size also tends to increase the time it takes to solve the problem. This is expected, since the size of the uncertainty budget directly impacts on the number of variables and constraints of the formulations. This behavior can be identified in both models and for any combination of Γ^t and Γ^d.

Now, it is interesting to evaluate the behavior of the models regarding the uncertainty level. To do this, we created Table 3, which has the same structure as Table 2 but instead of showing the results for all combinations of Γ^d and Γ^t, we present the results for the combinations of deviation in demand (Dev^d) and time (Dev^t).

Table 3 – Average objective values and computational times of the solutions obtained from the MTZ and CF formulations based on the deviation.

		LP Relaxation						**MIP Model**							
		MTZ			**CF**			**MTZ**				**CF**			
$\boldsymbol{Dev^d}$	$\boldsymbol{Dev^t}$	**Obj**	**QLR**	**T (s)**	**Obj**	**QLR**	**T (s)**	**Obj**	**Gap**	**T (s)**	**Opt**	**Obj**	**Gap**	**T (s)**	**Opt**
0%	**0%**	177.23	53.5%	0.011	267.43	80.7%	0.030	331.27	3.7%	577.04	48	331.27	0.7%	282.68	52
10%	**0%**	177.23	52.4%	0.021	267.67	79.2%	0.234	337.94	5.7%	798.89	138	338.53	2.2%	921.88	134
25%	**0%**	177.23	51.1%	0.021	268.38	77.3%	0.247	347.12	6.7%	956.96	132	347.68	3.8%	1112.17	122
50%	**0%**	177.25	50.2%	0.021	270.42	76.6%	0.268	353.04	7.5%	994.17	129	353.62	4.1%	1222.65	119
0%	**10%**	177.43	53.3%	0.024	267.78	80.4%	0.409	332.95	5.0%	750.79	139	333.21	1.7%	812.86	138
0%	**25%**	177.65	52.5%	0.025	268.56	79.4%	0.356	338.34	4.2%	679.25	143	337.00	1.7%	842.41	139
0%	**50%**	177.31	52.9%	0.025	251.50	75.0%	0.361	335.28	4.6%	667.71	127	332.80	2.6%	926.58	120
10%	**10%**	177.43	52.2%	0.045	268.01	78.8%	1.521	340.07	6.2%	946.30	133	341.28	3.7%	1366.64	118
25%	**25%**	177.65	50.4%	0.045	269.45	76.4%	1.501	352.68	7.1%	1040.06	128	354.46	5.7%	1517.34	108
50%	**50%**	176.80	50.6%	0.042	254.14	72.7%	1.569	349.41	7.4%	954.37	118	351.15	6.5%	1524.75	92
Total		177.32	51.9%	0.028	265.33	77.7%	0.650	341.81	5.8%	836.55	1235	342.10	3.3%	1055.29	1142

We note that, in this table, similar conclusions to the ones found in Table 2 can be inferred regarding the quality of the linear relaxation and average gaps, in which tests with the CF model presented a better QLR and lower gaps in all combinations of Dev^d and Dev^t. We also identified that the experiments with the MTZ model provided more optimal solutions than the CF formulation for all pairs of deviation, with exception of the deterministic case. This differs from the results sorted in function of Γ^d and Γ^t, in which the solver was best with the CF model in some of combinations of budgets. We also noticed that, unlike the uncertainty budget's size, the models do not have an uniform behavior on computational times when deviation changes. For example, while the solver's average running times with the standalone MTZ model increased with higher deviations in instances with deviation exclusively on demand, instances with deviation exclusively on travel times had decreasing computational times. On the other hand, the model took longer to solve instances using the CF model as we increase Dev^d and Dev^t, but with lower impact than changes in the budget's size (Γ^d and Γ^t). This is due to the fact that, the uncertainty budget directly affect the number of constraints and variables of the problem, while the deviation level just change the worst-case value parameters.

Finally, we studied the behavior of both models for the individual instance classes from

Solomon (1987). To assist in this analysis we created Tables 4 and 5, which present the average results of the MTZ and CF models, respectively. These tables follow a similar structure to the previous ones.

The best performance for both models is observed on instances with clustered customers (classes C1 and C2), where the MTZ formulation solved 88.02% (419) of instances to optimality, with an average running time of 511.40 seconds while the CF model solved 93.48% (435) of the instances with an average running time of 394.34 seconds. The main reason for this behavior is that a considerably number of arcs connecting nodes from different clusters in these instances are either infeasible or expensive, and thus they are less likely to be used in a LP relaxation and even if they are, the branch-and-bound algorithm can cut them off quickly.

Conversely, instances with mixed geographic distribution of customers (classes RC1 and RC2) were the most challenging for the models, the MTZ formulation solved only 48.43% (217) of instances to optimality with an average running time of 1931.08 seconds while the CF model solved 52.67% (236) of them with an average time of 1734.34 seconds. These instances were harder mainly because they have a greater variety of feasible arcs with smaller trade-offs than the ones from Class C1 and C2, making it harder to cutting off nodes from the branch-and-cut tree.

Using the MTZ formulation also performed particularly well on instances with randomly distributed customer, especially class R2. With this model, 87.20% (293) and 99.35% (306) of the instances from classes R1 and R2 were solved to optimality with average running times equal to 360.12 and 266.77 seconds, respectively. This was significantly better than the results of the CF formulation, which solved 77.38% (260) and 68.50% (211) of the instances in classes R1 and R2 with average running times of 825.85 and 1341.52 seconds, in that order.

These results can be better visualised with the help of Figure 4. This figure shows the number of instances each model resulted in an optimal solution within the time limit (*Opt*) per instance class and uncertainty budget. The hatched columns represent the results using the CF formulation whereas those completely filled show the number of optimal solutions obtained with the MTZ model. The closer a column is from the horizontal green bar, which represents the total number of instances tested for that combination of instance class and budget size, the better.

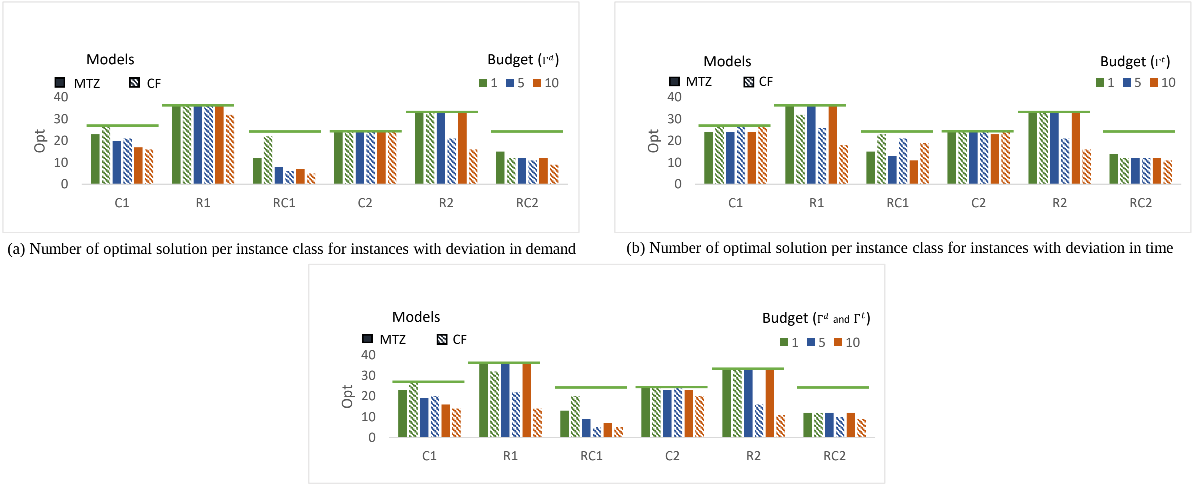
It is possible to see that, despite the generally better results from the MTZ model, using the CF model actually outperformed the use of the MTZ formulation in some instance classes, namely C1 and RC1. We also note that, other than class RC2, the CF formulation resulted in optimality for all instances in the deterministic case (Γ^d=Γ^t=0), while optimality was not proven for five of these instances with the MTZ formulation. Thus, we can infer that the MTZ formulation can be more suitable for companies with few clients dispersed over a given geographical area, whereas CF model adapts better to the situations where we have a greater number of clustered clients. Other than checking clients dispersion, the decision maker should also consider the budget of uncertainty size when choosing which model should be used to solve a particular instance. When considering a large budget, such as $\Gamma = 10$, it might be advisable to use the MTZ formulation to solve the problem, even if the customers are distributed in clusters. On the other hands, it will probably be quicker to solve the problem using the CF formulation if the decision maker is evaluating the deterministic case or a problem with a small uncertainty budget (e.g. $\Gamma = 1$), independently of the customers distribution.

Table 4 – Average results of the robust MTZ model with different values of budgets of uncertainty Γ using instances from classes C1, R1 and RC1 with 25 customers.

		MTZ																	
		C1						**R1**						**RC1**					
Γ^d	Γ^t	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**
0	0	190.59	0.0%	419.28	9	8	0	463.37	0.0%	128.72	12	12	0	350.24	10.6%	1802.80	8	4	0
1	0	202.68	0.0%	578.20	27	23	0	463.37	0.0%	249.80	36	36	0	360.14	13.2%	1836.27	24	12	0
5	0	238.59	0.0%	1089.52	27	20	0	463.37	0.7%	417.86	36	33	0	432.49	25.8%	2713.30	24	8	0
10	0	245.18	0.4%	1426.92	27	17	0	463.37	0.8%	648.14	36	33	0	439.15	29.0%	2745.56	24	7	0
0	1	192.34	0.0%	446.19	27	24	0	470.33	0.0%	153.23	36	32	4	356.87	8.5%	1460.90	24	15	1
0	5	195.83	0.7%	532.82	27	24	0	478.40	0.7%	349.07	36	29	4	383.14	13.6%	1570.08	24	13	1
0	10	195.83	2.5%	706.38	27	24	0	479.60	1.2%	411.35	36	29	4	392.71	15.9%	1884.34	24	11	1
1	1	202.86	0.0%	587.00	27	23	0	470.33	0.0%	216.44	36	32	4	368.53	11.8%	1623.02	24	13	1
5	5	239.11	1.5%	1231.88	27	19	0	478.77	1.2%	403.11	36	29	4	444.47	23.3%	2318.55	24	9	1
10	10	245.73	4.8%	1576.89	27	16	0	479.38	1.8%	623.48	36	28	4	448.29	26.9%	2577.59	24	7	1
All		214.87	1.0%	859.51	252	198	0	471.03	0.7%	360.12	336	293	24	397.60	17.9%	2053.24	224	99	6
		C2						**R2**						**RC2**					
Γ^d	Γ^t	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**
0	0	214.45	0.0%	13.28	8	8	0	382.15	0.0%	41.89	11	11	0	319.28	14.1%	1500.82	8	5	0
1	0	214.45	0.0%	17.20	24	24	0	382.15	0.0%	48.84	33	33	0	319.28	15.0%	1725.07	24	15	0
5	0	214.45	1.5%	52.44	24	24	0	382.15	0.0%	162.38	33	33	0	319.28	16.9%	1855.66	24	12	0
10	0	214.45	3.4%	106.79	24	24	0	382.15	0.0%	382.82	33	33	0	319.34	19.2%	1925.83	24	12	0
0	1	214.51	0.0%	17.81	24	24	0	383.86	0.0%	53.17	33	33	0	319.68	14.5%	1665.70	24	14	0
0	5	214.57	1.8%	164.66	24	24	0	384.80	0.0%	148.04	33	33	0	320.00	17.1%	1821.75	24	12	0
0	10	214.57	4.0%	245.14	24	23	0	384.89	0.0%	329.39	33	33	0	320.07	18.6%	1857.75	24	12	0
1	1	214.51	0.0%	24.22	24	24	0	383.86	0.0%	65.53	33	33	0	319.55	16.3%	1822.72	24	12	0
5	5	214.57	3.0%	229.61	24	23	0	384.80	0.0%	404.69	33	33	0	319.94	19.1%	1864.07	24	12	0
10	10	214.57	7.6%	326.61	24	23	0	384.92	0.2%	1031.00	33	31	0	320.38	19.4%	2049.80	24	12	0
All		214.51	2.1%	119.78	224	221	0	383.57	0.0%	266.77	308	306	0	319.68	17.0%	1808.92	224	118	0

Table 5 – Average results of the robust CF model with different values of budgets of uncertainty Γ using instances from classes C1, R1 and RC1 with 25 customers.

		CF																	
		C1						**R1**						**RC1**					
Γ^d	Γ^t	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**
0	0	190.59	0.0%	0.61	9	9	0	463.37	0.0%	30.99	12	12	0	350.24	0.0%	3.14	8	8	0
1	0	202.68	0.0%	4.40	27	27	0	463.37	0.0%	46.12	36	36	0	360.14	0.7%	360.84	24	22	0
5	0	238.54	1.7%	1023.78	27	21	0	463.37	0.0%	297.91	36	36	0	433.79	13.3%	2804.37	24	6	0
10	0	245.57	5.4%	1644.64	27	16	0	463.45	0.4%	1048.73	36	32	0	443.55	16.7%	2895.70	24	5	0
0	1	192.34	0.0%	1.53	27	27	0	469.41	0.0%	84.22	36	32	4	355.20	0.0%	6.64	24	23	1
0	5	195.83	0.0%	9.14	27	27	0	477.18	0.7%	1188.49	36	26	4	370.67	1.0%	426.15	24	21	1
0	10	195.83	0.0%	61.42	27	27	0	478.48	2.5%	1785.62	36	18	4	378.80	3.1%	811.90	24	19	1
1	1	202.86	0.0%	6.18	27	27	0	469.41	0.0%	128.27	36	32	4	367.58	1.3%	582.62	24	20	1
5	5	239.02	2.1%	1285.61	27	20	0	477.40	1.5%	1524.88	36	22	4	446.09	17.2%	2737.15	24	5	1
10	10	247.62	10.0%	1910.34	27	14	0	479.56	4.8%	2123.29	36	14	4	456.60	20.5%	2970.52	24	5	1
All		215.09	1.9%	594.77	252	215	0	470.50	1.0%	825.85	336	260	24	396.26	7.4%	1359.90	224	134	6
		C2						**R2**						**RC2**					
Γ^d	Γ^t	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**
0	0	214.45	0.0%	1.94	8	8	0	382.15	0.0%	83.19	11	11	0	319.28	5.0%	1812.10	8	4	0
1	0	214.45	0.0%	2.83	24	24	0	382.15	0.0%	180.54	33	33	0	319.28	5.6%	1817.78	24	12	0
5	0	214.45	0.0%	32.42	24	24	0	382.49	1.5%	1445.52	33	21	0	319.50	7.9%	2141.31	24	11	0
10	0	214.45	0.0%	203.66	24	24	0	384.01	3.4%	2070.16	33	16	0	321.91	10.7%	2359.64	24	9	0
0	1	214.51	0.0%	4.08	24	24	0	383.86	0.0%	230.42	33	33	0	319.64	6.0%	1819.76	24	12	0
0	5	214.57	0.0%	42.76	24	24	0	385.34	1.8%	1624.50	33	21	0	320.68	8.3%	1953.42	24	12	0
0	10	214.57	0.0%	277.31	24	24	0	387.36	4.0%	2356.57	33	16	0	323.15	10.3%	2271.57	24	11	0
1	1	214.51	0.0%	7.00	24	24	0	383.86	0.0%	414.57	33	33	0	319.73	6.6%	1824.06	24	12	0
5	5	214.57	0.0%	151.98	24	24	0	386.13	3.0%	2160.03	33	16	0	323.10	10.2%	2309.99	24	10	0
10	10	214.63	1.0%	934.88	24	20	0	393.21	7.6%	2849.72	33	11	0	328.01	15.2%	2778.25	24	9	0
All		214.51	0.1%	165.89	224	220	0	385.05	2.1%	1341.52	308	211	0	321.43	8.6%	2108.79	224	102	0



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

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

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

Figure 4 – Number of optimal solutions obtained by the robust models per instance class and uncertainty budget.

3.6.2.2 Knapsack uncertainty set

Similarly to the previous uncertainty set, we first compare the proposed formulations regarding computational performance. Table 6 presents, for every combination of Δ^d and Δ^t, the average objective solutions (Obj) and running times (T) of the LP relaxation and integer solutions. Moreover, for the LP relaxation the table shows the quality of the LP relaxation (QLR). Finally, for the solutions obtained with the MIP model, this table also presents the average difference between the lower bound and the best integer solution (Gap), which is greater than zero only if the solver was not able to obtain an optimal solution within the time limit of 3600 seconds, and the number of optimal solutions (Opt).

Table 6 – LP Relaxation and Integer Solution of the MTZ and CF formulations based on the single knapsack uncertainty set based on the uncertainty budgets' size

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

Similar conclusions as with the previous uncertainty set can be drawn from this table regarding the quality of the linear relaxation and computational performance of the models. Particularly, the CF formulation presented stronger linear relaxations overall, with a QLR of 79.9%, but presented worse results as a MIP model, since it resulted in less instances solved to optimality (263) than the MTZ model (484), in longer running times (2360.35 seconds against 1660.28 seconds from the MTZ model) and in greater average gaps (34% versus 12.4%). This is a consequence of the CF model taking considerably longer to find a solution for the LP relaxation in instances with larger Δ values (e.g., 1194.76 seconds for $\Delta^d = 0$ and $\Delta^t = 60$, and 2149.71 seconds for $\Delta^d = \Delta^t = 60$), hindering the CF formulation efficiency. Notably, instances with 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, as we are unable to apply the same constraint reduction strategy used for constraints related to demand deviation.

We also evaluate the behavior of the models regarding the uncertainty level. For this purpose, we created Table 7, which has the same structure as Table 6 but instead of showing the results for all combinations of Δ^d and Δ^t, we present the results for the combinations of deviation in demand (Dev^d) and travel time (Dev^t).

Table 7 – LP Relaxation and Integer Solution of the MTZ and CF formulations based on the single knapsack uncertainty set based on the deviation

		LP Relaxation						**MIP Model**							
		MTZ			**CF**			**MTZ**				**CF**			
Dev^d	Dev^t	**Obj**	**QLR**	**T (s)**	**Obj**	**QLR**	**T (s)**	**Obj**	**Gap**	**T (s)**	**Opt**	**Obj**	**Gap**	**T (s)**	**Opt**
0%	**0%**	165.78	49.1%	0.069	272.68	80.8%	0.048	337.59	3.8%	592.70	26	337.59	0.9%	259.87	28
10%	**0%**	165.78	47.8%	0.143	273.19	78.8%	21.904	346.60	9.5%	1373.52	61	356.27	12.7%	2223.61	40
25%	**0%**	165.78	46.4%	0.145	273.63	76.5%	28.514	357.54	11.2%	1562.26	57	366.89	13.3%	2229.67	40
50%	**0%**	165.82	46.4%	0.270	274.05	76.7%	18.557	357.49	11.3%	1548.53	56	363.88	11.5%	2064.41	45
0%	**10%**	165.99	48.5%	4.056	276.81	80.9%	883.191	342.05	13.1%	1807.72	49	942.71	51.4%	2881.59	21
0%	**25%**	166.05	48.1%	4.636	282.06	81.7%	772.431	345.04	12.6%	1821.62	50	911.20	47.8%	2825.56	20
0%	**50%**	165.92	47.7%	2.727	288.22	82.8%	550.700	347.90	12.9%	1833.86	50	834.38	41.3%	2340.24	20
10%	**10%**	165.99	47.0%	3.942	286.46	81.1%	1040.757	353.22	15.7%	1995.54	45	970.43	53.2%	3115.67	18
25%	**25%**	166.05	45.3%	4.388	293.74	80.2%	1181.163	366.38	17.5%	2051.54	45	1041.54	57.5%	3101.78	15
50%	**50%**	166.82	45.1%	2.540	292.91	79.2%	1045.024	369.66	16.9%	2015.52	45	998.02	50.9%	2561.07	16
All		166.00	47.1%	2.292	281.38	79.9%	554.229	352.35	12.4%	1660.28	484	3422.91	34.0%	2360.35	263

We note that, in this table, similar conclusions to the ones found in Table 6 can be inferred regarding the quality of the linear relaxation, average gaps and number of optimal solutions. While the experiments with the CF model presented stronger QLR for all combinations of deviation, solving the instances using the MTZ formulation provided better results in general, with more instances solved to optimality and lower gaps obtained in any pair of Dev^d and Dev^t. We also noticed that, similarly to the previous uncertainty set, there is not a standard behavior on computational times when the deviation changes. For example, in the experiments with the MTZ formulation, the average running time in instances with deviation exclusively on time ($Dev^d = 0$ and $Dev^t > 0$) increased with bigger Dev^t, but tests in instances with deviation exclusively on demand ($Dev^d > 0$ and $Dev^t = 0$) and those with deviation on both demand and travel time ($Dev^d > 0$ and $Dev^t > 0$) resulted in instances with maximum deviation of 25% actually taking slightly longer to be solved than the ones with a maximum deviation of 50%. This implies that, on the average, high deviations in demand reduces the solution space of the problem, when compared with smaller deviations, more than deviation of time, which helps the solver to prove an optimal solution.

Finally, we study the behavior of both models for the individual instance classes from (Solomon, 1987). To assist in this analysis we created Tables 8 and 9, which present the average results of instances for the MTZ and CF models, respectively. These tables follow a similar structure to Tables 4 and 5. First, we note that using the MTZ model was better than the CF formulation in every instance class, because more instances were solved to optimality and we observe lower gaps and times. Other than that, we note that the performances for each instance class follow the same patterns as the cardinality constrained uncertainty, with the models performing particularly better with instances from classes C1 and C2 as opposed to classes RC1 and RC2. Notably, the MTZ model had a performance with instances from class R1 and R2 similar to those from class C1 and C2.

3.6.3 Computational performance of the B&C methods

3.6.3.1 Cardinality constrained uncertainty set

We now analyze the results from the B&C algorithm developed for both models in Section 3.5. Table 10 summarizes the results obtained for both models, and follows a similar structure to the Table 2, showing, for each combination of Γ^d and Γ^t, the average objective solution value (*Obj*), average optimality gap (*Gap*), average running time in seconds (*T*), total number of instances (*Ins*), number of instances solved to optimality (*Opt*) and number of infeasible instances (*Inf*). This table does not show results for the LP relaxation, as they are equivalent to the ones presented in Table 2.

As expected, the B&C algorithm outperformed solving the compact model with the general-purpose MIP solver, since we start with the deterministic problem and add cut as they are needed, thus having an initial (and usually a final) problem with considerably less constraints.

Table 8 – Average results for the robust models with different values of budgets for the single knapsack uncertainty in instances from classes C1, R1 and RC1 with 25 customers.

		MTZ														
		C1					**R1**					**RC1**				
Δ^q	Δ^t	**Obj**	**Gap**	**Time**	**Ins**	**Opt**	**Obj**	**Gap**	**Time**	**Ins**	**Opt**	**Obj**	**Gap**	**Time**	**Ins**	**Opt**
0	0	190.59	0.0%	422.10	9	8	463.37	0.0%	112.61	12	12	350.24	8.9%	1684.16	8	5
20	0	226.10	2.1%	1216.99	27	19	463.38	1.4%	1057.21	36	30	382.01	24.4%	2223.21	24	10
40	0	232.13	5.1%	1564.00	27	16	464.25	3.4%	1453.07	36	24	432.23	34.5%	2917.12	24	5
60	0	244.85	7.3%	1797.94	27	14	465.01	4.2%	1684.97	36	24	437.15	37.4%	2930.06	24	5
0	20	190.69	28.1%	1163.16	27	19	471.23	4.9%	1749.50	36	23	352.49	23.2%	1974.09	24	12
0	40	191.35	66.7%	1315.33	27	18	486.54	10.4%	2367.80	36	15	359.29	30.0%	2389.90	24	12
0	60	193.81	62.3%	1519.03	27	18	500.02	15.3%	2657.09	36	12	366.65	35.6%	3109.74	24	4
20	20	226.76	21.7%	1456.19	27	18	472.13	5.9%	2004.27	36	19	386.52	29.7%	2706.76	24	9
40	40	236.94	63.0%	2017.29	27	12	487.22	11.4%	2479.03	36	14	455.22	44.9%	3156.42	24	3
60	60	249.25	60.9%	2097.39	27	12	503.57	16.7%	2818.31	36	10	465.47	48.5%	3162.98	24	3
All		218.25	31.7%	1456.94	252	154	477.67	7.4%	1838.39	336	183	398.73	31.7%	2625.44	224	68
		CF														
		C1					**R1**					**RC1**				
Δ^q	Δ^t	**Obj**	**Gap**	**Time**	**Ins**	**Opt**	**Obj**	**Gap**	**Time**	**Ins**	**Opt**	**Obj**	**Gap**	**Time**	**Ins**	**Opt**
0	0	190.59	0.0%	0.68	9	9	463.37	0.0%	25.68	12	12	350.24	0.0%	3.26	8	8
20	0	227.35	4.2%	1547.64	27	20	464.43	2.1%	2009.01	36	20	387.06	10.0%	2750.67	24	8
40	0	239.90	13.4%	2248.40	27	12	466.14	5.1%	2247.51	36	15	453.26	26.7%	3216.56	24	3
60	0	252.86	18.4%	2658.32	27	8	469.71	7.3%	2398.20	36	15	466.42	29.1%	3318.29	24	3
0	20	330.11	15.4%	2012.00	27	12	700.95	28.1%	2816.77	36	6	443.46	17.9%	3102.48	24	3
0	40	713.02	47.2%	2089.65	27	12	1131.21	66.7%	3096.48	36	5	1700.15	76.4%	3451.00	24	0
0	60	699.33	49.3%	2334.35	27	11	1062.45	62.3%	3029.12	36	3	1643.39	75.5%	3413.73	24	0
20	20	375.74	25.7%	2609.11	27	11	576.24	21.7%	2821.96	36	6	905.55	45.9%	3142.31	24	3
40	40	866.80	62.9%	3536.63	27	2	1131.74	63.0%	3047.59	36	5	1707.13	77.3%	3450.89	24	0
60	60	1018.93	79.0%	3722.74	27	0	1062.45	60.9%	3013.70	36	3	1884.40	88.1%	3562.96	24	0
All		491.46	31.6%	2275.95	252	97	752.87	31.7%	2450.60	336	90	994.11	44.7%	2941.22	224	28

Table 9 – Average results for the robust MTZ model with different values of budgets for the single knapsack uncertainty in instances from classes C2, R2 and RC2 with 25 customers.

									MTZ							
				C2					**R2**					**RC2**		
$\mathbf{\Delta^q}$	$\mathbf{\Delta^t}$	**Obj**	**Gap**	**Time**	**Ins**	**Opt**	**Obj**	**Gap**	**Time**	**Ins**	**Opt**	**Obj**	**Gap**	**Time**	**Ins**	**Opt**
0	0	214.45	0.0%	27.20	8	8	382.15	0.0%	34.39	11	11	319.28	15.8%	1721.63	8	5
20	0	214.45	7.1%	272.91	24	24	382.21	0.4%	979.44	33	29	319.60	19.3%	2048.29	24	12
40	0	214.51	12.0%	547.55	24	21	382.52	1.7%	1434.49	33	24	319.79	22.9%	2250.51	24	10
60	0	214.45	14.2%	702.61	24	21	382.51	2.7%	1664.01	33	21	320.02	25.0%	2330.29	24	9
0	20	214.72	24.1%	692.79	24	20	384.12	3.4%	1797.76	33	21	320.40	25.3%	2306.87	24	9
0	40	216.55	73.1%	1047.08	24	18	394.98	9.6%	2753.80	33	12	328.01	29.8%	2739.70	24	6
0	60	219.39	64.1%	1581.82	24	17	403.30	13.6%	3039.79	33	6	329.83	35.3%	2796.09	15	6
20	20	214.72	40.7%	672.55	24	21	387.18	5.0%	2086.67	33	20	321.12	26.4%	2482.18	24	8
40	40	217.23	71.0%	1187.89	24	18	396.36	10.7%	2913.00	33	9	327.77	32.9%	2789.72	24	6
60	60	217.90	74.5%	1636.94	24	16	436.02	15.7%	3021.30	30	6	332.31	36.9%	3045.28	23	5
All		215.84	38.1%	836.93	224	184	393.14	6.3%	1972.46	305	159	323.81	27.0%	2451.06	214	76
									CF							
				C2					**R2**					**RC2**		
$\mathbf{\Delta^q}$	$\mathbf{\Delta^t}$	**Obj**	**Gap**	**Time**	**Ins**	**Opt**	**Obj**	**Gap**	**Time**	**Ins**	**Opt**	**Obj**	**Gap**	**Time**	**Ins**	**Opt**
0	0	214.45	0.0%	1.67	8	8	382.15	0.0%	72.66	11	11	319.28	4.4%	1673.89	8	5
20	0	214.54	0.6%	794.73	24	22	387.28	7.1%	2781.50	33	12	323.57	15.6%	2605.81	24	9
40	0	221.38	6.2%	1383.65	24	16	392.23	12.0%	2963.86	33	6	331.08	27.3%	2784.29	24	6
60	0	220.05	6.4%	1807.14	24	14	397.03	14.2%	3104.11	33	6	335.05	34.8%	3153.44	24	3
0	20	320.05	17.8%	2337.77	24	11	511.44	24.1%	3278.74	33	3	600.02	43.7%	3160.29	24	3
0	40	1140.71	78.4%	3188.08	24	3	1173.57	73.1%	3351.50	33	3	1693.88	80.1%	3271.28	24	3
0	60	1105.90	75.9%	3126.86	24	3	986.11	64.1%	3668.83	33	2	1635.57	79.7%	3683.90	15	0
20	20	235.74	14.2%	2655.30	24	8	766.99	40.7%	3289.73	33	3	683.85	50.1%	3173.95	24	3
40	40	1052.52	71.4%	3178.28	24	4	1173.57	71.0%	3452.08	33	3	1693.88	80.5%	3404.89	24	3
60	60	1105.90	75.1%	3133.13	24	3	1090.23	74.5%	3724.72	30	0	1884.40	93.5%	3771.25	23	0
All		583.12	34.6%	2160.66	224	92	726.06	38.1%	2968.77	305	49	950.06	51.0%	3068.30	214	35

Table 10 – Average objective values and computing times of the solutions obtained for the cardinality constrained uncertainty set using the B&C algorithm.

				MTZ						**CF**			
$\mathbf{\Gamma^q}$	$\mathbf{\Gamma^t}$	**Sol**	**Gap**	**T (s)**	**Ins**	***Opt***	***Inf***	**Sol**	**Gap**	**T (s)**	**Ins**	**Opt**	**Inf**
0	**0**	331.27	0.6%	224.01	56	53	0	331.27	0.7%	279.52	56	52	0
1	**0**	334.63	0.9%	268.11	168	158	0	334.63	0.8%	299.43	168	156	0
5	**0**	350.68	0.8%	304.31	168	157	0	350.73	0.9%	426.13	168	153	0
10	**0**	354.92	1.2%	509.17	168	147	0	354.93	1.4%	651.36	168	144	0
0	**1**	330.74	0.7%	235.06	168	154	5	330.73	0.7%	307.42	168	151	5
0	**5**	337.08	1.1%	334.21	168	150	5	337.14	1.4%	563.33	168	143	5
0	**10**	338.64	1.3%	392.45	168	146	5	338.88	1.9%	700.17	168	137	5
1	**1**	334.36	1.0%	341.11	168	152	5	334.35	0.9%	384.56	168	149	5
5	**5**	353.00	1.2%	495.85	168	146	5	353.53	1.6%	895.58	168	132	5
10	**10**	356.44	1.5%	585.00	168	139	5	356.72	2.0%	1085.92	168	123	5
All		342.18	1.0%	368.93	1568	1402	30	342.29	1.2%	559.34	1568	1340	30

The average solution times and gaps decreased, while the number of optimal solutions increased in any combination of Γ^d and Γ^t. For example, the average solution time of the MTZ model decreased from 836.55 seconds to 368.93 seconds, the average gap reduced from 5.8% to 1.0% and the number of instances solved to optimality increased from 1235 to 1402. Similarly, the average time of the B&C algorithm for the CF model was almost halved, decreasing from 1055.29 to 559.34 seconds, the average gap decreased from 3.3% to 1.2% and 198 more instances were solved to optimality, in comparison to solving the compact formulation.

The MTZ formulation maintained the lead also within the proposed B&C, still taking less time than the B&C based on the CF model, on average, to solve the instances (368.93 seconds against 559.34) and finding more optimal solutions (1402 against 1340). Additionally, the B&C algorithm based on the MTZ model was superior even in the cases where its compact formulation had some advantage, namely the results from instances with Γ less than or equal to 1. Within the B&C, the optimality gaps for the MTZ model were lower than the ones for the CF formulation (0.9% against 1.2%), solving more instances in all pairs Γ^d and Γ^t.

There are two main reasons for this change in behavior: the ability of the CVRPSep package to strengthen the models' linear relaxation and the MTZ formulation being quicker to solve the LP relaxation. In more detailed terms, the first reason revolves around the strengthening of the linear relaxation brought by the CVRPSep, which is particularly more helpful to the MTZ formulation which usually has a weak LR. Consequently, this reduces one of the advantages the CF model has over the MTZ formulation. The second reason is related to the time it takes to solve each the LP relaxation on each node from the branch-and-bound. Since the solver usually obtains the optimal solution for the LP relaxation faster with the MTZ model, it is able to solve Branch-and-bound nodes quicker, and hence find an optimal solution before the CF model. In the case where the models do not find an optimal solution within the time limit, the MTZ formulation usually outperforms the CF model because, seeing as both models now have a stronger linear relaxation, they start with a similar solution in the root node and, since the MTZ formulation is quicker to compute the LP solution, it tends to get deeper in the decision tree and find a better solution by the end of the experiment.

As in the previous section, we also created tables to summarize the results for individual instance classes. Table 11 presents the results of the MTZ and CF model for the instance classes C1, R1 and RC1 and Table 12 shows, in the same order, the results for instances from classes C2, R2 and RC2. These tables follow the same structure of Tables 4-5. We observe that even with the improvements from the B&C algorithm, the behavior of instance classes remained unchanged. The instances with clustered customers (classes C1 and C2) still were the easiest to solve, and the ones with mixed geographic distribution of customers (classes RC1 and RC2)were the hardest. Both algorithms performed reasonably well in the instances from classes R1 and R2, which algorithm using the MTZ formulation solving 98.75% (608) of the instances while the CF model solved 92.70% (597) of them. Notably, the MTZ formulation had some advantage over the CF for this set of instances, even solving all instances from class R2 to optimality. These results can be visualised in figure 5, a column graph similar to Figure 4 that shows the number of instances each robust model was able to solve to optimality for each instance class and uncertainty budget.

As noted previously, the B&C method with the MTZ formulation outperformed the one with the CF model for the most classes, even in classes C1 and RC1 where it previously had a better performance, solving more instances to optimality, with lower times and lower average optimality gaps. Only in C2 the method with the CF model has a slight advantage, in which it solved all the instances to optimality while the one with MTZ solved all but one. Thus, the MTZ model proved to be more suitable for the B&C algorithm in the studied set of instances. Likewise, a decision maker will probably have better results using the B&C algorithm with the MTZ formulation to solve a real-world than with the CF model, when working with the cardinality constrained uncertainty set. Additionally, we can infer, based on the results, that the B&C algorithm is more suited for problems where the customers are either distributed in clusters or completely randomly distributed, and less efficient in instances that combine both behaviors.

3.6.3.2 Knapsack uncertainty set

Similarly to the previous topic, we present the results of the uncertainty set in function of the uncertainty budgets Δ^q and Δ^t. These results are summarized in Table 13, which follows a similar structure to Table 10, with the exception that we are using budget Δ instead of Γ. Additionally, as the solver did not find any infeasible instance, we omit column "Inf" from this table. Most of the inferred conclusions for the cardinality constrained uncertainty set are also true in the knapsack set. Using the B&C algorithm improved the computational performance of the model, indicated by the lower average times and higher number of solved instances than using the compact model. More specifically, the B&C algorithm with MTZ-based formulation solved 824 instances compared to only 484 solved by the general-purpose MIP solver with the compact formulation, and the B&C with the CF formulation solved 391 instances to optimality compared to 263 instances with the compact model. Additionally, the better performance of the B&C algorithm with the MTZ-based model is noticeable regarding computing times, number of solved solutions and optimality gaps. This is a consequence of the strengthening of the linear relaxation brought about by the algorithm, aided by the smaller size of the MTZ model when compared to the CF formulation.

Despite that, it is worth noting that both algorithms had considerably greater difficulty to solve most instances when compared to the deterministic solution. In fact, it is possible to notice that, for the studied uncertainty budgets, the algorithm has considerably more difficulty to solve the robust instances with the knapsack uncertainty set. To help visualize this information, we

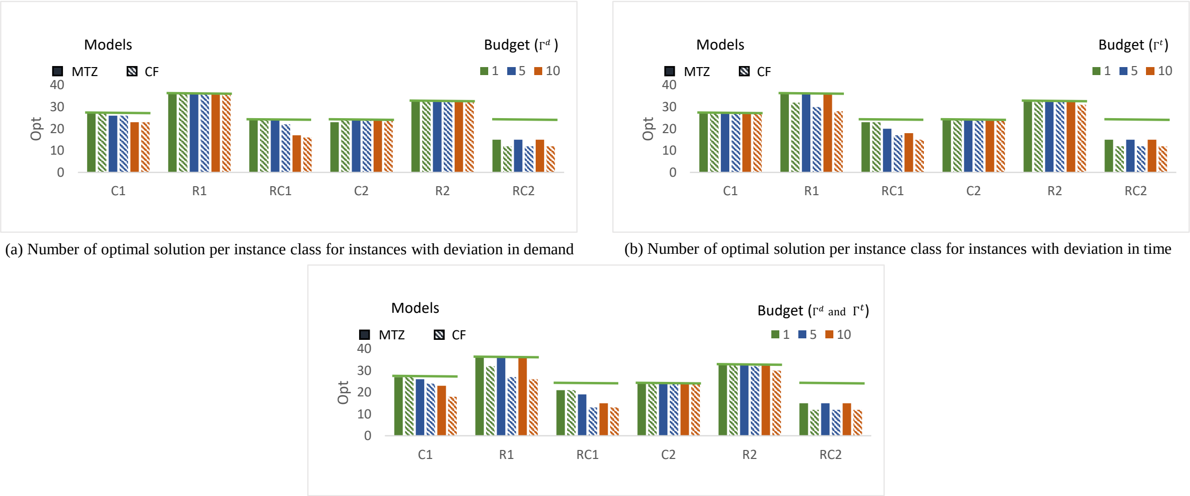
Table 11 – Average results of the tailored B&C algorithm with different values of budgets of uncertainty Γ using instances from classes C1, R1 and RC1 with 25 customers.

		MTZ																	
		C1						**R1**						**RC1**					
Γ^d	Γ^t	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**
0	0	190.59	0.0%	0.15	9	9	0	463.37	0.0%	63.52	12	12	0	350.24	0.0%	3.88	8	8	0
1	0	202.68	0.0%	0.32	27	27	0	463.37	0.0%	252.33	36	36	0	360.14	0.0%	9.26	24	24	0
5	0	238.50	0.0%	167.81	27	26	0	463.37	0.1%	253.71	36	35	0	432.17	0.0%	77.58	24	24	0
10	0	247.54	0.0%	597.95	27	23	0	463.37	0.1%	199.43	36	35	0	451.75	2.5%	1142.61	24	17	0
0	1	192.34	0.0%	0.13	27	27	0	470.33	0.0%	110.68	36	32	4	356.87	0.0%	5.37	24	23	1
0	5	195.83	0.3%	0.13	27	27	0	478.58	0.3%	265.85	36	31	4	383.09	1.7%	484.91	24	20	1
0	10	195.83	0.7%	0.14	27	27	0	478.99	0.4%	341.17	36	29	4	392.77	3.0%	763.52	24	18	1
1	1	202.86	0.0%	0.36	27	27	0	470.33	0.0%	389.09	36	32	4	368.53	0.4%	333.63	24	21	1
5	5	238.93	0.8%	167.94	27	26	0	478.53	0.6%	617.19	36	29	4	444.29	0.9%	857.61	24	19	1
10	10	248.19	1.0%	612.55	27	23	0	479.29	0.8%	518.72	36	29	4	461.78	2.7%	1143.81	24	15	1
All		215.33	0.3%	154.75	252	242	0	470.95	0.2%	301.17	336	300	24	400.16	1.1%	482.22	224	189	6
		CF																	
		C1						**R1**						**RC1**					
Γ^d	Γ^t	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**
0	0	190.59	0.0%	0.18	9	9	0	463.37	0.0%	31.21	12	12	0	350.24	0.0%	4.25	8	8	0
1	0	202.68	0.0%	0.53	27	27	0	463.37	0.0%	61.64	36	36	0	360.14	0.0%	14.38	24	24	0
5	0	238.52	0.1%	234.97	27	26	0	463.37	0.0%	74.08	36	36	0	432.56	0.2%	445.74	24	22	0
10	0	247.71	0.6%	822.89	27	23	0	463.37	0.0%	131.58	36	36	0	451.61	3.0%	1281.19	24	16	0
0	1	192.34	0.0%	0.18	27	27	0	470.33	0.0%	60.90	36	32	4	356.87	0.0%	81.70	24	23	1
0	5	195.83	0.0%	0.45	27	27	0	478.40	0.3%	505.23	36	30	4	383.53	3.5%	1000.97	24	17	1
0	10	195.83	0.0%	0.77	27	27	0	478.97	0.7%	759.51	36	28	4	393.50	5.4%	1241.24	24	15	1
1	1	202.86	0.0%	1.01	27	27	0	470.33	0.0%	123.91	36	32	4	368.59	0.4%	426.73	24	21	1
5	5	238.97	0.4%	471.38	27	24	0	478.50	0.8%	923.56	36	27	4	447.88	2.9%	1882.38	24	13	1
10	10	248.44	1.3%	1299.94	27	18	0	479.01	1.0%	1071.35	36	26	4	463.24	4.0%	1686.14	24	13	1
All		215.38	0.2%	283.23	252	235	0	470.90	0.3%	374.30	336	295	24	400.81	1.9%	806.47	224	172	6

Table 12 – Average results of the tailored B&C algorithm with different values of budgets of uncertainty Γ using instances from classes C2, R2 and RC2 with 25 customers.

		MTZ																	
		C2						**R2**						**RC2**					
Γ^d	Γ^t	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**
0	0	214.45	0.0%	0.95	8	8	0	382.15	0.0%	18.70	11	11	0	319.28	4.5%	1442.09	8	5	0
1	0	214.45	0.0%	1.26	24	23	0	382.15	0.0%	30.21	33	33	0	319.28	6.0%	1445.87	24	15	0
5	0	214.45	0.0%	1.37	24	24	0	382.15	0.0%	34.50	33	33	0	319.30	5.8%	1434.40	24	15	0
10	0	214.45	0.0%	1.47	24	24	0	382.15	0.0%	28.89	33	33	0	319.28	5.6%	1408.53	24	15	0
0	1	214.51	0.0%	0.92	24	24	0	383.86	0.0%	18.79	33	33	0	319.69	4.6%	1447.13	24	15	0
0	5	214.57	0.0%	1.41	24	24	0	384.80	0.0%	26.13	33	33	0	319.83	5.1%	1418.28	24	15	0
0	10	214.57	0.3%	1.38	24	24	0	384.89	0.0%	27.18	33	33	0	320.02	5.0%	1432.98	24	15	0
1	1	214.51	0.0%	1.62	24	24	0	383.86	0.0%	34.31	33	33	0	319.78	6.2%	1421.30	24	15	0
5	5	214.57	0.0%	1.58	24	24	0	384.80	0.0%	43.32	33	33	0	320.53	6.2%	1437.48	24	15	0
10	10	214.57	0.3%	1.63	24	24	0	384.89	0.0%	43.48	33	33	0	320.73	6.2%	1422.58	24	15	0
All		214.51	0.1%	1.36	224	223	0	383.57	0.0%	30.55	308	308	0	319.77	5.5%	1431.06	224	140	0

		CF																	
		C2						**R2**						**RC2**					
Γ^d	Γ^t	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**	**Obj**	**Gap**	**T(s)**	**Ins**	**Opt**	**Inf**
0	0	214.45	0.0%	1.67	8	8	0	382.15	0.0%	66.45	11	11	0	319.28	4.8%	1812.35	8	4	0
1	0	214.45	0.0%	1.54	24	24	0	382.15	0.0%	124.88	33	33	0	319.28	5.3%	1815.35	24	12	0
5	0	214.45	0.0%	2.72	24	24	0	382.15	0.0%	243.44	33	33	0	319.28	5.8%	1824.23	24	12	0
10	0	214.45	0.0%	3.92	24	24	0	382.15	0.0%	233.00	33	33	0	319.28	5.8%	1830.92	24	12	0
0	1	214.51	0.0%	1.53	24	24	0	383.86	0.0%	117.07	33	33	0	319.60	5.0%	1816.18	24	12	0
0	5	214.57	0.0%	2.84	24	24	0	384.80	0.0%	263.89	33	33	0	320.00	5.4%	1818.32	24	12	0
0	10	214.57	0.0%	2.94	24	24	0	385.08	0.3%	502.83	33	31	0	320.60	5.8%	1825.53	24	12	0
1	1	214.51	0.0%	2.76	24	24	0	383.86	0.0%	194.05	33	33	0	319.66	5.7%	1808.57	24	12	0
5	5	214.57	0.0%	3.81	24	24	0	384.80	0.0%	464.76	33	32	0	320.61	6.7%	1828.22	24	12	0
10	10	214.57	0.0%	7.88	24	24	0	385.06	0.3%	712.52	33	30	0	321.15	6.6%	1858.27	24	12	0
All		214.51	0.0%	3.16	224	224	0	383.60	0.1%	292.29	308	302	0	319.87	5.7%	1823.79	224	112	0



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

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

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

Figure 5 – Number of optimal solutions obtained by the robust models per instance class and uncertainty budget using the B&C algorithm.

Table 13 – Average objective values and computational times of the solutions obtained for the knapsack uncertainty set using the B&C algorithm.

		MTZ					CF				
Δ^q	**Δ^t**	**Sol**	**Gap**	**T (s)**	**Ins**	**Opt**	**Sol**	**Gap**	**T (s)**	**Ins**	**Opt**
0	**0**	331.27	0.6%	224.01	56	49	331.27	0.6%	224.01	56	53
20	**0**	341.58	7.7%	1263.72	168	124	344.30	6.3%	2104.34	168	91
40	**0**	350.00	11.1%	1660.96	168	100	359.16	14.2%	2480.07	168	58
60	**0**	352.93	12.7%	1828.73	168	94	365.21	17.4%	2733.56	168	49
0	**20**	333.88	9.7%	1625.50	168	104	496.49	24.6%	2799.64	168	38
0	**40**	341.72	14.5%	2142.09	168	81	1233.65	69.8%	3073.46	168	26
0	**60**	348.46	18.6%	2478.64	167	63	1127.36	64.8%	3045.10	105	19
20	**20**	345.43	12.2%	1910.75	168	95	595.95	32.6%	2951.88	168	34
40	**40**	363.23	19.0%	2446.77	168	62	1247.16	70.3%	3332.96	168	17
60	**60**	378.05	23.3%	2645.94	164	52	1303.35	76.9%	3386.76	108	6
Total		318.84	12.9%	1822.71	1563	824	710.58	37.7%	2613.18	1445	391

present Figure 6, which shows the number of instances solved to optimality for each combination of deviations ($[Dev^q, Dev^t]$), uncertainty sets ($Card$ or $Knap$) and models (MTZ and CF). The green bar over the columns show the total number of instances for that combination of Dev^q and Dev^t.

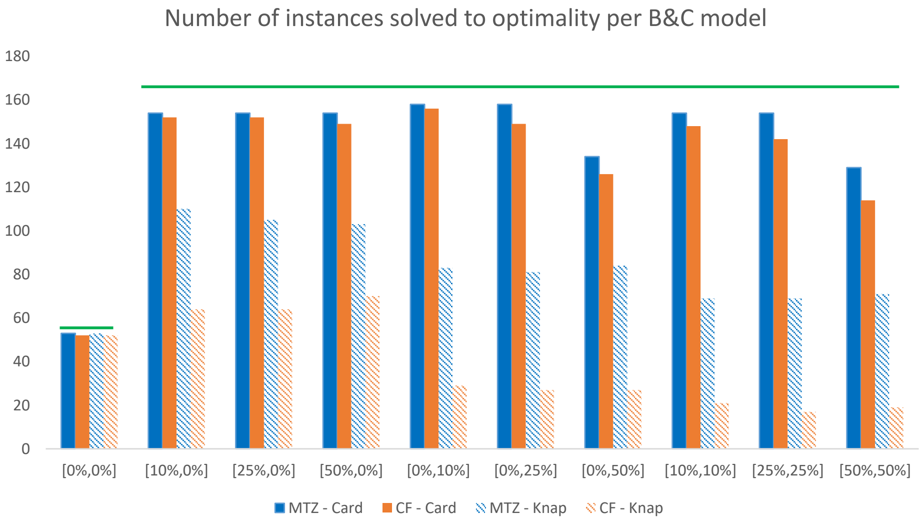


Figure 6 – Number of instances solved to optimality by the B&C models for each combination of $[Dev^q, Dev^t]$.

It is easy to see that the B&C methods with the cardinality constrained uncertainty set were able to find optimal solutions for considerably more instances than the methods with the knapsack uncertainty set. Since the proposed models index the uncertainty budget on the decision variables, by choosing larger budgets, the problem grows larger in number of variables and constraints, becoming more difficult to solve. Thus, as the chosen budgets for the knapsack uncertainty set ($\Delta = \{20, 40, 60\}$) are considerably larger that those used for the cardinality constrained set ($\Gamma = \{1, 5, 10\}$), instances using this set were considerably harder to solve. Additionally, it is possible to see that the algorithm with the knapsack uncertainty set had considerably more difficulty to solve instances with deviation on travel time, which is a consequence of the impossibility of reducing the dynamic programming equations for time, as done in demand.

3.6.4 Robustness analysis

In this subsection, we analyze the impact of the proposed RO approach regarding the objective function value and robustness of the solutions. The robustness of a solution is an important information in the decision-making process, as it can be used to evaluate the trade-off between cost and its “safety”, which in this work means the chance of a solution becoming infeasible in practice. For example, the decision-maker might prefer a more expensive solution that never fails over a cheaper solution that has a higher probability of becoming infeasible. To estimate the robustness of solutions, we applied a Monte Carlo simulation. Essentially, every solution was tested against 1000 randomly generated scenarios. In each scenario, the value of uncertain parameters was generated between its nominal value and maximum deviation, following a continuous uniform distribution. If a solution considers only uncertainty on demand (i.e., $\Gamma^d > 0$ and $\Gamma^t = 0$, or $\Delta^d > 0$ and $\Delta^t = 0$), the variability was applied to this parameter only. This is similar for uncertainty on travel times only. The solutions for deterministic instances, in particular, were tested for every combination of Dev^d and Dev^t in order to verify the risks when considering no uncertainty when compared to the robust solutions. We can interpret the percentage of simulated scenarios that were infeasible as the risk the solution becomes infeasible when the uncertain parameters become known.

3.6.4.1 Cardinality constrained uncertainty set

To assist in this analysis, Tables 14 to 16 show the results for all instance classes considering a maximum deviation of 10%, 25% and 50%, respectively. They summarize these results for each individual instance class and show, for each combination of budgets (Γ^d and Γ^t) and deviations (Dev), the price-of-robustness (PoR), which constitutes the percentage increase in costs the robust solution has over the deterministic one, and the percentage of infeasible scenarios in the Monte Carlo simulation (Risk). Furthermore, we also present the lowest and the highest PoR among the solutions, represented by *Best* and *Worst* in the tables respectively, and the normalized standard deviation (Std.Dev.) of all solutions for each combination. We show these final parameters to evaluate the variability of solutions 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 noted that, by construction, greater budgets (Γ^d and Γ^t) enlarge the uncertainty sets, and therefore, the probability of being infeasible decreases, but its cost increases accordingly. This can be identified by comparing the PoR and the risk between solutions with same level of deviation and different budgets. A clear example of this behaviour can be seen in the results of instances from class RC1 with $\Gamma^d > 0\%$ and $\Gamma^t = 0\%$ with maximum deviation of $Dev = 50\%$, in which the costs increase whenever a larger budget is chosen. For example, the solution considering $\Gamma^d = 1$ is 9.4% more expensive, on average, than the deterministic one. This growth is even higher with $\Gamma^d = 5$ and $\Gamma^d = 10$, in which the average costs increased by 33.4% and 39.0% over the deterministic solution, respectively.

The probability of constraint violation, on the other hand, tends to decrease the bigger the chosen uncertainty budget Γ is. Returning to the example of class RC1 with maximum deviation of $Dev = 50\%$ exclusively in the demand vector, we observe that for a higher budget Γ,

Table 14 – Average, best and worst Price of robustness (PoR), probability of constraint violation (Risk) and Standard Deviation of the solutions for instances with maximum deviation of 10%,considering the cradinality-constrained uncertainty set.

		Dev 10%														
		C1					**R1**					**RC1**				
Γ^q	Γ^t	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**
0	0	47.7%	0.0%	0.0%	0.0%	0.7%	0.0%	0.0%	0.0%	0.0%	14.0%	30.3%	0.0%	0.0%	0.0%	15.7%
1	0	41.1%	0.2%	0.0%	1.8%	0.2%	0.0%	0.0%	0.0%	0.0%	14.0%	30.0%	0.0%	0.0%	0.0%	15.7%
5	0	0.0%	18.6%	15.8%	23.2%	2.6%	0.0%	0.0%	0.0%	0.0%	14.0%	0.0%	9.4%	0.0%	20.7%	10.4%
10	0	0.0%	22.8%	20.0%	24.9%	1.6%	0.0%	0.0%	0.0%	0.0%	14.0%	0.0%	21.5%	0.0%	33.7%	5.8%
0	0	0.0%	0.0%	0.0%	0.0%	0.7%	23.1%	0.0%	0.0%	0.0%	14.0%	29.2%	0.0%	0.0%	0.0%	15.7%
0	1	0.0%	0.0%	0.0%	0.0%	0.7%	6.2%	0.8%	0.0%	6.8%	14.0%	17.7%	0.1%	0.0%	1.2%	16.0%
0	5	0.0%	0.0%	0.0%	0.0%	0.7%	0.0%	1.3%	0.0%	7.1%	13.7%	0.0%	0.6%	0.0%	1.3%	15.8%
0	10	0.0%	0.0%	0.0%	0.0%	0.7%	0.0%	1.5%	0.0%	7.1%	13.5%	0.0%	0.6%	0.0%	1.3%	15.8%
0	0	47.7%	0.0%	0.0%	0.0%	0.7%	23.1%	0.0%	0.0%	0.0%	14.0%	52.5%	0.0%	0.0%	0.0%	15.7%
1	1	42.4%	0.2%	0.0%	1.8%	0.2%	6.2%	0.8%	0.0%	6.8%	14.0%	40.6%	0.1%	0.0%	1.2%	16.0%
5	5	0.0%	18.7%	15.8%	23.2%	2.6%	0.0%	1.3%	0.0%	7.1%	13.7%	0.0%	11.1%	0.9%	21.0%	9.7%
10	10	0.0%	22.8%	20.0%	24.9%	1.6%	0.0%	1.5%	0.0%	7.1%	13.5%	0.0%	18.2%	1.2%	34.0%	6.0%
All		14.9%	6.9%	6.0%	8.3%	1.1%	4.9%	0.6%	0.0%	3.5%	13.8%	16.7%	5.1%	0.2%	9.5%	13.2%
		C2					**R2**					**RC2**				
Γ^q	Γ^t	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**
0	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
1	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
5	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
10	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
0	0	0.0%	0.0%	0.0%	0.0%	0.2%	9.4%	0.0%	0.0%	0.0%	9.1%	1.3%	0.0%	0.0%	0.0%	8.4%
0	1	0.0%	0.0%	0.0%	0.0%	0.2%	9.4%	0.0%	0.0%	0.0%	9.1%	1.5%	0.0%	0.0%	0.0%	8.4%
0	5	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.4%	0.0%	2.4%	8.7%	0.0%	0.0%	0.0%	1.2%	8.2%
0	10	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.4%	0.0%	2.4%	8.7%	0.0%	0.1%	0.0%	1.4%	8.1%
0	0	0.0%	0.0%	0.0%	0.0%	0.2%	9.4%	0.0%	0.0%	0.0%	9.1%	1.3%	0.0%	0.0%	0.0%	8.4%
1	1	0.0%	0.0%	0.0%	0.0%	0.2%	9.4%	0.0%	0.0%	0.0%	9.1%	1.4%	0.0%	0.0%	2.0%	8.0%
5	5	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.4%	0.0%	2.4%	8.7%	0.0%	0.0%	0.0%	0.3%	8.4%
10	10	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.4%	0.0%	2.4%	8.7%	0.0%	0.2%	0.0%	1.4%	8.1%
All		0.0%	0.0%	0.0%	0.0%	0.2%	3.1%	0.1%	0.0%	0.8%	9.0%	0.5%	0.0%	0.0%	0.5%	8.3%

the risk decreases from failing 98.8% of times in the deterministic case to always being feasible when $\Gamma^d = 10$. It is up to the decision maker to select the most appropriate level of robustness. For instance, he or she might opt for the solution of $\Gamma^d = 5$, resulting in an average risk of only 0.1%, a value considerably lower than the 98.8% of the deterministic solution and the 96.9% from the solution with $\Gamma^d = 1$, but costing 33.4% more than the solution that considers no uncertainty. They may also choose to increase the PoR yet 5.6% more to completely nullify the risks by taking the solution with budget of uncertainty $\Gamma^d = 10$. This behavior consistently happens for any deviation combination.

Interestingly, we note that the standard deviation of the robust solutions are close to, or even smaller than, the deterministic solutions' in the majority of the instances. The only outliers of this trend are the instances from class C1 with worst-case deviation of 50% in time, whose solutions' relative standard deviation are more than 10 times greater than the deterministic's. This behavior happens because the deterministic solutions have relatively close optimal solutions with different levels of tightness on time constraints, thus when we introduce deviation in this parameter these instances' optimal solutions grows in different rates, with some, like instances C107-C109, not changing at all while others having a high increase. Since the standard deviations are relatively small, it is expected that the majority of combinations of budget and instance classes present a relatively small difference between their best and worst PoR, which is true for most instances with the exception of the aforementioned instances of class C1 with high deviation

Table 15 – Average, best and worst Price of robustness (PoR), probability of constraint violation (Risk) and Standard Deviation of the solutions for instances with maximum deviation of 25%, considering the cradinality constrained uncertainty set.

								Dev 25%								
				C1				**R1**					**RC1**			
Γ^q	Γ^t	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**
0	0	99.7%	0.0%	0.0%	0.0%	0.7%	0.0%	0.0%	0.0%	0.0%	14.0%	84.0%	0.0%	0.0%	0.0%	15.7%
1	0	99.9%	0.2%	0.0%	1.8%	0.2%	0.0%	0.0%	0.0%	0.0%	14.0%	83.9%	0.0%	0.0%	0.0%	15.7%
5	0	0.0%	23.2%	20.0%	24.9%	1.6%	0.0%	0.0%	0.0%	0.0%	14.0%	0.0%	31.8%	10.1%	44.5%	6.1%
10	0	0.0%	32.4%	31.4%	33.5%	1.0%	0.0%	0.0%	0.0%	0.0%	14.0%	0.0%	32.0%	11.2%	44.5%	6.3%
0	0	0.0%	0.0%	0.0%	0.0%	0.7%	62.7%	0.0%	0.0%	0.0%	14.0%	72.6%	0.0%	0.0%	0.0%	15.7%
0	1	0.0%	0.0%	0.0%	0.0%	0.7%	15.7%	3.0%	0.0%	8.5%	15.2%	30.2%	0.7%	0.0%	3.1%	15.8%
0	5	0.0%	0.0%	0.0%	0.0%	0.7%	0.0%	3.9%	0.0%	8.5%	14.9%	0.0%	7.1%	0.1%	14.9%	19.5%
0	10	0.0%	0.0%	0.0%	0.0%	0.7%	0.0%	3.9%	0.0%	8.5%	14.9%	0.0%	8.6%	1.1%	23.5%	19.9%
0	0	99.7%	0.0%	0.0%	0.0%	0.7%	62.2%	0.0%	0.0%	0.0%	14.0%	98.8%	0.0%	0.0%	0.0%	15.7%
1	1	99.9%	0.2%	0.0%	1.8%	0.2%	14.9%	3.0%	0.0%	8.5%	15.2%	85.5%	0.7%	0.0%	3.1%	15.8%
5	5	0.0%	23.4%	20.0%	24.9%	1.7%	0.0%	3.9%	0.0%	8.5%	14.9%	0.0%	34.8%	11.6%	45.2%	7.0%
10	10	0.0%	32.8%	31.4%	34.0%	0.8%	0.0%	3.9%	0.0%	8.5%	14.7%	0.0%	35.2%	13.1%	45.2%	7.2%
All		33.3%	9.4%	8.6%	10.1%	0.8%	13.0%	1.8%	0.0%	4.3%	14.5%	37.9%	12.6%	3.9%	18.7%	13.4%
				C2				**R2**					**RC2**			
Γ^q	Γ^t	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**
0	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
1	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
5	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.3%	8.4%
10	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
0	0	10.6%	0.0%	0.0%	0.0%	0.2%	31.6%	0.0%	0.0%	0.0%	9.1%	11.3%	0.0%	0.0%	0.0%	8.4%
0	1	10.9%	0.0%	0.0%	0.0%	0.2%	3.4%	0.4%	0.0%	2.4%	8.7%	0.0%	0.1%	0.0%	1.4%	8.1%
0	5	0.0%	0.1%	0.0%	0.7%	0.0%	0.0%	0.5%	0.0%	3.0%	8.7%	0.0%	0.2%	0.0%	1.4%	8.1%
0	10	0.0%	0.1%	0.0%	0.7%	0.0%	0.0%	0.5%	0.0%	3.8%	8.6%	0.0%	0.2%	0.0%	1.4%	8.1%
0	0	10.7%	0.0%	0.0%	0.0%	0.2%	31.4%	0.0%	0.0%	0.0%	9.1%	11.3%	0.0%	0.0%	0.0%	8.4%
1	1	10.6%	0.0%	0.0%	0.0%	0.2%	3.4%	0.4%	0.0%	2.4%	8.7%	5.0%	0.1%	0.0%	0.7%	8.3%
5	5	0.0%	0.1%	0.0%	0.7%	0.0%	0.0%	0.5%	0.0%	3.0%	8.7%	0.0%	0.2%	0.0%	1.4%	8.1%
10	10	0.0%	0.1%	0.0%	0.7%	0.0%	0.0%	0.5%	0.0%	3.8%	8.6%	0.0%	0.3%	0.0%	2.2%	8.0%
All		3.6%	0.0%	0.0%	0.2%	0.2%	5.8%	0.2%	0.0%	1.5%	8.9%	2.3%	0.1%	0.0%	0.7%	8.3%

in time and instances from class RC1. From a decision maker's standpoint, that means that using the RO paradigms in an environment with customers randomly dispersed or localized in clusters usually results in solutions with similar increments in costs over the deterministic solution, with relatively low chances of increasing above the expected. This is particularly good for planning because, once the relevant uncertainty budget's are set, there is little need to test different RO parameters for the new data when working with a certain target for costs. In the clustered environments, there could unexpected surges in the PoR when operating in an environment with high deviation in times, but it is worth noting a deviation of 50% is considerably extreme in most real-world applications, and probably is a consequence of errors in data collection or analysis processes.

Another side effect of choosing higher budgets of uncertainty is the chance of not existing a feasible robust solution for that specific combination of Γ and Dev. This happens because it might be impossible to determine feasible routes in a way that Γ parameters attain their worst-case values simultaneously, while the deterministic formulation, or small enough Γ, might not suffer with this problem. This happened in some instances when the maximum deviation in travel time was $Dev^t = 50\%$, namely R101, R102, R103, R104 and RC105. The models were able to get deterministic solutions for instances, but considering any non-null budget of uncertainty in travel time deemed them infeasible. Particularly for the tested instances, the ones that became infeasible with this particular deviation of time happened because of one or more nodes whose

Table 16 – Average, best and worst Price of robustness (PoR), probability of constraint violation (Risk) and Standard Deviation of the solutions for instances with maximum deviation of 50%,considering the cradinality constrained uncertainty set.

		Dev 50%														
		C1					**R1**					**RC1**				
Γ^q	Γ^t	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**
0	0	100.0%	0.0%	0.0%	0.0%	0.7%	0.0%	0.0%	0.0%	0.0%	14.0%	98.8%	0.0%	0.0%	0.0%	15.7%
1	0	99.8%	18.6%	15.8%	23.2%	2.6%	0.0%	0.0%	0.0%	0.0%	14.0%	98.8%	9.4%	0.0%	20.7%	10.4%
5	0	1.1%	33.6%	31.9%	35.4%	1.2%	0.0%	0.0%	0.0%	0.0%	14.0%	0.1%	33.4%	11.2%	44.9%	6.5%
10	0	0.0%	34.4%	32.4%	35.6%	0.9%	0.0%	0.0%	0.0%	0.0%	14.0%	0.0%	39.0%	12.1%	56.1%	3.9%
0	0	0.0%	0.0%	0.0%	0.0%	0.7%	95.1%	0.0%	0.0%	0.0%	14.0%	96.5%	0.0%	0.0%	0.0%	15.7%
0	1	0.0%	2.7%	0.0%	24.7%	7.7%	57.9%	3.4%	0.0%	10.0%	12.1%	38.3%	6.2%	2.0%	16.6%	20.9%
0	5	0.0%	8.2%	0.0%	30.0%	10.7%	0.0%	6.9%	6.6%	17.6%	11.4%	0.0%	21.5%	6.8%	33.9%	15.6%
0	10	0.0%	8.2%	0.0%	30.0%	10.7%	0.0%	7.1%	6.6%	17.8%	11.3%	0.0%	28.7%	16.6%	53.6%	11.5%
0	0	100.0%	0.0%	0.0%	0.0%	0.7%	95.1%	0.0%	0.0%	0.0%	14.0%	100.0%	0.0%	0.0%	0.0%	15.7%
1	1	99.2%	18.9%	15.8%	24.7%	2.9%	56.3%	3.4%	0.0%	10.0%	12.1%	85.6%	16.8%	14.9%	27.0%	15.1%
5	5	1.0%	33.9%	31.9%	36.5%	1.4%	0.0%	7.0%	6.6%	17.6%	11.5%	0.0%	38.7%	16.6%	54.7%	5.3%
10	10	0.0%	35.1%	32.4%	38.1%	1.5%	0.0%	7.1%	6.6%	17.8%	11.3%	0.0%	40.2%	16.6%	56.2%	5.4%
All		33.4%	16.1%	13.4%	23.2%	3.5%	25.4%	2.9%	2.2%	7.6%	12.8%	43.2%	19.5%	8.1%	30.3%	11.8%
		C2					**R2**					**RC2**				
Γ^q	Γ^t	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std.Dev.**
0	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
1	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
5	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
10	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
0	0	12.5%	0.0%	0.0%	0.0%	0.2%	41.6%	0.0%	0.0%	0.0%	9.1%	12.5%	0.0%	0.0%	0.0%	8.4%
0	1	12.5%	0.1%	0.0%	0.7%	0.0%	9.1%	0.9%	0.0%	3.5%	9.6%	4.4%	0.3%	0.0%	2.2%	8.0%
0	5	0.0%	0.1%	0.0%	0.7%	0.0%	0.0%	1.3%	0.0%	3.8%	9.5%	0.0%	0.3%	0.0%	2.3%	8.0%
0	10	0.0%	0.1%	0.0%	0.7%	0.0%	0.0%	1.3%	0.0%	3.8%	9.5%	0.0%	0.5%	0.0%	3.7%	7.7%
0	0	12.5%	0.0%	0.0%	0.0%	0.2%	41.6%	0.0%	0.0%	0.0%	9.1%	12.5%	0.0%	0.0%	0.0%	8.4%
1	1	12.5%	0.1%	0.0%	0.7%	0.0%	9.1%	0.9%	0.0%	3.5%	9.6%	4.4%	0.2%	0.0%	1.8%	8.1%
5	5	0.0%	0.1%	0.0%	0.7%	0.0%	0.0%	1.3%	0.0%	3.8%	9.5%	0.0%	1.2%	0.0%	9.4%	6.8%
10	10	0.0%	0.1%	0.0%	0.7%	0.0%	0.0%	1.3%	0.0%	3.8%	9.5%	0.0%	1.2%	0.0%	9.3%	6.8%
All		4.2%	0.0%	0.0%	0.3%	0.1%	8.5%	0.6%	0.0%	1.9%	9.3%	2.8%	0.3%	0.0%	2.4%	8.0%

worst-case travel time from the depot already surpassed the closing time of the time window, i.e $a_0 + 1.5\bar{t}_{0j} > b_j$.

We also evaluate the impact of the maximum percentage deviation of the parameters (Dev) on the solution by analyzing the changes of costs and risks when Γ^d and Γ^t are fixed while deviation changes. A first identified of higher deviation levels is, for any non-null combination of budgets, an increase on costs of robust solutions. As an example, we can point the solutions of class C1 using $\Gamma^d = 5$ and $\Gamma^t = 0$, in which the average PoR increases from 18.6% when considering a deviation in time of 10%, and this value increases to 23.2%, when the deviation is 25%, and 33.6%, when $Dev = 50\%$. This behaviour is expected, since the robust model might have to take more conservative routes in order to consider this additional deviation.

Furthermore, an environment with high deviation have solutions with significantly more risks, especially when we consider smaller budgets. For example, the solutions of instances from class R1 with deviation only on time becomes riskier with greater values of Dev. We can highlight solutions with $\Gamma^t = 1$ and $\Gamma^d = 0$, whose average risk is 6.2% when deviation is 10%, 15.7% when it is 25% and 57.9% when the maximum deviation is 50%. This is expected since this additional deviation makes it easier for parameters to surpass the capacity or time windows constraints, and if the realization of these parameters do not fall inside the polyhedral set, the robust solution does not ensure feasibility. Finally, higher deviation also increases the odds of feasible solutions not existing, as the robust model might not be able to find feasible robust solutions for high levels of uncertainty. As previously noted, instances R101, R102, R103, R104 and RC105 were infeasible whenever $\Gamma^t > 0$ and $Dev = 50\%$.

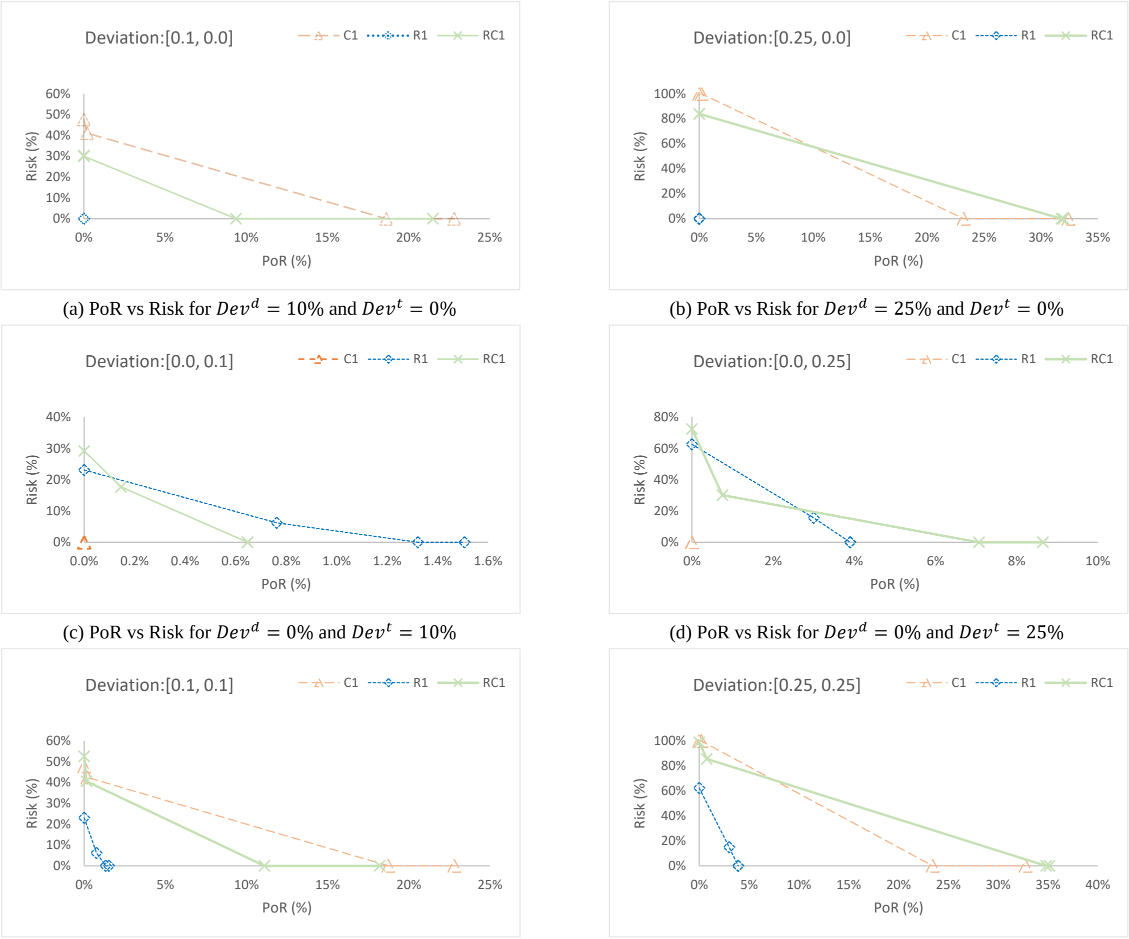
These findings are illustrated by the graphs in Figure 7. They plot the risk versus the PoR for each instance set (triangles for C1, diamonds for R1 and cross marks for RC1) and different budget of uncertainty value. The values of deviation are indicated in the figures as "[Dev^d; Dev^t]". A more convex curve represents a better trade-off between risk and PoR of the solution, because this means that the robust solution effectively reduce the risks with a small increase in costs. Thus, some of solutions with the most expensive trade-off would be solutions for instance class RC1 in Figures 7(c) and 7(f), where an increase of 30% in the costs is needed to null the risks. We can also note that in some cases, namely class C1 in Figures 7(a), 7(b), 7(e) and 7(f), class R1 in 7(c) and class RC1 in 7(d), increasing the budget's size increase the solution costs with no impact in the risks. These cases happened when the solution with the previous budgets had no risk already and using a more conservative Γ only increased the costs of the solution. For 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, we also note that some robust solutions, particularly the ones with deviation in time, greatly improve the robustness of the solution, voiding all risks in the instances with a worst-case deviation of 10% with an increase in costs as little as 1.3% .

Analysing the behavior of each individual instance classes when faced with uncertainty, we noted that the ones with clustered customers (classes C1 and C2) were highly resistant to uncertainty in travel time, particularly class C1 whose deterministic solution have not violated any constraint in any instance (indicated by Risk=0.0%) with deviation exclusively in travel time. Although there were deterministic solutions from class C2 that failed in some simulations, resulting in an average fail rate of 10.6%, when $Dev = 25\%$, and 12.5%, when $Dev = 50\%$, the robust alternative was able to void this risk for a negligible cost ($PoR = 0.1\%$), so we can infer this set of instances is also resistant to time variability.

Similarly, instances with randomly distributed customers (R1 and R2) were completely unaffected by demand uncertainty, i.e., not only the risks of violating a constraint being always zero but also the robust solutions were always the same as the deterministic. This can be identified by PoR being zero in instances with uncertainty exclusively on demand. Since the distance between customers in instances of these classes are relatively large, the routes in solution have few customers, which allowed the vehicles to absorb the demand uncertainty naturally. On the other hand, the instances from class RC1 were affected by both types of uncertainty.

We observed that classes R1 and R2 were not the only ones completely unaffected by demand uncertainty, classes C2 and RC2 also presented this behavior but for a different reason. The cause of this immunity is the vehicles' capacity in this sets of instances being five times larger than the ones in classes C1 and RC1 while the demands remain in a similar level and thus the vehicles are able to absorb all deviation. Note that both class C2 and RC2 are affected by uncertainty on demand, although this was less noticeable in class C2.

To visually exemplify on how the solutions can change when considering the RO approach, Figure 8 presents the optimal routes for the deterministic (a) and robust (b) cases when considering uncertain of instance C101 with 25 customers. This particular robust solution allows up to 5 nodes in a route to assume its worst-case value, i.e., $\Gamma^d = 5$, and the maximum deviation in



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

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

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

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

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

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

Figure 7 – Trade-off between price of robustness (PoR) and probability of constraint violation (Risk) for the instances of class C1, R1 and RC1.

demand is 10% of its nominal value. We also present the total nominal load on each route. The capacity of each vehicle is 200 units.

As expected, the robust solution is more expensive than the deterministic, with the optimal solution costs increasing from 191.3 to 235.6, or 23.2% more. However, the probability of constraint violation in the robust solution is substantially lower. With the mild deviation of only 10%, the deterministic solution failed in 39.5% of cases, while the robust solution was feasible in every simulation. This analysis is also illustrated in Figure 9, where we plot the risk versus the PoR for solutions of instance C101 with different levels of deviation. Each ordered pair highlighted in the plot is associated with a different budget of uncertainty value.

Thus, it is possible to see that the RO approach can provide relevant solutions to support the decision-making process of choosing vehicle routes. The decision-maker can create a set of solutions with different trade-offs for risks and costs that should be considered. With this information, they can make an informed choice and opt for a more robust solution, which deteriorates the costs but ensure a better service level for customers, or assume more risks and taking a cheaper solution.

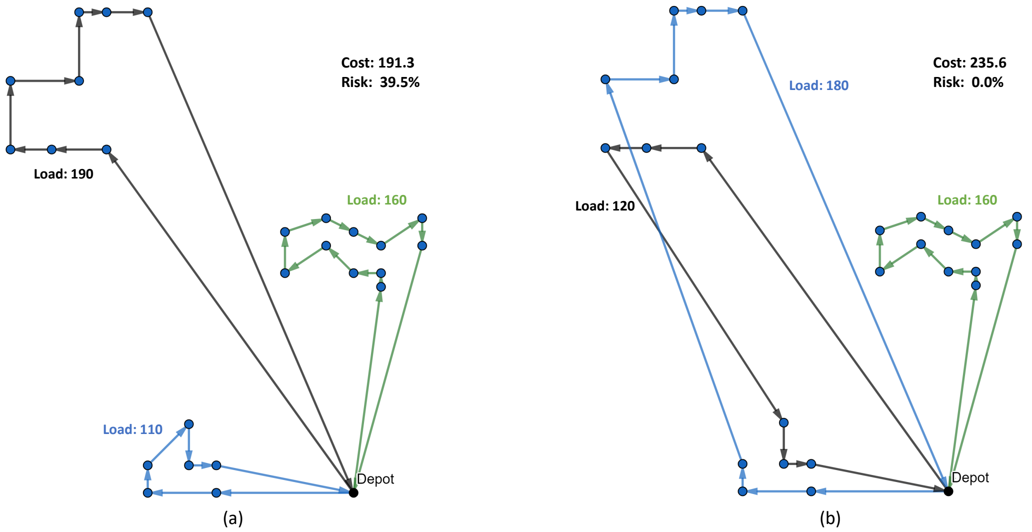


Figure 8 – Deterministic (a) and robust (b) solutions of instance C101 under uncertainty on demand with maximum deviation of 10% over nominal demand

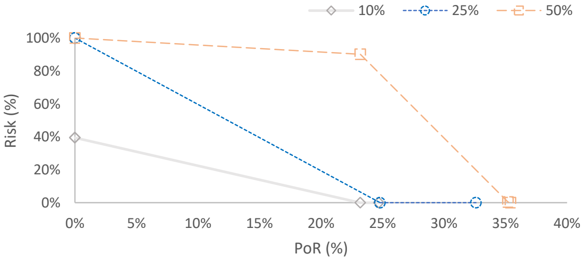


Figure 9 – Trade-off between PoR and Risk for instance C101.

3.6.4.2 Knapsack uncertainty set

Similarly to the results for the cardinality constrained set, Tables 17 to 19 summarize the results for each individual instance class and show, for each combination of budgets (Δ^d and Δ^t) and deviations (Dev), the average, best and worst the price-of-robustness (PoR), the probability of constraint violation (Risk) and the normalized standard deviation of the solution. Table 17 presents the results for instance of all classes considering a maximum deviation of 10%; Table 18 summarize the results for the deviation of 25%; and Table 19 does the same with the results from the instances with maximum deviation of 50%.

Similarly to the previous uncertainty set, as the budgets Δ^d and Δ^t increase, the solutions become more conservative and the instances harder to solve. Thus, for the same level of deviation, the price of robustness and solving times tend to get higher while the risks tend to lower. A good example of this behavior is found in the solutions from instances of class C1 with $Dev = 50\%$ and deviation on both demand and travel time ($\Delta^d = \Delta^t > 0$), in which the price-of-robustness increases up to 20.3% depending of the budget's size, and the risks lower from 100.0%, when $\Delta^d = \Delta^t = 0$ to 13.5%, when $\Delta^d = \Delta^t = 60$, consistently decreasing for each level of robustness, as noted by the risk being 99.8% when $\Delta^d = \Delta^t = 20$ and 60.7% when $\Delta^d = \Delta^t = 40$. An expected consequence of the uncertainty design, which becomes more

Table 17 – Average, best and worst Price of robustness (PoR), probability of constraint violation (Risk) and Standard Deviation of the solutions for instances with maximum deviation of 10%, considering the single knapsack uncertainty set.

		Dev 10%														
		C1					**R1**					**RC1**				
Δ^q	Δ^t	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**
0	0	52.2%	0.0%	0.0%	0.0%	0.7%	0.0%	0.0%	0.0%	0.0%	14.0%	30.3%	0.0%	0.0%	0.0%	15.7%
20	0	0.0%	18.6%	15.8%	23.2%	2.6%	0.0%	0.0%	0.0%	0.0%	14.0%	30.0%	10.3%	0.0%	21.0%	9.6%
40	0	0.0%	18.7%	15.8%	23.2%	2.6%	0.0%	0.0%	0.0%	0.4%	14.0%	0.0%	10.3%	0.0%	21.0%	9.6%
60	0	0.0%	18.7%	15.8%	23.2%	2.6%	0.0%	0.0%	0.0%	0.0%	14.0%	0.0%	10.7%	0.0%	22.4%	9.4%
0	0	0.0%	0.0%	0.0%	0.0%	0.7%	23.1%	0.0%	0.0%	0.0%	14.0%	29.2%	0.0%	0.0%	0.0%	15.7%
0	20	0.0%	0.0%	0.0%	0.0%	0.4%	6.2%	1.3%	0.0%	12.4%	13.1%	17.7%	0.6%	0.0%	2.2%	15.6%
0	40	0.0%	0.4%	0.0%	3.0%	1.2%	0.0%	3.9%	0.0%	20.0%	12.1%	0.0%	2.4%	0.0%	7.4%	14.5%
0	60	0.0%	1.1%	0.0%	5.2%	1.6%	0.0%	6.2%	0.0%	25.0%	10.8%	0.0%	3.9%	0.1%	9.9%	13.6%
0	0	52.9%	0.0%	0.0%	0.0%	0.7%	23.1%	0.0%	0.0%	0.0%	14.0%	52.5%	0.0%	0.0%	0.0%	15.7%
20	20	0.0%	19.1%	15.8%	23.2%	2.3%	6.2%	1.8%	0.0%	7.5%	12.7%	40.6%	10.9%	0.0%	25.3%	9.9%
40	40	0.0%	21.1%	15.8%	26.8%	2.2%	0.0%	4.2%	0.0%	12.0%	11.7%	0.0%	14.6%	0.9%	24.5%	7.6%
60	60	0.0%	20.6%	18.0%	23.2%	1.6%	0.0%	8.5%	0.0%	25.4%	9.9%	0.0%	16.2%	1.2%	29.5%	7.3%
All		8.8%	9.9%	8.1%	12.6%	1.6%	4.9%	2.1%	0.0%	8.6%	12.8%	16.7%	6.7%	0.2%	13.6%	12.0%
		C2					**R2**					**RC2**				
Δ^q	Δ^t	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**
0	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
20	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.3%	9.1%	0.0%	0.1%	0.0%	0.5%	8.4%
40	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.2%	0.0%	2.4%	8.9%	0.0%	0.4%	0.0%	2.8%	7.9%
60	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.2%	0.0%	1.5%	9.0%	0.0%	0.3%	0.0%	1.2%	8.1%
0	0	0.0%	0.0%	0.0%	0.0%	0.2%	9.4%	0.0%	0.0%	0.0%	9.1%	1.3%	0.0%	0.0%	0.0%	8.4%
0	20	0.0%	0.0%	0.0%	0.0%	0.2%	9.4%	0.7%	0.0%	3.6%	8.7%	1.5%	0.2%	0.0%	1.2%	8.3%
0	40	0.0%	0.6%	0.0%	4.7%	1.3%	0.0%	3.5%	0.0%	10.4%	7.1%	0.0%	2.9%	0.0%	12.5%	5.8%
0	60	0.0%	1.9%	0.0%	10.4%	3.3%	0.0%	5.9%	0.0%	15.3%	5.8%	0.0%	3.8%	0.0%	15.1%	4.9%
0	0	0.0%	0.0%	0.0%	0.0%	0.2%	9.4%	0.0%	0.0%	0.0%	9.1%	1.3%	0.0%	0.0%	0.0%	8.4%
20	20	0.0%	0.3%	0.0%	2.4%	0.5%	9.4%	1.3%	0.0%	8.0%	8.4%	1.4%	0.8%	0.0%	4.4%	7.6%
40	40	0.0%	1.5%	0.0%	10.0%	3.0%	0.0%	3.6%	0.0%	9.7%	7.2%	0.0%	2.2%	0.0%	5.3%	7.0%
60	60	0.0%	1.2%	0.0%	9.8%	2.9%	0.0%	7.5%	0.0%	32.8%	7.2%	0.0%	4.8%	0.0%	16.9%	4.9%
All		0.0%	0.5%	0.0%	3.1%	1.1%	3.1%	1.9%	0.0%	7.0%	8.2%	0.5%	1.3%	0.0%	5.0%	7.3%

Table 18 – Average, best and worst Price of robustness (PoR), probability of constraint violation (Risk) and Standard Deviation of the solutions for instances with maximum deviation of 25%, considering the single knapsack uncertainty set.

		Dev 25%														
		C1					**R1**					**RC1**				
Δ^q	Δ^t	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**
0	0	99.7%	0.0%	0.0%	0.0%	0.7%	0.0%	0.0%	0.0%	0.0%	14.0%	84.0%	0.0%	0.0%	0.0%	15.7%
20	0	66.8%	18.6%	15.8%	23.2%	2.6%	0.0%	0.0%	0.0%	0.1%	14.0%	83.9%	9.5%	0.0%	20.7%	10.2%
40	0	0.0%	23.4%	20.0%	24.9%	1.5%	0.0%	0.1%	0.0%	1.1%	13.5%	0.0%	32.4%	10.1%	46.2%	5.8%
60	0	0.0%	33.2%	31.8%	34.3%	0.7%	0.0%	0.3%	0.0%	2.0%	13.6%	0.0%	33.8%	11.2%	47.5%	5.4%
0	0	0.0%	0.0%	0.0%	0.0%	0.7%	62.5%	0.0%	0.0%	0.0%	14.0%	72.6%	0.0%	0.0%	0.0%	15.7%
0	20	0.0%	0.0%	0.0%	0.0%	0.7%	15.7%	1.7%	0.0%	8.7%	15.5%	30.2%	0.4%	0.0%	1.3%	15.4%
0	40	0.0%	0.3%	0.0%	1.8%	0.7%	0.0%	4.5%	0.0%	8.5%	14.3%	0.0%	2.9%	0.1%	8.5%	14.0%
0	60	0.0%	1.8%	0.0%	11.0%	3.6%	0.0%	8.2%	0.0%	24.2%	12.4%	0.0%	5.4%	1.1%	14.3%	14.0%
0	0	99.7%	0.0%	0.0%	0.0%	0.7%	62.1%	0.0%	0.0%	0.0%	14.0%	98.8%	0.0%	0.0%	0.0%	15.7%
20	20	70.2%	18.9%	15.8%	23.2%	2.4%	14.9%	2.1%	0.0%	9.2%	15.2%	85.5%	12.1%	0.0%	22.4%	9.3%
40	40	0.0%	25.6%	22.0%	30.3%	2.1%	0.0%	4.7%	0.0%	11.4%	14.1%	0.0%	40.8%	11.6%	63.9%	3.5%
60	60	0.0%	35.8%	32.8%	42.0%	2.1%	0.0%	7.9%	0.0%	20.5%	12.0%	0.0%	43.7%	11.6%	72.2%	4.1%
All		28.0%	13.1%	11.5%	15.9%	1.5%	12.9%	2.5%	0.0%	7.2%	13.9%	37.9%	15.1%	3.8%	24.8%	10.7%
		C2					**R2**					**RC2**				
Δ^q	Δ^t	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**
0	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
20	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.3%	8.3%
40	0	0.0%	0.1%	0.0%	0.7%	0.0%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
60	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.1%	0.0%	0.7%	9.1%	0.0%	0.3%	0.0%	1.5%	8.2%
0	0	17.1%	0.0%	0.0%	0.0%	0.2%	31.6%	0.0%	0.0%	0.0%	9.1%	11.3%	0.0%	0.0%	0.0%	8.4%
0	20	0.0%	0.1%	0.0%	0.7%	0.0%	3.4%	0.1%	0.0%	0.4%	9.1%	0.0%	0.6%	0.0%	2.2%	8.2%
0	40	0.0%	1.3%	0.0%	10.5%	3.2%	0.0%	4.0%	0.0%	11.2%	6.5%	0.0%	3.5%	0.0%	12.5%	5.7%
0	60	0.0%	2.8%	0.0%	10.0%	3.7%	0.0%	4.6%	0.0%	12.9%	6.0%	0.0%	3.6%	0.0%	15.4%	4.8%
0	0	17.0%	0.0%	0.0%	0.0%	0.2%	31.4%	0.0%	0.0%	0.0%	9.1%	11.3%	0.0%	0.0%	0.0%	8.4%
20	20	0.0%	0.1%	0.0%	0.7%	0.0%	3.4%	1.4%	0.0%	9.6%	8.5%	5.0%	0.6%	0.0%	2.8%	7.7%
40	40	0.0%	0.8%	0.0%	6.6%	1.9%	0.0%	4.1%	0.0%	15.1%	6.1%	0.0%	3.8%	0.0%	12.4%	4.9%
60	60	0.0%	1.9%	0.0%	8.8%	2.8%	0.0%	27.9%	0.0%	25.5%	50.5%	0.0%	3.4%	0.0%	18.5%	5.4%
All		2.8%	0.6%	0.0%	3.2%	1.1%	5.8%	3.5%	0.0%	6.3%	11.8%	2.3%	1.3%	0.0%	5.5%	7.2%

Table 19 – Average, best and worst Price of robustness (PoR), probability of constraint violation (Risk) and Standard Deviation of the solutions for instances with maximum deviation of 50%, considering the single knapsack uncertainty set.

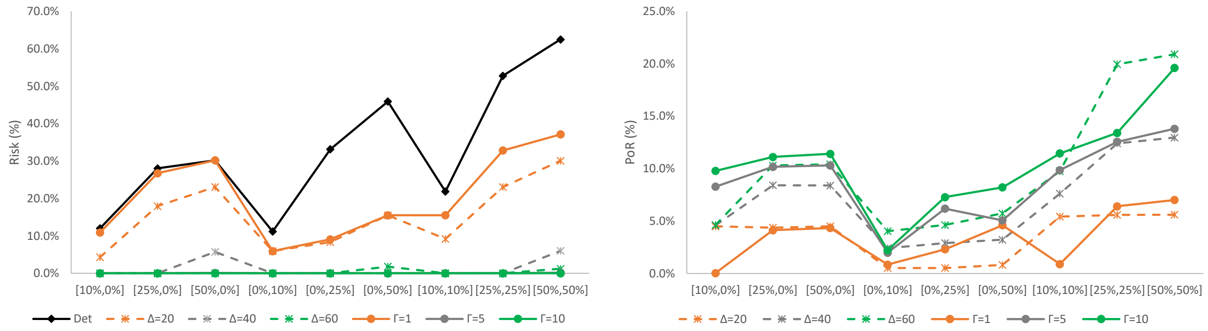
		Dev 50%														
		C1					**R1**					**RC1**				
$\mathbf{\Delta^q}$	$\mathbf{\Delta^t}$	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**
0	0	100.0%	0.0%	0.0%	0.0%	0.7%	0.0%	0.0%	0.0%	0.0%	14.0%	98.8%	0.0%	0.0%	0.0%	15.7%
20	0	100.0%	18.6%	15.8%	23.2%	2.6%	0.0%	0.0%	0.0%	0.0%	14.0%	98.9%	10.3%	0.0%	21.0%	9.6%
40	0	63.9%	23.3%	20.0%	24.9%	1.6%	0.0%	0.0%	0.0%	0.1%	13.9%	0.1%	32.2%	10.1%	46.2%	5.9%
60	0	0.0%	33.4%	31.9%	35.6%	0.7%	0.0%	0.1%	0.0%	0.8%	13.4%	0.0%	34.9%	11.2%	54.2%	5.3%
0	0	0.0%	0.0%	0.0%	0.0%	0.7%	95.1%	0.0%	0.0%	0.0%	14.0%	96.5%	0.0%	0.0%	0.0%	15.7%
0	20	0.0%	0.0%	0.0%	0.2%	0.7%	57.9%	2.1%	0.0%	11.2%	15.3%	38.3%	0.5%	0.0%	2.3%	15.6%
0	40	0.0%	0.5%	0.0%	2.6%	0.6%	0.0%	6.9%	0.0%	21.2%	16.7%	0.0%	3.2%	0.3%	7.7%	14.2%
0	60	0.0%	2.2%	0.0%	10.6%	3.6%	0.0%	10.6%	1.1%	40.1%	15.7%	0.0%	5.8%	0.3%	15.3%	14.0%
0	0	100.0%	0.0%	0.0%	0.0%	0.7%	95.1%	0.0%	0.0%	0.0%	14.0%	100.0%	0.0%	0.0%	0.0%	15.7%
20	20	99.8%	18.9%	15.8%	23.2%	2.4%	56.3%	1.8%	0.0%	8.5%	15.5%	85.6%	11.3%	0.0%	22.1%	9.4%
40	40	67.0%	26.3%	24.0%	32.3%	2.2%	0.0%	6.8%	1.1%	17.1%	16.8%	0.0%	40.8%	12.4%	60.4%	3.9%
60	60	13.5%	36.0%	32.8%	40.4%	2.0%	0.0%	11.0%	1.1%	36.7%	15.7%	0.0%	45.7%	12.8%	70.4%	2.8%
All		45.4%	13.3%	11.7%	16.1%	1.5%	25.4%	3.3%	0.3%	11.3%	14.9%	43.2%	15.4%	3.9%	25.0%	10.6%
		C2					**R2**					**RC2**				
$\mathbf{\Delta^q}$	$\mathbf{\Delta^t}$	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**	**Risk**	**PoR**	**Best**	**Worst**	**Std. Dev.**
0	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	9.1%	0.0%	0.0%	0.0%	0.0%	8.4%
20	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.3%	9.1%	0.0%	0.2%	0.0%	1.4%	8.2%
40	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.1%	0.0%	0.8%	9.0%	0.0%	0.1%	0.0%	1.1%	8.2%
60	0	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.1%	0.0%	0.7%	9.1%	0.0%	0.1%	0.0%	0.8%	8.5%
0	0	20.0%	0.0%	0.0%	0.0%	0.2%	41.6%	0.0%	0.0%	0.0%	9.1%	12.5%	0.0%	0.0%	0.0%	8.4%
0	20	0.0%	0.3%	0.0%	2.4%	0.5%	9.1%	0.9%	0.0%	9.4%	8.6%	4.4%	0.4%	0.0%	2.0%	8.3%
0	40	0.0%	1.1%	0.0%	8.4%	2.5%	0.0%	3.4%	0.0%	10.3%	6.7%	0.0%	2.6%	0.0%	11.3%	5.6%
0	60	19.8%	2.2%	0.0%	13.8%	4.2%	0.0%	7.4%	0.3%	20.1%	6.3%	0.0%	3.5%	0.0%	16.0%	5.2%
0	0	20.0%	0.0%	0.0%	0.0%	0.2%	41.6%	0.0%	0.0%	0.0%	9.1%	12.5%	0.0%	0.0%	0.0%	8.4%
20	20	0.0%	0.0%	0.0%	0.0%	0.2%	9.1%	1.6%	0.0%	8.8%	8.1%	4.4%	0.5%	0.0%	2.8%	7.9%
40	40	0.0%	1.5%	0.0%	8.1%	2.6%	0.0%	4.3%	0.0%	15.7%	6.6%	0.0%	2.7%	0.0%	7.5%	6.9%
60	60	0.0%	1.7%	0.0%	10.7%	3.2%	0.0%	3.3%	0.0%	16.4%	6.6%	0.0%	2.4%	0.0%	6.9%	4.2%
All		5.0%	0.6%	0.0%	3.6%	1.2%	8.5%	1.8%	0.0%	6.9%	8.1%	2.8%	1.0%	0.0%	4.1%	7.4%

conservative as the budget of uncertainty increases. Additionally, similarly to the results for the cardinality constrained uncertainty set, the standard deviation of the robust solutions are relatively close to, and sometimes even smaller than, the deterministic one. This implies that the robust solutions provided by the algorithm, for a given combination of *Dev* and Δ, have a similar level of increment for all instances of the same class. 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.

Regarding the maximum deviation parameters (*Dev*), we note they mostly affect the conservatism and risks of solutions. It is possible to evaluate this impact by analysing a fixed budget parameters with different deviation levels. A good example from this behavior is found in the instances from class R1 with deviation on both demand and travel time. By fixing $\Delta^d = \Delta^t = 20$ we note that the costs increase from an average PoR of 1.8% when $Dev = 10\%$, to 2.1% when $Dev = 25\%$ and 3.1% when $Dev = 50\%$. The increased costs arise from the models trying to create a solution capable of absorb this additional variability. Despite of this, the risk also increased for each level of deviation, for instance, in the previous example the average risks were 6.2% when $Dev = 10\%$, 14.9% when $Dev = 25\%$ and 56.3% when $Dev = 50\%$. This happens due to the budget taken being not enough to completely protect from this additional deviation.

Notably, with the chosen uncertainty budgets, the solutions using the knapsack uncertainty set were relatively similar to those considering the cardinality constrained uncertainty set with the same deviation level (*Dev*). This can be visualized in Figure 10, which presents the average

Risk (a) and PoR (b) for each combination of [Dev^q,Dev^t] of the solutions obtained considering the cardinality constrained and the knapsack uncertainty set.



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

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

Figure 10 – Average Risk and PoR of for different deviation levels [Dev^q,Dev^t] and uncertainty sets.

Particularly, the solutions considering the cardinality constrained set with $\Gamma = 1$ were relatively similar to the solutions with the knapsack set and $\Delta = 20$, although the latter presented slightly lower risks. However, the solutions considering $\Delta = 20$ were relatively cheaper on instances with deviations exclusively on travel times, while retaining the same robustness level, highlighting advantages of using the knapsack uncertainty set. Similarly, the solutions considering $\Delta = 40$ and $\Delta = 60$ outperform those with $\Gamma = 5$ and $\Gamma = 10$ in terms of risk and cost in the groups with $Dev = 10\%$ and $Dev = 25\%$. In the instances with $Dev = 50\%$, on the other hand, there is a trade-off between the solutions with $\Gamma = 5$ and $\Gamma = 10$, which have zero risk, and the solutions with $\Delta = 40$ and $\Delta = 60$, which usually had slightly lower costs in exchange for slightly higher risks. These results show that using the knapsack uncertainty set might be useful from the decision maker's point of view, as it can provide competitive solutions that can outperform some solutions with the cardinality uncertainty set in some configurations.

Finally, by analyzing each instance class individually, we note a similar behavior as identified in the previous uncertainty set. The instances with clustered customers (C1 and C2) presented a strong natural resistance to travel time deviation, in which the deterministic solutions were able to absorb all of this uncertainty (C1) or it was possible to null the risks with a negligible increase in costs (C2). On the other hand, the instances with randomly distributed customers (R1 and R2) were immune to uncertainty on demand, although they were still affected by variability in travel time. Class RC1, which is composed by instances with mixed geographic distribution were susceptible by both demand and travel time, while instances from class RC2 are only affected by travel time uncertainty due to the large capacity on their vehicles. We already discussed the computational times in the previous topic, where we compared the CF formulation with the MTZ model, but, in short, instance classes RC1 and RC2 were the hardest to solve while there was no relevant difference between instances with randomly distributed and clustered customers regarding the average running times.

4 Robust aircraft routing for on-demand air transportation

In the previous chapters, we investigated robust optimization approaches for the VRPTW, which is classical and traditional problem in the VRP literature. We now extend these approaches to model and solve a real-world application of the VRPTW. This is not a straightforward application though, as we need to further incorporate additional characteristics that are faced by the company in practice. The addressed problem commonly arises in the context of on-demand air transportation companies, particularly in services in which customers choose the desired flights and times and the company must service them while focusing on minimizing its operational costs (Yao et al., 2008). In addition to designing the routes and schedules for each aircraft, we also need to guarantee that these schedules obey the crew working hours and regulations, which makes this problem even more challenging to solve.

Our contributions in this context are threefold. First, we develop a B&C method to effectively solve the deterministic version of the addressed problem. This method starts from a formulation that determines only the aircraft routes, and then gradually imposes crew requirements into this model. As a second contribution, we design a labeling algorithm to verify the feasibility of aircraft routes regarding the crew regulations. In particular, we address most crew requirements found in literature Haouari2019 and introduce the *split duty rule*, commonly used by airlines to extend a crew's duty based on the time they remain grounded. Finally, we extend the model and B&C method to incorporate uncertainty on travel times via robust optimization (RO).

The remainder of this chapter is structured as follows. The description of the real-world problem is given in Section 4.1. We then present the deterministic and robust aircraft routing models that are used as the starting model in the proposed B&C algorithm in Section 4.2. We follow by explaining the labeling-based B&C algorithm designed to insert crew requirements into the problems and its robust extension in Section 4.3. Then, in Section 4.4, we present the computational results of the proposed algorithm using real-life data provided by a company.

4.1 Problem description

When dealing with real-world applications in logistics, traditional mathematical programming formulations are usually insufficient to represent all the company's requirements and hence different constraints need to be added to these formulations. Particularly, in aircraft routing, the subject of our study, some common additional characteristics usually considered are fleet heterogeneity, i.e., each aircraft has different costs, speed and types of flights it can service (Yao et al., 2008); different types of services, e.g., customer flight or maintenance event (Zwan; Wils; Ghijs, 2011); and crew requirements such as minimum rest time and maximum flight time (Shebalov; Klabjan, 2006; Haouari; Mansour; Sherali, 2019).

A particular type of air company, whose services have been significantly growing in the last years, relates to on-demand air transportation services (Yang et al., 2008; Yao et al., 2008; Zwan; Wils; Ghijs, 2011). These services typically arise in air taxi and fractional ownership contract companies. Unlike traditional airlines, these companies allow customers to choose the itinerary, the departing time and the aircraft type they need. The company must ensure that the desired aircraft will be available in the right place on time.

Fractional ownership companies, similar to the one studied in this work, are air transportation companies in which customers acquire a contract giving them a fraction of an airship (Lacasse-Guay; Desaulniers; Soumis, 2010). This contract owner, then, has the rights to fly in the airship for a determined number of hours yearly. Customers have complete freedom to choose the time and destination they wish. The main advantage of this type of service for the contract owner is having a similar flexibility to owning a particular aircraft at a fraction of the costs.

For the company, this type of services results into some contractual obligations (Lacasse-Guay; Desaulniers; Soumis, 2010). Namely, if the flight was booked at least 4 hours before the departure, the company must guarantee that there will be an aircraft similar to, or better than, the one hired by the customer ready at the departure airport on the agreed time. Thus, these companies usually deal with a more dynamic and unpredictable environment than traditional airlines. This results in a relatively short planning horizon (48-72h), that must be quickly updated with new demands.

Since the flights with costumers are mandatory and generate revenue, the focus for this type of company is to minimize their operational costs, often related to the positioning flights. In this type of flight, the aircraft travels without passengers to a customer's departure airport without direct revenue (Yao et al., 2008). These positioning flights can comprise 35% or more of the company's total flying time. Since the company bears the costs from this type of flight, focusing on minimizing them can improve its efficiency and competitiveness.

Additionally, when scheduling routes, the company must follow some requirements. Among those, are internal policies regarding the aircraft routing planning, such as the maximum service delay and type of aircraft that can be used for each demand. Moreover, the company must also comply with international regulations related to crew working hours and resting periods. These requirements are summarized in Table 20 and detailed in the following subsections.

Table 20 – Requirements for the route planning

Aircraft routing rules	Aircraft type for customer request	Requested type or better
	Aircraft in a maintenance event	The one requested
	Maximum delay for customer request	15 min
	Maximum delay for a maintenance event	1440 min | 24h
	Maximum time a maintenance event can be brought forward	1440 min | 24h
Crew requirement	Maximum duty time	780 min | 13h
	Maximum flying time in a duty	600 min | 10h
	Minimum resting time	600 min | 10h
	Presentation time before a duty start (PRE)	40 min
	Presentation time at the end of a duty (POS)	30 min

Considering these routing and crew requirements simultaneously in the planning stage, albeit a more complex activity, brings considerable economic advantages to the solution (Sherali; Bish; Zhu, 2006; Mercier; Soumis, 2007; Dunbar; Froyland; Wu, 2014). This is compatible with the company's focus on minimizing operating costs and can be interesting to apply in practice. However, it is worth noting that due to the high dynamics of this sector, it is essential to not only generating an efficient solution, but also obtaining it in a short time span. This is particularly more relevant in on-demand air transportation companies, such as the one studied in this paper, because, by contract, customers can request a flight as little as four hours in advance. If for some reason there is no aircraft available to service a customer, there is the possibility of outsourcing the flight to another company, although this might be costly.

In the literature addressing models and solution approaches for aiding decision-making in the described context, authors typically resort to specialized algorithms to effectively obtain solutions, such as decomposition techniques (Mercier; Soumis, 2007), branch-and-price methods (Yang et al., 2008) and heuristic approaches (Dunbar; Froyland; Wu, 2014). We emphasize that the majority of works addressing integrated routing and crew requirements were developed for traditional companies (Mercier; Soumis, 2007; Dunbar; Froyland; Wu, 2014). Furthermore, we are not aware of any other study that considers all aspects of the situation addressed in this work, in the context of on-demand air transportation.

4.1.1 Routing planning requirements

When defining the aircraft routes, the company must ensure that there is an aircraft available at the departure airport of any given request with a maximum delay of 15 minutes. Additionally, this aircraft must be of the same type or better than the one hired. Servicing a customer with an aircraft better, and more expensive, than the one defined in the contract incur in additional costs called *upgrade*. Despite the additional cost, an upgrade can be strategically used by the company to save money with positioning flights. This allows the company to reduce their total costs and meet requests that would be impossible otherwise (Munari et al., 2019). The downside is that this freedom considerably increases the complexity of the decision-making process. It should be noted that the company's policy usually does not allow customers to be serviced by aircraft that are inferior to the one requested (downgrade).

Moreover, the company is responsible for the fleet maintenance. Periodically, each aircraft must go through a planned checking and maintenance process, becoming unavailable until this event is finished. Although the start time of a maintenance event is pre-scheduled, the company is typically allowed to advance or delay this time within a relatively large margin of 24 hours. Thus, maintenance can be seen as a request in which the aircraft must be stationary at a single airport for a certain period of time and which presents a comprehensive time window, allowing greater flexibility.

4.1.2 Crew Requirements

Another important point the company must consider while planning the aircraft routes is the various regulations related to the crew's working and resting times. Some of the most relevant are the maximum time allowed in duty, the maximum accumulated flight time in duty and the minimum resting time between two duties (Shebalov; Klabjan, 2006; Haouari; Mansour; Sherali, 2019). For the first requirement, the company must allow a maximum duty ($MaxDuty$) of 13 hours. This is enforced by international regulations due to the risks associated with crew fatigue. Additionally, since flying the aircraft is an exhausting activity, regulators also impose a maximum time the crew is allowed to fly in a single duty (*MaxFly*) of 10 hours. Eventually, there might be situations in which the crew is asked to work overtime (when total duty time exceeds the *MaxDuty* or the total flying time surpasses *MaxFly*). In these cases, the company may negotiate with the crew to work this additional time in exchange for additional rest time or extra pay. In this work, we incorporate this strategy by allowing the crew to work overtime, but punishing the objective function with additional cost per minute worked.

Moreover, the company must ensure the crew has at least 10 hours (*MinRest*) of uninterrupted rest between two consecutive duties. Every time a complete rest occurs, the accumulated *duty* is set to zero. A particularity on maintenance events is that, since the aircraft is parked during the entire process, the crew is allowed to rest during the event and, hence, the company can take advantage of the maintenance time. If maintenance lasts longer than *MinRest*, it is interesting to extend the crew's free time until it is finished, as there is no reason to keep the pilots on stand-by if the aircraft is not ready yet. Conversely, if the duration of maintenance is less than *MinRest*, the crew still needs to rest for the minimum time, even if the aircraft is available earlier. Anyway, it is interesting to take advantage of maintenance to cover part of, or completely, the resting time, as this allows for time saving and allows some solutions that would be impossible otherwise.

Other important elements imposed to the company are the crew presentation times in a duty to check the aircraft and analyze the weather conditions and itinerary. The first one occurs at the beginning of the duty (PRE), taking 40 minutes. This first preparation time is counted within the crew's duty. The second one happens at the ending of a duty (POS), lasting 30 minutes and is neither counted in the duty nor the rest time. Thus, whenever there is a rest in the planning horizon we must also insert the presentation times at the start and end of the duty. Figure 11 illustrate these concepts by showing two examples of the different resting behaviors. In both examples we have a route that, during the planning horizon, starts from a rest, followed by a customer's request. Then, the crew have a positioning flight in order to service another request, takes a rest and finally services a last request. The only difference between the examples is that in Example 1 the second request serviced by the aircraft is made by a customer while in Example 2 we have a maintenance event. We can note that in Example 2, the crew is allowed to start its rest after landing and do not need to be available during the whole maintenance (represented by the green outline in the rest period), while in Example 1 the crew is required to finish the service before starting their rest.

Another important strategy that the company employs regarding the crew total rest time is the *split duty rule*, in which the crew is allowed to take a break, a pause period shorter than a normal rest, and extend the duty duration by a fraction of the break time (BT). Essentially, depending on the total break duration, the company is allowed to compute only a fraction of that time as paid duty time, following the calculations in Table 21. If the break lasts for less than 90 minutes, it is computed in the total duty time normally. Now, if the BT is longer than 90 minutes, but under 6 hours, any minute over 90 minutes is computed as only half. Finally, if staying stationary for over 6 hours, but less than the minimum resting time (10 hours), the company may add only 60 minutes in the accumulated duty time. If the break time would exceed the minimum rest time, the crew takes a normal rest instead, setting the accumulated duty time

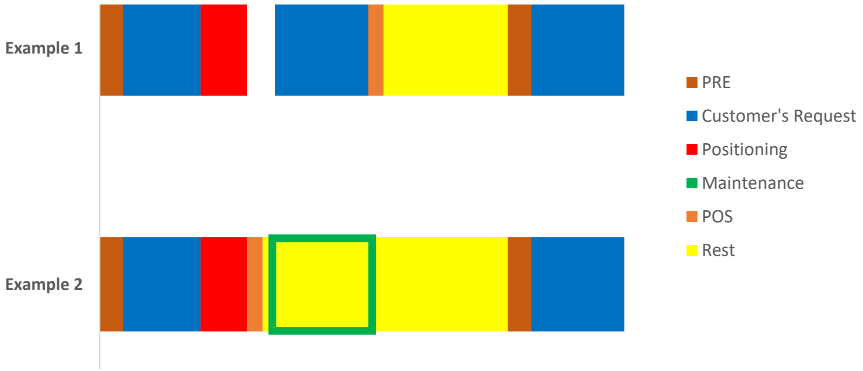


Figure 11 – Examples of rest behaviour for two vehicle routes.

to zero. To the best of our knowledge, no work in literature utilize this common strategy in a solution method in an aircraft routing problem.

Table 21 – Split duty rule

	Break time duration (min)	**Increase in duty time**
Split duty rule	BT<90	BT
	90≤ BT<360	$\frac{BT-90}{2}+90$
	360≤ BT<*MinRest*	60
	MinRest≤ BT	Mandatory rest

4.2 Aircraft routing formulation

The B&C algorithm we propose in the next section is based on formulations of the aircraft routing problem that is initially created without considering any crew requirements (characteristics discussed in subsection 4.1.2). In this section, we present two base models that are used to provide the initial solutions and that will later be used in the B&C algorithm. One is based on MTZ constraints and has already been introduced by Munari and Alvarez (2019), whereas the other was inspired by the CF formulation proposed in Chapter 3. The deterministic formulation of these models are detailed in Subsection 4.2.1. These models are then extended to account for uncertainty on travel times by following the RO paradigms and considering the cardinality constrained uncertainty set, in Subsection 4.2.2, and the knapsack uncertainty set, in Subsection 4.2.3.

4.2.1 Deterministic aircraft routing formulation

4.2.1.1 MTZ-based aircraft routing formulation

The model proposed by Munari and Alvarez (2019) addresses some of the critical requirements of the company, such as maintenance events and possibility of upgrades, while minimizing the overall positioning and upgrade costs. Let V be a fleet of aircraft, partitioned in subsets V_p for each p in an ordered set of aircraft types P. The set of all requests, R, is composed by the union of a dummy request 0, the set L of customer requests, and the set M

of maintenance events. The airports are represented by the set K. The parameters used in the model are as follow:

- c_v: flight cost per unit of time of aircraft $v \in V$;
- C_{vrs}: repositioning cost of aircraft $v \in V$ when it fulfills the request $s \in R$ immediately after request $r \in R$;
- T^p_{ij}: travel time between airports $i \in K$ and $j \in K$ for an aircraft of type p;
- AV_v: the instant when the aircraft $v \in V$ becomes available to fly for the first time in the planning horizon;
- k_v: is the initial airport of aircraft $v \in V$ on the start in the planning horizon;
- t_v: is the type of aircraft $v \in V$;
- TAT^r_k: the turnaround time, i.e., the waiting time between flights at airport $k \in K$ before servicing the request $r \in R$;
- ST_r: the starting time of the request $r \in R$;
- Δ_L: maximum time shift allowed to start a customer request;
- Δ_M: maximum time shift allowed to start a maintenance event;
- i_r: origin airport of request $r \in R$;
- j_r: destination airport of request $r \in R$;
- p_r: aircraft type required for the request $r \in R$;
- TL_r: maintenance time of request $r \in M$;
- v_r: index of the aircraft that must undergo the maintenance event $r \in M$.

A feature that makes this model stand out among other compact formulations (Yang et al., 2008; Jamili, 2017) is that instead of the usual graph representation for the VRP, in which the airports are represented by nodes and the decision variables are associated to arcs linking the nodes, Munari and Alvarez (2019) represented the problem using a request network, in which nodes are associated to customer requests. Figure 12 is a visual representation of this network.

In this structure, every aircraft of the fleet starts in a dummy request used to represent the start of a route in the planning horizon. From there, an aircraft can be assigned to any compatible request r, in which a flight from airport i_r to airport j_r is performed. After servicing a request, the aircraft can be assigned to another one, which may require a positioning flight, or return to the dummy request, representing the end of its route in the planning horizon. As observed by the authors, this approach is likely to reduce the problem size and result in better computational times. With this graph representation in mind, the decision variables used in their model are as follows:

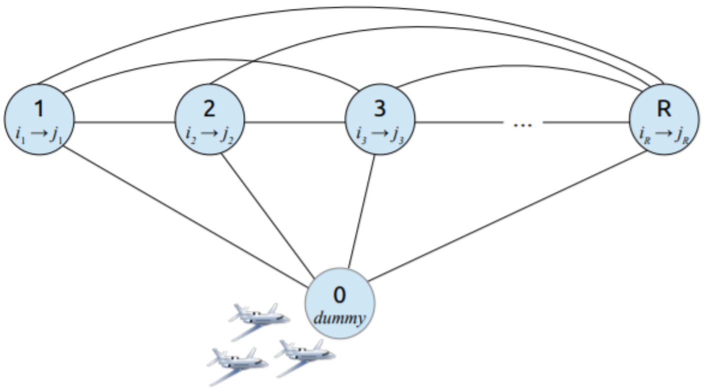


Figure 12 – Representation of the flow network through requests. Source: Munari and Alvarez (2019)

- $y_{vrs} = \begin{cases} 1, & \text{if aircraft } v \in V \text{services request } s \in R \text{ immediately after request } r \in R; \\ 0, & \text{otherwise.} \end{cases}$
- w_r: earliest time that request $r \in R$ can start.

We extended this formulation to consider the possibility of outsourcing a customer's request if needed. For this purpose, we define a new binary decision variable out_r, which represents if request $r \in L$ is outsourced in the solution, and parameter $Cout_r$, which represents the cost of outsourcing request r.

Finally, with all this information, the model is stated as:

$$\min \sum_{v\in V}\sum_{r\in R}\sum_{s\in R} C_{vrs}y_{vrs} + \sum_{r\in L}\sum_{\substack{s\in R\\ s\neq r}}\sum_{\substack{p\in P\\ p>p_r}}\sum_{v\in V_p}(c_{t_v}T^{t_v}_{i_r j_r} - c_{p_r}T^{p_r}_{i_r j_r})y_{vrs} + \sum_{r\in L} Cout_r out_r \tag{4.2.1}$$

$$\text{s.t.} \sum_{\substack{p\in P\\ p\geq p_r}}\sum_{v\in V_p}\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} + out_r = 1, \qquad r\in L, \tag{4.2.2}$$

$$\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} = 1, \qquad r\in M, v\in V, \tag{4.2.3}$$

$$\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} = \sum_{\substack{s\in R\\ s\neq r}} y_{vsr}, \qquad v\in V, r\in R, r>0, \tag{4.2.4}$$

$$\sum_{s\in R} y_{v0s} = 1 = \sum_{r\in R} y_{vr0}, \qquad v\in V, \tag{4.2.5}$$

$$ST_r \leq w_r \leq ST_r + \Delta_L, \qquad r\in L, \tag{4.2.6}$$

$$ST_r - \Delta_M \leq w_r \leq ST_r + \Delta_M, \qquad r\in M, \tag{4.2.7}$$

$$w_s \geq w_r + \sum_{v\in V}(T^{p_r}_{i_r j_r} + TAT^r_{j_r} + T^{p_s}_{j_r i_s} + TAT^s_{i_s})y_{vrs} + M^1_{rs}(\sum_{v\in V} y_{vrs} - 1),$$
$$r\in L, s\in R, r\neq s, s>0, j_r\neq i_s, \tag{4.2.8}$$

$$w_s \geq w_r + \sum_{v\in V}(T^{p_r}_{i_r j_r} + TAT^s_{js})y_{vrs} + M^2_{rs}(\sum_{v\in V} y_{vrs} - 1),$$
$$r\in L, s\in R, r\neq s, s>0, j_r = i_s, \tag{4.2.9}$$

$$w_s \geq (AV_v + T^{t_v}_{k_v i_s} + TAT^s_{i_s})y_{v0s}, \qquad s\in L, v\in V, k_v\neq i_s, \tag{4.2.10}$$

$$w_s \geq (AV_v + TAT^s_{i_s})y_{v0s}, \qquad s \in L, v \in V, k_v = i_s, \quad (4.2.11)$$

$$w_s \geq w_r + TL_r + T^{p_r}_{j_r i_s} + TAT^s_{i_s} + M^3_{rs}(y_{v_r rs} - 1),$$
$$r \in M, s \in R, r \neq s, s > 0, j_r \neq i_s, \quad (4.2.12)$$

$$w_s \geq w_r + TAT^s_{i_s} + M^4_{rs}(y_{v_r rs} - 1),$$
$$r \in M, s \in R, r \neq s, s > 0, j_r = i_s, \quad (4.2.13)$$

$$w_s \geq (AV_{v_s} + T^{p_s}_{k_{v_s} i_s})y_{v_s 0s}, \qquad s \in M, k_{v_s} \neq i_s, \quad (4.2.14)$$

$$w_s \geq y_{v_s 0s} AV_{v_s}, \qquad s \in M, k_{v_s} = i_s, \quad (4.2.15)$$

$$w_r \geq 0, \qquad r \in R, \quad (4.2.16)$$

$$y_{vrs} \in \{0,1\}, out_r \in \{0,1\}, \qquad v \in V, s, r \in R. \quad (4.2.17)$$

The objective function (4.2.1) consists of minimizing the operational costs. These costs are composed by aircraft positioning, which arise in trips that the aircraft fly without any passenger, represented by the first term; the upgrade cost, the increase in cost when servicing a request with an aircraft better than the one contracted, represented by the second term; and the outsourcing costs, in which another company is contracted to service a specific request, represented by the third term. Constraints (4.2.2) ensure that each customers' request $r \in L$ is serviced exactly once by an aircraft with the appropriate type (or better) or by outsourcing. Likewise, constraints (4.2.3) ensure that every maintenance event $r \in M$ is attended by the corresponding aircraft (a maintenance event is defined for a particular aircraft). The flow of aircraft is imposed by constraints (4.2.4). These constraints enforce that an aircraft can service a request only if it services one request before it and another after it (both these requests can be a dummy request 0). Constraints (4.2.5) guarantee that the first and last requests for every aircraft are dummy requests. Constraints (4.2.6)-(4.2.7) impose that the time windows for costumers and maintenance events are respected, respectively. Constraints (4.2.8)-(4.2.11) compute the starting time of requests that are serviced right after customer requests. Constraints (4.2.8) compute the starting time for request $s \in R$ for the case in which the departure airport for this request is different from the airport from its previous request. Constraints (4.2.9) are similar, but consider the case where the destination airport from previous request and the departure airport from request $s \in L$ are the same. Constraints (4.2.10) and (4.2.11) compute the starting time for the first request on the planning horizon for each aircraft, if it is a customer request; the former is activated when a repositioning flight is needed and the latter is activated when the aircraft already is in the departure airport of request $s \in L$. Constraints (4.2.12)-(4.2.13) act similarly to (4.2.8)-(4.2.9), but for requests that succeed maintenance events. Likewise, constraints (4.2.14)-(4.2.15) refer to the first request of each aircraft in the planning horizon, just like in (4.2.10)-(4.2.11), but for the case in which the first request is a maintenance one. An important difference between this type of request, when compared to customer requests, is that they must be serviced by a particular aircraft, and not by any aircraft of one or more types. Moreover, in these requests, the vehicle must stay still in the ground during the whole service, instead of flying to other location, during a certain time (TL_r). In addition to compute the starting times of every request, constraints (4.2.8)-(4.2.15) also prevent subtours. Finally, constraints (4.2.16) and (4.2.17) state the domain of the decision variables.

The size of the big-M constants strongly impact the quality of the linear relaxation.

To avoid a big-M excessively large, Munari and Alvarez (2019) suggests $M^1_{rs} = ST_r + \Delta_L + T^{p_r}_{i_r j_r} + TAT^s_{j_r} + T^{p_s}_{j_r i_s} + TAT^s_{i_s}$; $M^2_{rs} = ST_r + \Delta_L + T^{p_r}_{i_r j_r} + TAT^s_{i_s}$, in constraints (4.2.9); $M^3_{rs} = ST_r + \Delta_M + TL_r + T^{p_r}_{j_r i_s} + TAT^s_{i_s}$, in constraints (4.2.12); and finally $M^4_{rs} = ST_r + \Delta_M + TL_r + TAT^s_{i_s}$, in constraints (4.2.13).

A robust counterpart of formulation (4.2.1)–(4.2.17) was developed by Campos, Alvarez and Munari (2019) considering uncertainty on travel times and using the cardinality constrained set, which is presented in Subsection 4.2.2 for completeness sake. Before that, we develop a novel formulation for the deterministic problem based on the CF formulation, similarly as the one presented in Chapter 3 for the traditional VRPTW.

4.2.1.2 CF aircraft routing formulation

We can develop a different compact model for the addressed deterministic aircraft routing problem based on the CF formulation proposed in Section 3.1 for the traditional VRPTW. This formulation was expected to provide stronger liner relaxation and possibly better deterministic solutions than the MTZ model, similarly to the results in the literature instances for the VRPTW. For this purpose, we define the continuous variable g_{rs} that represents the total elapsed time of the aircraft when it arrives in node s immediately after servicing request r. The parameters and remaining decision variables are the same as in the MTZ-based model (4.2.33)-(4.2.56), except for variables w_s that are not needed anymore. Therefore, the resulting CF formulation is stated as follows:

$$\min \sum_{v\in V}\sum_{r\in R}\sum_{s\in R} C_{vrs} y_{vrs} + \sum_{r\in L}\sum_{\substack{s\in R\\ s\neq r}}\sum_{\substack{p\in P\\ p>p_r}}\sum_{v\in V_p} (c_{t_v} T^{t_v}_{i_r j_r} - c_{p_r} T^{p_r}_{i_r j_r}) y_{vrs} + \sum_{r\in L} Cout_r out_r \tag{4.2.18}$$

$$\text{s.t.} \sum_{\substack{p\in P\\ p\geq p_r}}\sum_{v\in V_p}\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} + out_r = 1, \qquad r\in L, \tag{4.2.19}$$

$$\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} = 1, \qquad r\in M, v\in V, \tag{4.2.20}$$

$$\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} = \sum_{\substack{s\in R\\ s\neq r}} y_{vsr}, \qquad v\in V, r\in R, r>0, \tag{4.2.21}$$

$$\sum_{s\in R} y_{v0s} = 1 = \sum_{r\in R} y_{vr0}, \qquad v\in V, \tag{4.2.22}$$

$$\sum_{v\in V} ST_s y_{vrs} \leq g_{rs} \leq \sum_{v\in V} (ST_s + \Delta_L) y_{vrs}, \qquad r\in R, s\in L, \tag{4.2.23}$$

$$\sum_{v\in V} (ST_s - \Delta_M) y_{vrs} \leq g_{rs} \leq \sum_{v\in V} (ST_s + \Delta_M) y_{vrs}, \qquad r\in R, s\in M, \tag{4.2.24}$$

$$\begin{aligned}\sum_{\substack{s\in R, s>0\\ s\neq r, j_r\neq i_s}} &(g_{rs} - \sum_{v\in V} (T^{t_v}_{i_r j_r} + TAT^r_{j_r} + T^{t_v}_{j_r i_s} + TAT^s_{i_s}) y_{vrs}) + \\ &\sum_{\substack{s\in R, s>0\\ s\neq r, j_r=i_s}} (g_{rs} - \sum_{v\in V} (T^{t_v}_{i_r j_r} + TAT^s_{i_s}) y_{vrs}) \geq \sum_{\substack{h\in R\\ h\neq r}} g_{hr}, \qquad r\in L,\end{aligned} \tag{4.2.25}$$

$$g_{0s} - \sum_{v\in V} (AV_v + T^{t_v}_{k_v i_s} + TAT^s_{i_s}) y_{v0s} \geq 0, \qquad s\in L, k_v\neq i_s, \tag{4.2.26}$$

$$g_{0s} - \sum_{v \in V} (AV_v + TAT^s_{i_s}) y_{v0s} \geq 0, \qquad s \in L, v \in V, k_v = i_s, \quad (4.2.27)$$

$$\sum_{\substack{s \in R, s>0 \\ s \neq r, j_r \neq i_s}} (g_{rs} - (TL_r + T^{p_r}_{j_r i_s} + TAT^s_{i_s}) y_{v_r rs}) +$$

$$\sum_{\substack{s \in R, s>0 \\ s \neq r, j_r = i_s}} (g_{rs} - (TL_r + TAT^s_{i_s}) y_{v_r rs}) \geq \sum_{\substack{h \in R \\ h \neq r}} g_{hr}, \qquad r \in M, \quad (4.2.28)$$

$$g_{0s} - (AV_{v_s} + T^{p_s}_{k_{v_s} i_s}) y_{v_s 0s} \geq 0, \qquad s \in M, k_{v_s} \neq i_s, \quad (4.2.29)$$

$$g_{0s} - AV_{v_s} y_{v_s 0s} \geq 0, \qquad s \in M, k_{v_s} = i_s, \quad (4.2.30)$$

$$g_{rs} \geq 0, \qquad r, s \in R, \quad (4.2.31)$$

$$y_{vrs} \in \{0, 1\}, out_r \in \{0, 1\}, \qquad v \in V, r, s \in R. \quad (4.2.32)$$

The objective function (4.2.18) and constraints (4.2.19)-(4.2.22) are exactly the same as (4.2.1)-(4.2.5) in the MTZ-based formulation. In a similar fashion as the MTZ model, the time windows for costumer and maintenance requests are enforced by constraints (4.2.23)-(4.2.24), respectively. The time flow constraints for each type of request, given by constraints (4.2.25)-(4.2.30), act similarly as constraints (4.2.8)-(4.2.15) in the MTZ-based model, they only differ on the structure due to variables w_r being replaced for the commodity flow variables g_{rs}. Particularly constraints (4.2.25), and their counterpart for maintenance events (4.2.28), encompass two sets of constraints from the MTZ model: (4.2.8) and (4.2.9), or (4.2.12) and (4.2.13) in the case of maintenance events. The first sum in constraints (4.2.25) and (4.2.28) are analogous to constraints (4.2.8) and (4.2.12), respectively, in which we compute the elapsed time case $j_r \neq i_s$ whereas the second term refers to the constraints where $j_r = i_s$, (4.2.9 and 4.2.13). The remaining constraints related to the first node in a route (4.2.26)-(4.2.27) and (4.2.29)-(4.2.30) are analogous to their counterpart in the MTZ-based formulation (4.2.10)-(4.2.11) and (4.2.14)-(4.2.15). Remarkably, the constraints in the new formulation do not need big-M parameters, strengthening their linear relaxation. Finally, constraints (4.2.31) and (4.2.32) set the domain of decision variables.

4.2.2 RO formulations based on the cardinality constrained uncertainty set

4.2.2.1 MTZ-based RO formulation

To obtain a robust counterpart of the MTZ-based model (4.2.1)–(4.2.17), Campos, Alvarez and Munari (2019) added the index γ to the continuous variable w_s, thus resulting in the decision variable $w_{s\gamma}$. This index is to indicate the number of traversed arcs in which the travel costs attain their worst-case values. Thus, variable $w_{s\gamma}$ represents the earliest time at which request $r \in R$ can be started assuming that γ traversed arcs attain their worst-case values in travel time simultaneously. In addition, we now have that the travel time in a given arc is represented by its nominal value ($\bar{T}^{t_v}_{ij}$) and maximum deviation ($\hat{T}^{t_v}_{ij}$). With these new parameters, the robust counterpart of the MTZ-based model using the linearization technique based on dynamic programming is given as follows:

$$\min \sum_{v \in V} \sum_{r \in R} \sum_{s \in R} C_{vrs} y_{vrs} + \sum_{r \in L} \sum_{\substack{s \in R \\ s \neq r}} \sum_{\substack{p \in P \\ p > p_r}} \sum_{v \in V_p} (c_{t_v} \bar{T}^{t_v}_{i_r j_r} - c_{p_r} \bar{T}^{p_r}_{i_r j_r}) y_{vrs} + \sum_{r \in L} Cout_r out_r \quad (4.2.33)$$

$$\text{s.t.} \sum_{\substack{p\in P\\ p\geq p_r}} \sum_{v\in V_p} \sum_{\substack{s\in R\\ s\neq r}} y_{vrs} + out_r = 1, \qquad r \in L, \tag{4.2.34}$$

$$\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} = 1, \qquad r \in M, v \in V, \tag{4.2.35}$$

$$\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} = \sum_{\substack{s\in R\\ s\neq r}} y_{vsr}, \qquad v \in V, r \in R, r > 0, \tag{4.2.36}$$

$$\sum_{s\in R} y_{v0s} = 1 = \sum_{r\in R} y_{vr0}, \qquad v \in V, \tag{4.2.37}$$

$$ST_r \leq w_{r\gamma} \leq ST_r + \Delta_L, \qquad r \in L, \gamma \leq \Gamma, \tag{4.2.38}$$

$$ST_r - \Delta_M \leq w_{r\gamma} \leq ST_r + \Delta_M, \qquad r \in M, \gamma \leq \Gamma, \tag{4.2.39}$$

$$w_{s\gamma} \geq w_{r\gamma} + \sum_{v\in V} (\bar{T}^{p_r}_{i_r j_r} + TAT^r_{j_r} + \bar{T}^{p_s}_{j_r i_s} + TAT^s_{i_s}) y_{vrs} + M^1_{rs}(\sum_{v\in V} y_{vrs} - 1),$$

$$r \in L, s \in R, r \neq s, s > 0, j_r \neq i_s, \gamma \leq \Gamma, \tag{4.2.40}$$

$$w_{s\gamma} \geq w_{r(\gamma-1)} + \sum_{v\in V} (\bar{T}^{p_r}_{i_r j_r} + \hat{T}^{p_r}_{i_r j_r} + TAT^r_{j_r} + \bar{T}^{p_s}_{j_r i_s} + TAT^s_{i_s}) y_{vrs} + M^1_{rs}(\sum_{v\in V} y_{vrs} - 1),$$

$$r \in L, s \in R, r \neq s, s > 0, j_r \neq i_s, 0 < \gamma \leq \Gamma, \tag{4.2.41}$$

$$w_{s\gamma} \geq w_{r(\gamma-1)} + \sum_{v\in V} (\bar{T}^{p_r}_{i_r j_r} + TAT^r_{j_r} + \bar{T}^{p_s}_{j_r i_s} + \hat{T}^{p_s}_{j_r i_s} + TAT^s_{i_s}) y_{vrs} + M^1_{rs}(\sum_{v\in V} y_{vrs} - 1),$$

$$r \in L, s \in R, r \neq s, s > 0, j_r \neq i_s, 0 < \gamma \leq \Gamma, \tag{4.2.42}$$

$$w_{s\gamma} \geq w_{r(\gamma-2)} + \sum_{v\in V} (\bar{T}^{p_r}_{i_r j_r} + \hat{T}^{p_r}_{i_r j_r} + TAT^r_{j_r} + \bar{T}^{p_s}_{j_r i_s} + \hat{T}^{p_s}_{j_r i_s} + TAT^s_{i_s}) y_{vrs}$$

$$+ M^1_{rs}(\sum_{v\in V} y_{vrs} - 1), \qquad r \in L, s \in R, r \neq s, s > 0, j_r \neq i_s, 1 < \gamma \leq \Gamma, \tag{4.2.43}$$

$$w_{s\gamma} \geq w_{r\gamma} + \sum_{v\in V} (\bar{T}^{p_r}_{i_r j_r} + TAT^s_{js}) y_{vrs} + M^2_{rs}(\sum_{v\in V} y_{vrs} - 1),$$

$$r \in L, s \in R, r \neq s, s > 0, j_r = i_s, \gamma \leq \Gamma, \tag{4.2.44}$$

$$w_{s\gamma} \geq w_{r(\gamma-1)} + \sum_{v\in V} (\bar{T}^{p_r}_{i_r j_r} + \hat{T}^{p_r}_{i_r j_r} + TAT^s_{js}) y_{vrs} + M^2_{rs}(\sum_{v\in V} y_{vrs} - 1),$$

$$r \in L, s \in R, r \neq s, s > 0, j_r = i_s, 0 < \gamma \leq \Gamma, \tag{4.2.45}$$

$$w_{s\gamma} \geq (AV_v + \bar{T}^{t_v}_{k_v i_s} + TAT^s_{i_s}) y_{v0s}, \qquad s \in L, v \in V, k_v \neq i_s, \gamma \leq \Gamma, \tag{4.2.46}$$

$$w_{s\gamma} \geq (AV_v + \bar{T}^{t_v}_{k_v i_s} + \hat{T}^{t_v}_{k_v i_s} + TAT^s_{i_s}) y_{v0s},$$

$$s \in L, v \in V, k_v \neq i_s, 0 < \gamma \leq \Gamma, \tag{4.2.47}$$

$$w_{s\gamma} \geq (AV_v + TAT^s_{i_s}) y_{v0s}, \qquad s \in L, v \in V, k_v = i_s, \gamma \leq \Gamma, \tag{4.2.48}$$

$$w_{s\gamma} \geq w_{r\gamma} + TL_r + \bar{T}^{p_r}_{j_r i_s} + TAT^s_{i_s} + M^3_{rs}(y_{v_r rs} - 1),$$

$$r \in M, s \in R, r \neq s, s > 0, j_r \neq i_s, \gamma \leq \Gamma, \tag{4.2.49}$$

$$w_{s\gamma} \geq w_{r(\gamma-1)} + TL_r + \bar{T}^{p_r}_{j_r i_s} + \hat{T}^{p_r}_{j_r i_s} + TAT^s_{i_s} + M^3_{rs}(y_{v_r rs} - 1),$$

$$r \in M, s \in R, r \neq s, s > 0, j_r \neq i_s, 0 < \gamma \leq \Gamma, \tag{4.2.50}$$

$$w_{s\gamma} \geq w_{r\gamma} + TAT^s_{i_s} + M^4_{rs}(y_{v_r rs} - 1),$$

$$r \in M, s \in R, r \neq s, s > 0, j_r = i_s, \gamma \leq \Gamma, \tag{4.2.51}$$

$$w_{s\gamma} \geq (AV_{v_s} + \bar{T}^{p_s}_{k_{v_s} i_s}) y_{v_s 0 s}, \qquad s \in M, k_{v_s} \neq i_s, \gamma \leq \Gamma, \tag{4.2.52}$$

$$w_{s\gamma} \geq (AV_{v_s} + \bar{T}^{p_s}_{k_{v_s} i_s} + \hat{T}^{p_s}_{k_{v_s} i_s}) y_{v_s 0 s}, \qquad s \in M, k_{v_s} \neq i_s, 0 < \gamma \leq \Gamma, \quad (4.2.53)$$

$$w_{s\gamma} \geq y_{v_s 0 s} AV_{v_s}, \qquad s \in M, k_{v_s} = i_s, \gamma \leq \Gamma, \quad (4.2.54)$$

$$w_{r\gamma} \geq 0, \qquad r \in R, \gamma \leq \Gamma, \quad (4.2.55)$$

$$y_{vrs} \in \{0,1\}, out_r \in \{0,1\}, \qquad v \in V, r, s \in R. \quad (4.2.56)$$

The objective function (4.2.33) and constraints (4.2.34)-(4.2.37) are exactly the same as in the deterministic MTZ-based model. Similarly to (4.2.6)-(4.2.7), constraints (4.2.38)-(4.2.39) enforce the time windows of requests. The only difference to the deterministic model is the addition of the index γ in the time variables.

Constraints (4.2.40)-(4.2.43) are the robust counterpart of constraints (4.2.8). Essentially, these constraints compute the earliest possible starting time for requests that follow a customer request when considering that the travel times of γ traversed arcs attain their worst-case value simultaneously. They determine the greatest value among four possibilities: (a) take only the nominal travel times regarding the service of request $r \in L$ and repositioning for the next request while considering γ parameters attained their worst-case values on previous requests (4.2.40); (b) consider that only the travel times for service $r \in L$ assumes its worst-case value while the repositioning flight assumes its nominal value and the other $\gamma - 1$ worst-case realizations happen on previous requests (4.2.41); (c) consider that the travel times for service $r \in L$ assumes a nominal value while the repositioning flight assumes its worst-case value and the other $\gamma - 1$ worst-case realizations happen on previous requests (4.2.42); or (d) both flights assume their worst-case value and the other $\gamma - 2$ worst-case realizations happen on previous requests (4.2.43). The remaining constraints in the model follow a similar rationale and correspond to robust counterparts of the time constraints in the MTZ-based model of the deterministic case.

4.2.2.2 Commodity Flow RO formulation

In this topic, we develop the robust counterpart of model (4.2.18)-(4.2.32) by adopting the same strategy as in Section 3.2. In this formulation, we add the index γ to the variable g_{rs}, thus resulting in the new robust variable $g_{rs\gamma}$ that represents the elapsed time for the vehicle that services requests r and s consecutively, considering that the travel times of γ traversed arcs simultaneously attain their worst-case value. Changing the parameters the same way as in previous section, we have the following CF model:

$$\min \sum_{v \in V} \sum_{r \in R} \sum_{s \in R} C_{vrs} y_{vrs} + \sum_{r \in L} \sum_{\substack{s \in R \\ s \neq r}} \sum_{\substack{p \in P \\ p > p_r}} \sum_{v \in V_p} (c_{t_v} \bar{T}^{t_v}_{i_r j_r} - c_{p_r} \bar{T}^{p_r}_{i_r j_r}) y_{vrs} + \sum_{r \in L} Cout_r out_r \quad (4.2.57)$$

$$\text{s.t.} \sum_{\substack{p \in P \\ p \geq p_r}} \sum_{v \in V_p} \sum_{\substack{s \in R \\ s \neq r}} y_{vrs} + out_r = 1, \qquad r \in L, \quad (4.2.58)$$

$$\sum_{\substack{s \in R \\ s \neq r}} y_{vrs} = 1, \qquad r \in M, v \in V, \quad (4.2.59)$$

$$\sum_{\substack{s \in R \\ s \neq r}} y_{vrs} = \sum_{\substack{s \in R \\ s \neq r}} y_{vsr}, \qquad v \in V, r \in R, r > 0, \quad (4.2.60)$$

$$\sum_{s \in R} y_{v0s} = 1 = \sum_{r \in R} y_{vr0}, \qquad v \in V, \quad (4.2.61)$$

$$\sum_{v\in V} ST_r y_{vrs} \leq g_{rs\gamma} \leq \sum_{v\in V}(ST_r + \Delta_L)y_{vrs}, \qquad r \in R, s \in L, \gamma \leq \Gamma, \tag{4.2.62}$$

$$\sum_{v\in V}(ST_r - \Delta_M)y_{vrs} \leq g_{rs\gamma} \leq \sum_{v\in V}(ST_r + \Delta_M)y_{vrs}, \qquad r \in R, s \in M, \gamma \leq \Gamma, \tag{4.2.63}$$

$$\sum_{\substack{s\in R, s>0\\ s\neq r, j_r\neq i_s}} (g_{rs\gamma} - \sum_{v\in V}(\bar{T}^{p_r}_{i_r j_r} + TAT^r_{j_r} + \bar{T}^{p_s}_{j_r i_s} + TAT^s_{i_s})y_{vrs}) +$$

$$\sum_{\substack{s\in R, s>0\\ s\neq r, j_r = i_s}} (g_{rs\gamma} - \sum_{v\in V}(\bar{T}^{p_r}_{i_r j_r} + TAT^s_{i_s})y_{vrs}) \geq \sum_{\substack{h\in R\\ h\neq r}} g_{hr\gamma}, \qquad v \in V, r \in L, \gamma \leq \Gamma, \tag{4.2.64}$$

$$\sum_{\substack{s\in R, s>0\\ s\neq r, j_r\neq i_s}} (g_{rs\gamma} - \sum_{v\in V}(\bar{T}^{p_r}_{i_r j_r} + \hat{T}^{p_r}_{i_r j_r} + TAT^r_{j_r} + \bar{T}^{p_s}_{j_r i_s} + TAT^s_{i_s})y_{vrs}) +,$$

$$\sum_{\substack{s\in R, s>0\\ s\neq r, j_r = i_s}} (g_{rs\gamma} - \sum_{v\in V}(\bar{T}^{p_r}_{i_r j_r} + \hat{T}^{p_r}_{i_r j_r} + TAT^s_{i_s})y_{vrs}) \geq \sum_{\substack{h\in R\\ h\neq r}} g_{hr(\gamma-1)},$$

$$v \in V, r \in L, 0 < \gamma \leq \Gamma, \tag{4.2.65}$$

$$\sum_{\substack{s\in R, s>0\\ s\neq r, j_r\neq i_s}} (g_{rs\gamma} - \sum_{v\in V}(\bar{T}^{p_r}_{i_r j_r} + TAT^r_{j_r} + \bar{T}^{p_s}_{j_r i_s} + \hat{T}^{p_s}_{j_r i_s} + TAT^s_{i_s})y_{vrs}) \geq \sum_{\substack{h\in R\\ h\neq r}} g_{hr(\gamma-1)},$$

$$r \in L, r > 0, 0 < \gamma \leq \Gamma, \tag{4.2.66}$$

$$\sum_{\substack{s\in R, s>0\\ s\neq r, j_r\neq i_s}} (g_{rs\gamma} - \sum_{v\in V}(\bar{T}^{p_r}_{i_r j_r} + \hat{T}^{p_r}_{i_r j_r} + TAT^r_{j_r} + \bar{T}^{p_s}_{j_r i_s} + \hat{T}^{p_s}_{j_r i_s} + TAT^s_{i_s})y_{vrs}) \geq \sum_{\substack{h\in R\\ h\neq r}} g_{hr(\gamma-2)},$$

$$r \in L, r > 0, 1 < \gamma \leq \Gamma, \tag{4.2.67}$$

$$g_{0s\gamma} - \sum_{v\in V}(AV_v + \bar{T}^{t_v}_{k_v i_s} + TAT^s_{i_s})y_{v0s} \geq 0, \qquad s \in L, k_v \neq i_s, \gamma \leq \Gamma, \tag{4.2.68}$$

$$g_{0s\gamma} - \sum_{v\in V}(AV_v + \bar{T}^{t_v}_{k_v i_s} + \hat{T}^{t_v}_{k_v i_s} + TAT^s_{i_s} y_{v0s}) \geq 0,$$

$$s \in L, k_v \neq i_s, 0 < \gamma \leq \Gamma, \tag{4.2.69}$$

$$g_{0s\gamma} - \sum_{v\in V}(AV_v + TAT^s_{i_s})y_{v0s} \geq 0, \qquad s \in L, v \in V, k_v = i_s, \gamma \leq \Gamma, \tag{4.2.70}$$

$$\sum_{\substack{s\in R, s>0\\ s\neq r, j_r\neq i_s}} (g_{rs\gamma} - \sum_{v\in V}(TL_r + \bar{T}^{p_r}_{j_r i_s} + TAT^s_{i_s})y_{v_r rs}) +,$$

$$\sum_{\substack{s\in R, s>0\\ s\neq r, j_r = i_s}} (g_{rs\gamma} - \sum_{v\in V}(TL_r + TAT^s_{i_s})y_{v_r rs}) \geq \sum_{\substack{h\in R\\ h\neq r}} g_{hr\gamma}, \qquad r \in M, \gamma \leq \Gamma, \tag{4.2.71}$$

$$\sum_{\substack{s\in R, s>0\\ s\neq r, j_r\neq i_s}} (g_{rs\gamma} - \sum_{v\in V}(TL_r + \bar{T}^{p_r}_{j_r i_s} + \hat{T}^{p_r}_{j_r i_s} + TAT^s_{i_s})y_{v_r rs}) \geq \sum_{\substack{h\in R\\ h\neq r}} g_{hr(\gamma-1)},$$

$$r \in M, 0 < \gamma \leq \Gamma, \tag{4.2.72}$$

$$\sum_{\substack{s\in R, s>0\\ s\neq r, j_r = i_s}} (g_{rs\gamma} - \sum_{v\in V} TAT^s_{i_s} y_{v_r rs}) \geq 0, \qquad r \in M, \gamma \leq \Gamma, \tag{4.2.73}$$

$$g_{0s\gamma} - \sum_{v\in V}(AV_{v_s} + \bar{T}^{p_s}_{k_{v_s} i_s})y_{v_s 0s} \geq 0, \qquad s \in M, k_{v_s} \neq i_s, \gamma \leq \Gamma, \tag{4.2.74}$$

$$g_{0s\gamma} - \sum_{v\in V}(AV_{v_s} + \bar{T}^{p_s}_{k_{v_s} i_s} + \hat{T}^{p_s}_{k_{v_s} i_s})y_{v_s 0s} \geq 0, \qquad s \in M, k_{v_s} \neq i_s, 0 < \gamma \leq \Gamma, \tag{4.2.75}$$

$$g_{0s\gamma} - \sum_{v\in V} AV_{v_s} y_{v_s 0s} \geq 0, \qquad s \in M, k_{v_s} = i_s, \gamma \leq \Gamma, \tag{4.2.76}$$

$$g_{rs\gamma} \geq 0, \qquad r, s \in R, \gamma \leq \Gamma, \tag{4.2.77}$$

$$y_{vrs} \in \{0,1\}, out_r \in \{0,1\}, \qquad v \in V, r, s \in R. \tag{4.2.78}$$

The objective function (4.2.57) and constraints (4.2.58)-(4.2.61) are exactly the same as in the RO formulation with MTZ constraints presented in the previous section (4.2.33)-(4.2.37). The time windows for costumer and maintenance requests are enforced by constraints (4.2.62)-(4.2.63), respectively. Robust time flow constraints for each type of request, found in constraints (4.2.64)-(4.2.76) works exactly the same for their equivalent in (4.2.40)-(4.2.54). The main difference is that, similarly to the deterministic formulation, constraints (4.2.64), (4.2.65) and (4.2.71) encompass a pair of set of constrains from MTZ, one for each sum in those constraints, the first one of these sum in each constraint refers to the arcs where positioning is needed and the second sum represent the case where the this type of flight is not necessary. For example, constraints (4.2.64) are equivalent to (4.2.40) and (4.2.44) in the MTZ model. Note that some time flow constraints, namely (4.2.66), (4.2.67) and (4.2.72) have a single sum term, and not two as the other time flow constraints. This happens because these constraints, which represent either delay in the positioning (4.2.66) and (4.2.72) or in both the service and positioning flights (4.2.67), require a positioning to exist, thus we only need to consider the arcs in which $j_r \neq i_s$ for these constraints. Finally, constraints (4.2.77) and (4.2.78) set the domain of variables.

4.2.3 RO formulations based on the single knapsack uncertainty set

4.2.3.1 MTZ-based RO formulation

To extend model (4.2.1)-(4.2.17) to incorporate this uncertainty set, we use the same strategy we used for the model in Section 3.3. We add an index δ that represents the total deviation already considered in a route into the time load variable w_r. The new variable, $w_{r\delta}$, represents the starting time of request $r \in R$ considering that a total deviation $\delta \leq \Delta$ already occurred in the route, Δ being the maximum deviation allowed. The travel time parameter (T_{ij}^p) is represented again by its nominal value $(\bar{T}_{ij}^p)$ and maximum deviation $(\hat{T}_{ij}^p)$. With no more noticeable changes, the full robust model is as follows:

$$\min \sum_{v\in V}\sum_{r\in R}\sum_{s\in R} C_{vrs}y_{vrs} + \sum_{r\in L}\sum_{\substack{s\in R\\ s\neq r}}\sum_{\substack{p\in P\\ p>p_r}}\sum_{v\in V_p}(c_{t_v}\bar{T}_{i_rj_r}^{t_v} - c_{p_r}\bar{T}_{i_rj_r}^{p_r})y_{vrs} + \sum_{r\in L} Cout_r out_r \tag{4.2.79}$$

$$\text{s.t.} \sum_{\substack{p\in P\\ p\geq p_r}}\sum_{v\in V_p}\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} + out_r = 1, \qquad r \in L, \tag{4.2.80}$$

$$\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} = 1, \qquad r \in M, v \in V, \tag{4.2.81}$$

$$\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} = \sum_{\substack{s\in R\\ s\neq r}} y_{vsr}, \qquad v \in V, r \in R, r > 0, \tag{4.2.82}$$

$$\sum_{s\in R} y_{v0s} = 1 = \sum_{r\in R} y_{vr0}, \qquad v \in V, \tag{4.2.83}$$

$$ST_r \leq w_{r\delta} \leq ST_r + \Delta_L, \qquad r \in L, \delta \leq \Delta, \tag{4.2.84}$$

$$ST_r - \Delta_M \leq w_{r\delta} \leq ST_r + \Delta_M, \qquad r \in M, \delta \leq \Delta, \tag{4.2.85}$$

$$w_{s\delta} \geq w_{r\delta} + \sum_{v\in V}(\bar{T}_{i_rj_r}^{p_r} + TAT_{j_r}^r + \bar{T}_{j_ri_s}^{p_s} + TAT_{i_s}^s)y_{vrs} + M_{rs}^1(\sum_{v\in V} y_{vrs} - 1),$$

$$r \in L, s \in R, r \neq s, s > 0, j_r \neq i_s, \delta \leq \Delta, \tag{4.2.86}$$

$$w_{s\delta} \geq w_{r(\delta-\lambda)} + \sum_{v \in V} (\bar{T}^{p_r}_{i_r j_r} + TAT^r_{j_r} + \bar{T}^{p_s}_{j_r i_s} + TAT^s_{i_s} + \lambda) y_{vrs} + M^1_{rs}(\sum_{v \in V} y_{vrs}),$$

$$r \in L, s \in R, r \neq s, s > 0, j_r \neq i_s, 0 < \lambda \leq \delta \leq \Delta, \lambda \leq (\hat{T}^{p_r}_{i_r j_r} + \hat{T}^{p_s}_{j_r i_s}), \tag{4.2.87}$$

$$w_{s\delta} \geq w_{r\delta} + \sum_{v \in V} (\bar{T}^{p_r}_{i_r j_r} + TAT^s_{js}) y_{vrs} + M^2_{rs}(\sum_{v \in V} y_{vrs} - 1),$$

$$r \in L, s \in R, r \neq s, s > 0, j_r = i_s, \delta \leq \Delta, \tag{4.2.88}$$

$$w_{s\delta} \geq w_{r(\delta-\lambda)} + \sum_{v \in V} (\bar{T}^{p_r}_{i_r j_r} + TAT^s_{js} + \lambda) y_{vrs} + M^2_{rs}(\sum_{v \in V} y_{vrs} - 1),$$

$$r \in L, s \in R, r \neq s, s > 0, j_r = i_s, 0 < \lambda \leq \delta \leq \Delta, \lambda \leq \hat{T}^{p_r}_{i_r j_r}, \tag{4.2.89}$$

$$w_{s\delta} \geq (AV_v + \bar{T}^{t_v}_{k_v i_s} + TAT^s_{i_s}) y_{v0s}, \qquad s \in L, v \in V, k_v \neq i_s, \delta \leq \Delta, \tag{4.2.90}$$

$$w_{s\delta} \geq (AV_v + \bar{T}^{t_v}_{k_v i_s} + TAT^s_{i_s} + \lambda) y_{v0s},$$

$$s \in L, v \in V, k_v \neq i_s, 0 < \lambda \leq \delta \leq \Delta, \lambda \leq \hat{T}^{t_v}_{k_v i_s}, \tag{4.2.91}$$

$$w_{s\delta} \geq (AV_v + TAT^s_{i_s}) y_{v0s}, \qquad s \in L, v \in V, k_v = i_s, \delta \leq \Delta, \tag{4.2.92}$$

$$w_{s\delta} \geq w_{r\delta} + TL_r + \bar{T}^{p_r}_{j_r i_s} + TAT^s_{i_s} + M^3_{rs}(y_{v_r rs} - 1),$$

$$r \in M, s \in R, r \neq s, s > 0, j_r \neq i_s, \delta \leq \Delta, \tag{4.2.93}$$

$$w_{s\delta} \geq w_{r(\delta-\lambda)} + TL_r + \bar{T}^{p_r}_{j_r i_s} + TAT^s_{i_s} + \lambda + M^3_{rs}(y_{v_r rs} - 1),$$

$$r \in M, s \in R, r \neq s, s > 0, j_r \neq i_s, 0 < \lambda \leq \delta \leq \Delta, \lambda \leq \hat{T}^{p_r}_{i_r j_r}, \tag{4.2.94}$$

$$w_{s\delta} \geq w_{r\delta} + TAT^s_{i_s} + M^4_{rs}(y_{v_r rs} - 1),$$

$$r \in M, s \in R, r \neq s, s > 0, j_r = i_s, \delta \leq \Delta, \tag{4.2.95}$$

$$w_{s\delta} \geq (AV_{v_s} + \bar{T}^{p_s}_{k_{v_s} i_s}) y_{v_s 0s}, \qquad s \in M, k_{v_s} \neq i_s, \delta \leq \Delta, \tag{4.2.96}$$

$$w_{s\delta} \geq (AV_{v_s} + \bar{T}^{p_s}_{k_{v_s} i_s} + \lambda) y_{v_s 0s}, \qquad s \in M, k_{v_s} \neq i_s, 0 < \lambda \leq \delta \leq \Delta, \lambda \leq \hat{T}^{p_s}_{k_{v_s} i_s}, \tag{4.2.97}$$

$$w_{s\delta} \geq AV_{v_s} y_{v_s 0s}, \qquad s \in M, k_{v_s} = i_s, \delta \leq \Delta, \tag{4.2.98}$$

$$w_{r\delta} \geq 0, \qquad r \in R, \delta \leq \Delta. \tag{4.2.99}$$

$$y_{vrs} \in \{0, 1\}, out_r \in \{0, 1\}, \qquad v \in V, r, s \in R. \tag{4.2.100}$$

Like the previous models, the objective function (4.2.79) and the first four sets of constraints, the ones ensuring that every request is serviced and vehicle flow (4.2.80-4.2.83), are identical to the ones in the original formulation (4.2.1-4.2.5). Likewise, the time windows constraints, (4.2.84) and (4.2.85), are similar to (4.2.6) and (4.2.7), respectively, the only difference between the ones in the robust model from the original model is the addition of index δ into the time variable.

Constraints (4.2.86) and (4.2.87) compute the earliest starting time of request $s \in R$ if it is serviced immediately after a customer request $r \in L$, and a positioning flight between requests is needed. The starting time of a given request $s \in R$ is the latest according to two cases: considering a deviation of δ time units during the route but not in the previous request (4.2.86), or by adding $\lambda \leq \delta$ deviation into the request r and consider a deviation of only $\delta - \lambda$ in the route previous to this request (4.2.87). The value of λ is also limited by the deviation on the flights related to request r i.e., the sum of deviations from the repositioning flight, if needed, and the flight boarded by the customer. Constraints (4.2.88) and (4.2.89) are similar to (4.2.86) and (4.2.87), but for the case in which repositioning flight is not required. Constraints

(4.2.90) and (4.2.91) are the robust extension of (4.2.10), and refers to the first request attended by the aircraft during the planning horizon. (4.2.92) are essentially the same constraints as (4.2.11), and refer to the first request of planing horizon of each aircraft when repositioning is not needed. Constraints (4.2.93)-(4.2.98), are analogous to(4.2.86)-(4.2.92) but for the case in which the previous attended request r was a maintenance one. Similarly to (4.2.8)-(4.2.15), constraints (4.2.86)-(4.2.98) act twofold, not only they compute starting times of each request, they also prevent the formation of subtours. Finally, constraints (4.2.99) and (4.2.100) determine the domain of the variables.

4.2.3.2 Commodity Flow RO formulation

The same approach was used to develop a robust model based on the CF formulation for the company's case considering the knapsack uncertainty set. Similarly to the CF formulation proposed for the cardinality constrained uncertainty set, the main differences from this robust formulation to the deterministic model (4.2.18)-(4.2.32) is the addition of the index δ, that represents the accumulate deviation in the route, into the variable g_{rs} and the split of the travel time parameter into its nominal and maximum deviation value. The new variable $g_{rs\delta}$ represents the total time load carried between requests r and s considering a total deviation of $\delta \leq \Delta$ already computed in the route. The parameter Δ is the budget of total deviation allowed in a route.

$$\min \sum_{v\in V}\sum_{r\in R}\sum_{s\in R} C_{vrs}y_{vrs} + \sum_{r\in L}\sum_{\substack{s\in R\\ s\neq r}}\sum_{\substack{p\in P\\ p>p_r}}\sum_{v\in V_p}(c_{t_v}T^{t_v}_{i_rj_r} - c_{p_r}T^{p_r}_{i_rj_r})y_{vrs} \\ + \sum_{r\in L} Cout_r out_r \tag{4.2.101}$$

$$\text{s.t.} \sum_{\substack{p\in P\\ p\geq p_r}}\sum_{v\in V_p}\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} + out_r = 1, \qquad r\in L, \tag{4.2.102}$$

$$\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} = 1, \qquad r\in M, v\in V, \tag{4.2.103}$$

$$\sum_{\substack{s\in R\\ s\neq r}} y_{vrs} = \sum_{\substack{s\in R\\ s\neq r}} y_{vsr}, \qquad v\in V, r\in R, r>0, \tag{4.2.104}$$

$$\sum_{s\in R} y_{v0s} = 1 = \sum_{r\in R} y_{vr0}, \qquad v\in V, \tag{4.2.105}$$

$$\sum_{v\in V} ST_r y_{vrs} \leq g_{rs\delta} \leq \sum_{v\in V}(ST_r + \Delta_L)y_{vrs}, \qquad r\in R, s\in L, \delta\leq\Delta, \tag{4.2.106}$$

$$\sum_{v\in V}(ST_r - \Delta_M)y_{vrs} \leq g_{rs\delta} \leq \sum_{v\in V}(ST_r + \Delta_M)y_{vrs}, \; r\in R, s\in M, \delta\leq\Delta, \tag{4.2.107}$$

$$\sum_{\substack{s\in R, s>0\\ s\neq r, j_r\neq i_s}} (g_{rs\delta} - (\bar{T}^{p_r}_{i_rj_r} + TAT^r_{j_r} + \bar{T}^{p_s}_{j_ri_s} + TAT^s_{i_s})y_{vrs}) + \\ \sum_{\substack{s\in R, s>0\\ s\neq r, j_r=i_s}} (g_{rs\delta} - (\bar{T}^{p_r}_{i_rj_r} + TAT^s_{i_s})y_{vrs}) \geq \sum_{\substack{h\in R\\ h\neq r}} g_{hr\delta}, \qquad v\in V, r\in L, \delta\leq\Delta, \tag{4.2.108}$$

$$\sum_{\substack{\lambda\leq(\hat{T}^{p_r}_{i_rj_r} + \hat{T}^{p_s}_{j_ri_s})\\ s\in R, s>0\\ s\neq r, j_r\neq i_s}} (g_{rs\delta} - (\bar{T}^{p_r}_{i_rj_r} + TAT^r_{j_r} + \bar{T}^{p_s}_{j_ri_s} + TAT^s_{i_s} + \lambda)y_{vrs}) +$$

$$\sum_{\substack{\lambda\le\hat{T}^{p_r}_{i_rj_r}\\ s\in R,s>0\\ s\neq r,j_r=i_s}} (g_{rs\delta} - (\bar{T}^{p_r}_{i_rj_r} + TAT^s_{i_s} + \lambda)y_{vrs}) \geq \sum_{\substack{h\in R\\ h\neq r}} g_{hr(\delta-\lambda)},$$

$$v \in V, r \in L, 0 < \lambda \le \delta \le \Delta, \tag{4.2.109}$$

$$g_{v0s\delta} - (AV_v + \bar{T}^{t_v}_{k_vi_s} + TAT^s_{i_s})y_{v0s} \geq 0,$$

$$v \in V, s \in L, k_v \neq i_s, 0 < \delta \le \Delta, \tag{4.2.110}$$

$$g_{v0s\delta} - (AV_v + \bar{T}^{t_v}_{k_vi_s} + TAT^s_{i_s} + \lambda)y_{v0s} \geq 0,$$

$$v \in V, s \in L, k_v \neq i_s, 0 < \lambda \le \delta \le \Delta, \lambda \le \hat{T}^{t_v}_{k_vi_s}, \tag{4.2.111}$$

$$g_{v0s\delta} - (AV_v + TAT^s_{i_s})y_{v0s} \geq 0,$$

$$v \in V, s \in L, k_v = i_s, \delta \le \Delta, \tag{4.2.112}$$

$$\begin{aligned}&\sum_{\substack{s\in R,s>0\\ s\neq r,j_r\neq i_s}} (g_{rs\delta} - TL_r + \bar{T}^{p_r}_{j_ri_s} + TAT^s_{i_s})y_{vrs} + \\ &\qquad \sum_{\substack{s\in R,s>0\\ s\neq r,j_r=i_s}} (g_{rs\delta} - TL_r - TAT^s_{i_s})y_{v_rrs} \geq \sum_{\substack{h\in R\\ h\neq r}} g_{hr\delta}, \qquad r \in M, \delta \le \Delta,\end{aligned} \tag{4.2.113}$$

$$\sum_{\substack{\lambda\le\hat{T}^{p_r}_{j_ri_s}\\ s\in R,s>0\\ s\neq r,j_r\neq i_s}} (g_{rs\delta} - TL_r + \bar{T}^{p_r}_{j_ri_s} + TAT^s_{i_s} + \lambda)y_{v_rrs} \geq \sum_{\substack{h\in R\\ h\neq r}} g_{hr(\delta-\lambda)},$$

$$r \in M, 0 < \lambda \le \delta \le \Delta, \tag{4.2.114}$$

$$g_{v0s\delta} - (AV_{v_s} + \bar{T}^{p_s}_{k_{v_s}i_s})y_{v_s0s} \geq 0, \qquad s \in M, k_{v_s} \neq i_s, \delta \le \Delta, \tag{4.2.115}$$

$$g_{v0s\delta} - (AV_{v_s} + \bar{T}^{p_s}_{k_{v_s}i_s} + \lambda)y_{v_s0s} \geq 0,$$

$$s \in M, k_{v_s} \neq i_s, 0 < \lambda \le \delta \le \Delta, \lambda \le \hat{T}^{p_s}_{k_{v_s}i_s}, \tag{4.2.116}$$

$$g_{v0s\delta} - y_{v_s0s}AV_{v_s} \geq 0, \qquad s \in M, k_{v_s} = i_s, \delta \le \Delta, \tag{4.2.117}$$

$$g_{rs\delta} \geq 0, \qquad v \in V, r, s \in R, \lambda \le \Delta, \tag{4.2.118}$$

$$y_{vrs} \in \{0, 1\}, out_r \in \{0, 1\}, \qquad v \in V, r, s \in R. \tag{4.2.119}$$

This formulation is equivalent to the MTZ one for the same uncertainty set, with each constraint acting similarly to their respective counterpart from the MTZ formulation. The only difference between then is how the constraints are structured. The objective function of minimizing the positioning, upgrade and outsourcing costs (4.2.101), is identical to the deterministic model. The vehicle flow constraints (4.2.102)-(4.2.105), also have no changes from the deterministic formulation. The time windows constraints, (4.2.102) and (4.2.105), are also similar to the deterministic ones, and the only notable difference is the addition of index δ into the variable. Time flow constraints for every type of request are found in (4.2.108)-(4.2.117). Constraints (4.2.108)-(4.2.109) compute the earliest arrival time possible in request s, considering a total deviation of δ, when the previous request serviced by aircraft v is a customer request. Constraints (4.2.108) are used when no deviation is considered in the arc that arrives in s, while constraints (4.2.109) consider λ units in this deviation. In both cases there are two sums, the first encompasses the requests in which positioning is needed, whereas the second represents the case in which the destination airport from the previous request is the same as the departure airport of request s. Note that the index λ found in constraints (4.2.109) represent the total deviation considered in arc (r, s), thus its value is limited not only by δ but also the maximum deviation in that arc, which is represented by $(\hat{T}^{p_r}_{i_r j_r} + \hat{T}^{p_r}_{j_r i_s})$ if there is a positioning between nodes r and s, or simply $(\hat{T}^{p_r}_{i_r j_r})$ if no positioning is required. Constraints (4.2.110)-(4.2.112) are used for the first request in the planning horizon of each aircraft, if it is a customer request. The first two constraints are activated when repositioning flight is needed and the last one is when it is not. Constraints (4.2.113)-(4.2.117) are similar to (4.2.108)-(4.2.112), but for maintenance requests. Lastly, the domain of decisions variables are set in constraints (4.2.118) and (4.2.119).

4.3 Branch-and-cut algorithm

To incorporate crew assignment to the models described in the previous section, we develop a B&C algorithm that dynamically introduces cuts into the proposed models when a candidate solution is infeasible regarding crew rules. Additionally, to consider the possibility and costs of allowing the crew members to work overtime, we adapt the proposed models by introducing a new parameter cov_r, which represents an estimate unit cost for a crew to work overtime on request r, and the positive continuous decision variable $over_r$, which represent the total overtime in request r in the solution. With these additional information, we update the objective function of all models to also consider the overtime costs in the problem, resulting in following objective function:

$$\min \sum_{v\in V}\sum_{r\in R}\sum_{s\in R} C_{vrs}y_{vrs} + \sum_{r\in L}\sum_{\substack{s\in R\\ s\neq r}}\sum_{\substack{p\in P\\ p>p_r}}\sum_{v\in V_p}(c_{t_v}T^{t_v}_{i_r j_r} - c_{p_r}T^{p_r}_{i_r j_r})y_{vrs} + \sum_{r\in L} Cout_r out_r + \sum_{r\in R} cov_r over_r.$$

This objective function minimizes the total operational cost with the last term representing the costs related to overtime. Notice that we do not directly introduce any new constraint related to overtime into the models, as the overtime is only related to the crew's requirements and thus will only be changed by the cuts generated by the algorithm. In the remainder of this section, we present the separation algorithm developed for the problem, first for the deterministic case (Subsection 4.3.1) and then its robust counterpart (Subsection 4.3.2).

4.3.1 Separation algorithm for the deterministic problem

The proposed separation algorithm consists of a labelling strategy (Feillet, 2010) based on dynamic programming to verify whether a solution is feasible in the scope of crew requirements, since it is impossible to define when the crew should rest in a route when analyzing the route only up to a specific node, as this decision impacts following nodes. For example, resting before the nth node in a route might not violate its time windows, but this extra time might make the time window of the $(n+1)$th node infeasible. Thus, one should account for the different resting options and possible future impacts in a route. For this purpose, we developed a labeling algorithm. Firstly, in a given solution, let $\mathcal{R}_v = (r_0, r_1, r_2, \ldots, r_{n+1})$ be a route of aircraft v, where r_0 and r_{n+1} are nodes that represent the initial and final artificial requests. In the labeling framework, we assign a *bucket* to each node and, inside each bucket, there are different labels related to this node. Labels are data structures responsible to carry a series of essential information through the buckets for calculating accumulated resources and checking the feasibility condition.

Regarding the crew requirements, we consider four resources: the total elapsed time $(Elap_l)$; the total flying time (Fly_l); the total duty time (DT_l) since the last rest; and the total overtime cost (OT_l) accumulated up to label l. In addition, each label keeps track of the information of the parent (previous) label, useful for backtracking in post-processing. This procedure is summarized in Algorithm 1.

Algorithm 1: Separation algorithm

```
   Input: Solution candidate.
   Output: Feasibility solution for the crew assignments.
 1 foreach aircraft $v \in V$ do
 2     Get $\mathcal{R}_v$;
 3     Create a bucket for each node in $\mathcal{R}_v$;
 4     Create label $l$ in the bucket of node $r_0$ with the initial characteristics of the aircraft;
 5     for $i \leftarrow 0$ to $n$, step $+1$ do
 6         foreach label $l \in$ bucket of node $r_i$ do
 7             Extend the current label $l$ to all six possible cases and create child labels for
                 those that are feasible in $r_{i+1}$;
 8         end
 9         Check whether the created child labels dominate or are dominated by other
             existing labels in the bucket of node $r_i$ or by themselves;
10         Delete the dominated labels;
11         if there is no label in the current node then
12             Infeasible route: Add no-good cuts to eliminate solution;
13             break;
14         end
15     end
16 end
17 if cuts were not inserted in the model then
18     return Feasible solution for the crew requirements;
19 end
20 else
21     return Flag the solution candidate as infeasible for the crew requirements;
22 end
```

The first operation in Algorithm 1, after getting the route R_v and creating its respective *buckets,* is defining a label at the bucket of node r_0 with the initial characteristics of the aircraft, such as the time it is available and the accumulated duty time since the last rest at the beginning of the planning horizon. Accumulated duty values are non-zero when the company's planning starts in the middle of a duty of a specific aircraft. After creating the label for node r_0, the algorithm extends it to the bucket of the first request in the route, r_1. Likewise, each generated label will be extended to the next nodes of the route, or at least an attempt will be made. This process is repeated until the last node in the route is reached or there is no way to create any label for the next node in the route, in which case the solution is deemed infeasible. As pointed out in line 5 of the algorithm, when extending a label, there are six different possible cases related to rest decisions and whenever a verified case is feasible, a child label is generated using its strategy. Thus, each label can have up to six children. The options related to rest between requests that should be checked are represented in Figure 13. In this figure, a request that can be of any type is represented by a blue box. The time windows for the start time of the second request is represented with a black outline besides or inside it. If a case requires specifically a maintenance event, it is represented by a green box. Additionally, before the last request of each case, there is a positioning flight (ferry), represented by a red box. Finally, resting and presentation times, are represented by, respectively, yellow and brown/orange (depending if it is at the beginning or end of the duty) boxes.

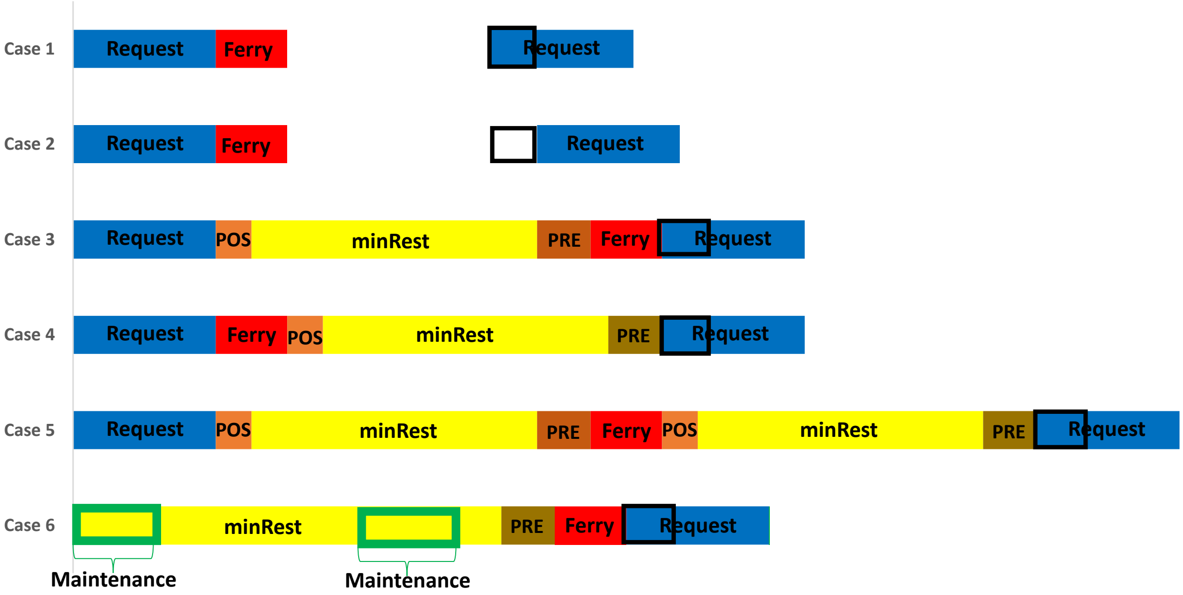


Figure 13 – Visual representation of all cases checked when extending a label to the next node.

The first case is the option of executing request r_i and immediately prepare to start request r_{i+1}, without any rest. In case 2, the crew delays the start of the service as much as possible in order to take advantage of the split duty rule. Particularly, we try to extend the time the crew stays stationary to make the total break duration fall under the third category of the split duty rule (over 6 hours) so only 60 minutes are added in the accumulated duty time for the whole break. Case 3 occurs when the crew takes a rest between requests r_i and r_{i+1} at the departure airport of request r_{i+1}. In this case the aircraft will start request r_{i+1} immediately after ending the rest. If a positioning is required, Case 3 can be interpreted as taking a rest after the positioning flight. Conversely, if there is no positioning the rest starts immediately after the end of request r_i, if it is a customer request, or at the moment the aircraft arrived at the designated airport, if it is a maintenance event. Case 4 only exists if a positioning between requests r_i and r_{i+1} is required and represents the possibility to rest before the positioning. This option is usually taken when positioning would exceed the duty limit. Case 5 is a combination of cases 3 and 4, and represents the attempt to take a complete rest before and after positioning, this one is especially useful on particularly long positioning trips. Finally, case 6 can be chosen in a particular situation where two consecutive maintenance events on an aircraft happen at the same airport. This allows the second maintenance to start while the crew is still resting, instead of forcing the crew to return to work to start the second maintenance like in case 2. Note that the label generated in case 6 is not inserted in the next node of the route (r_{i+1}), but in the following one (r_{i+2}).

When a label is created, the algorithm verifies if it dominates or is dominated by any other existing label in the evaluated node. A label l_1 dominates a label l_2 in the same node or bucket if and only if:

- $Elap_{l_1} \leq Elap_{l_2}$;
- $DT_{l_1} \leq DT_{l_2}$;
- $Fly_{l_1} \leq Fly_{l_2}$;
- $OT_{l_1} \leq OT_{l_2}$;

In other words, label l_1 dominates l_2 if, and only if, it has lower or equal elapsed time, less accumulated duty, flying time and accumulated overtime costs at the start of the evaluated node. This can happen, for example, when it is possible to take a rest between two requests and the total elapsed time is still lower than the opening time windows, and thus case 1 will be dominated by the ones that allow resting. Another situation in which this usually happens is when we compare cases 3 and 4. The former usually dominates the latter, since its accumulated duty time at the start of the next request is lower, except in the situation case 3 is infeasible.

A label l in the bucket representing request r_i is considered infeasible, and thus not allowed to be inserted, only if it disrespect the time windows of request r_i ($Elap_l > ST_{r_i} + \Delta$). Exceeding the maximum duty and flying times, on the other hand, is allowed and only results in an increase in the accumulated overtime cost. Let l_p be the parent node of request l, then one can generally compute OT_l using the following expression:

$$OT_l = OT_{l_p} + \max\{0, Fly_l - MaxFly, DT_l - MaxDuty\}.$$

However, if the evaluated case uses a rest (Cases 3-6), it is also important to check if the crew needs to work overtime before the rest start of the rest, as the total duty and flying times are reset after the rest and this information would be lost. For example, let r_p be the request associated to the bucket where l_p belongs, then the total overtime for case 3 can be computed with the following expression:

$$OT_l = \begin{cases} OT_{l_p} + \max\{0, DT_l + T^{pv}_{i_{r_p} j_{r_p}} - MaxDuty, Fly_l + T^{pv}_{i_{r_p} j_{r_p}} - MaxFly\} & \\ \qquad + \max\{0, Fly_l - MaxFly, DT_l - MaxDuty\} & \text{if } r_p \in \mathcal{L}, \\ OT_{l_p} + \max\{0, DT_l + TL_{r_p} - MaxDuty, Fly_l + TL_{r_p} - MaxFly\} & \\ \qquad + \max\{0, Fly_l - MaxFly, DT_l - MaxDuty\} & \text{if } r_p \in \mathcal{M}. \end{cases}$$

Figure 14 presents a numerical and visual example on how the B&C algorithm checks the feasibility of a route. In this example, we want to verify if route $\mathcal{R} = (r_0, r_1, r_2, r_3, r_0)$ is feasible. This particular route is composed by the artificial node r_0 (from where the aircraft departs and returns) and three customer requests. The time windows for each request is found just below their respective bucket and the duration of each request is found above it, i.e., 50 minutes for r_1, 150 for r_2 and 180 for r_3. The positioning times are found over the arrow connecting two consecutive requests, thus 50 minutes between r_0 and r_1, 20 between r_1 and r_2 and 90 minutes between r_2 and r_3.

We start the algorithm by generating a label in node r_0. This label has a total elapsed time (E) and accumulated duty time (D) equal to 0. We then extend the generated label with the six possible cases, not forgetting to add the presentation time (PRE) in this first extension. Since the time window from request r_1 closes very early, cases 3 to 5 are all infeasible because if the crew takes a rest the elapsed time will exceed the closing time windows. Furthermore, since there is no maintenance event in this problem, case 6 will never be used. Thus, only two labels remain, one generated by case 1 and other generated by case 2. In this example, extending the start of the service (case 2) at customer r_1 was not worth it, as it resulted in more accumulated duty and elapsed time than simply start the service on time, and thus only the label generated by case 1 remained. This label was then extended to request r_2 and a behavior similar to the previous case was identified. The only notable difference was that there was not a dominance relationship between the labels generated by cases 1 and 2, as one presented lower total elapsed time (case 1) and the other presented lower accumulated duty times (case 2). We now extend both labels to the bucket of r_3. This time, the label generated by case 3 dominated the other ones. This bucket also exemplify why case 2 is relevant, as the child of the label generated by this case was the one that provided the best result. Finally, we can easily extend the generated label to the remaining node (r_0). Thus, we were able to successfully generate labels for all nodes in the route and confirm that this route is feasible.

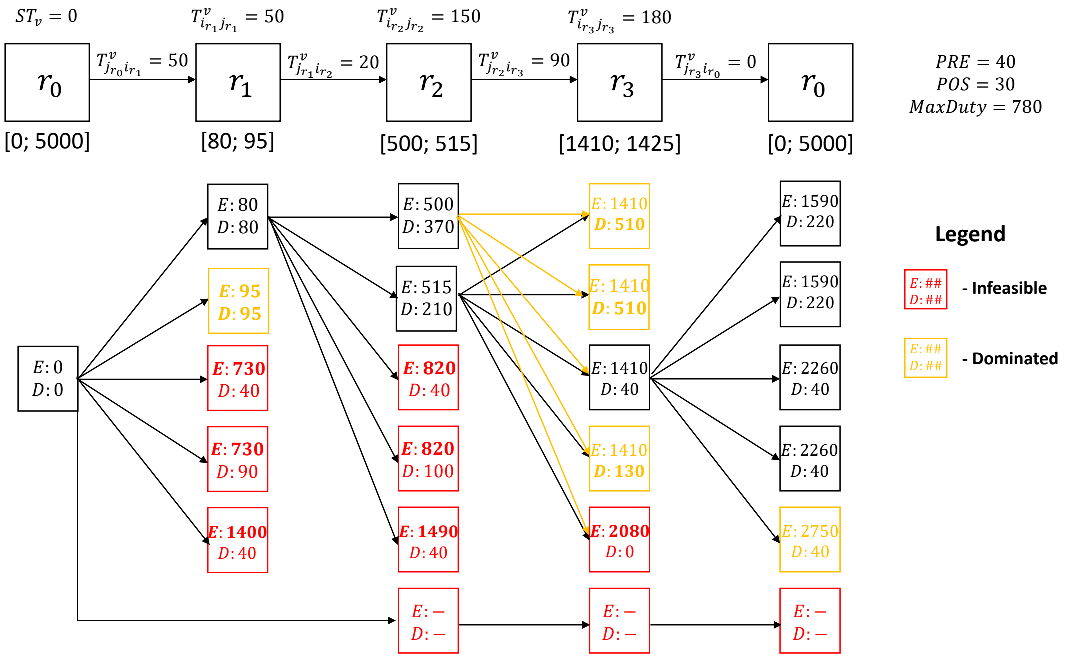


Figure 14 – Visual representation of a labeling structure.

If it is not possible to generate any label in some node, we consider the route infeasible and cut-off the solution. For this purpose, we introduce *feasibility cuts*, in which we cut the route up to the node with no labels, limiting the sum of the binary variables. Suppose route $\mathcal{R}_v = (r_0, r_1, \dots, r_k \dots, r_n)$ is assigned to vehicle v and it becomes infeasible at node r_k. The algorithm would generate the following cut:

$$y_{vr_0r_1} + y_{vr_1r_2} + y_{vr_2r_3} + \cdots + y_{vr_{k-1}r_k} \leq k - 1. \tag{4.3.1}$$

This inequality prevents vehicle v to take this sequence of nodes up to node r_k, and thus cutting not only this route, but any other that would try this combination in v up to node r_k.

If a route is feasible, but an overtime is required, we check if the solution has already set an appropriate value for each variable $over_r$ for each request r_k in route $\mathcal{R}$, i.e., we check if $over_r \geq OT_{r_k} - OT_{r_{k-1}}$. If the solution consider a total overtime in a request r smaller than the value computed by the labelling algorithm, then the following cut is introduced into the problem:

$$(OT_{r_k} - OT_{r_{k-1}})\Big(y_{vr_0r_1} + y_{vr_1r_2} + y_{vr_2r_3} + \cdots + y_{vr_{k-1}r_k} - (k-1)\Big) \leq over_r. \tag{4.3.2}$$

This cut ensures that the overtime costs are properly computed in the objective function if that specific route is taken.

4.3.2 Separation algorithm for the RO problem under time uncertainty

To extend the deterministic separation algorithm previously discussed, we adapt the labelling structure of the problem to consider the uncertainty on travelling times with the studied uncertainty sets. Particularly, instead of using a single value for $Elap_l$ in label l, we use an array of size $\Gamma + 1$ (or $\Delta + 1$ if using the knapsack uncertainty set). Thus, in a problem with the cardinality constrained uncertainty set, the elapsed time is represented by matrix $Elap_{l\gamma}$, which constitutes the total elapsed time in label l when up to $\gamma \in \{0 \ldots \Gamma\}$ parameters attain their worst-case value in the route. Similarly, in a problem with the knapsack uncertainty set we use matrix $Elap_{l\delta}$ to represent the total elapsed time in label l with a total deviation of $\delta = \{0, \ldots, \Delta\}$. The same changes are done to the remaining time variables computed in each label (DT_l, Fly_l and OT_l).

We use arrays to represent the total elapsed time, duty time, flying time and overtime in each label so we can use a dynamic programming strategy inspired in equations (2.3.90), if considering the cardinality constrained uncertainty set, and (3.3.3), for the knapsack uncertainty set, to compute the worst-case value of these parameters when generating a label with one of the six cases. For example, let l_1 be a label children of l_0 generated by case 1 in a problem that considers the cardinality constrained uncertainty set and let the request related to the bucket of label l_0, r_0, be a customer request which requires a positioning flight to the departure of the request in the bucket of label l_1, r_1. Then, the total elapsed time can be computed with the following expression:

$$Elap_{l_1\gamma} = \begin{cases} \max\{a_{r_1}, Elap_{l_0\gamma} + \bar{T}^{p_{r_0}}_{i_0j_0} + TAT^r_{j_0} + \bar{T}^{p_1}_{j_0i_1} + TAT^s_{i_1}\}, & \text{if } \gamma = 0, \\ \max\{a_{r_1}, Elap_{l_0\gamma} + \bar{T}^{p_{r_0}}_{i_0j_0} + TAT^r_{j_0} + \bar{T}^{p_1}_{j_0i_1} + TAT^s_{i_1}, & \\ \qquad Elap_{l_0\gamma-1} + \bar{T}^{p_{r_0}}_{i_0j_0} + \hat{T}^{p_{r_0}}_{i_0j_0} + TAT^r_{j_0} + \bar{T}^{p_1}_{j_0i_1} + TAT^s_{i_1}, & \\ \qquad Elap_{l_0\gamma-1} + \bar{T}^{p_{r_0}}_{i_0j_0} + TAT^r_{j_0} + \bar{T}^{p_1}_{j_0i_1} + \hat{T}^{p_1}_{j_0i_1} + TAT^s_{i_1}\}, & \text{if } \gamma = 1, \\ \max\{a_{r_1}, Elap_{l_0\gamma} + \bar{T}^{p_{r_0}}_{i_0j_0} + TAT^r_{j_0} + \bar{T}^{p_1}_{j_0i_1} + TAT^s_{i_1}, & \\ \qquad Elap_{l_0\gamma-1} + \bar{T}^{p_{r_0}}_{i_0j_0} + \hat{T}^{p_{r_0}}_{i_0j_0} + TAT^r_{j_0} + \bar{T}^{p_1}_{j_0i_1} + TAT^s_{i_1}, & \\ \qquad Elap_{l_0\gamma-1} + \bar{T}^{p_{r_0}}_{i_0j_0} + TAT^r_{j_0} + \bar{T}^{p_1}_{j_0i_1} + \hat{T}^{p_1}_{j_0i_1} + TAT^s_{i_1}, & \\ \qquad Elap_{l_0\gamma-2} + \bar{T}^{p_{r_0}}_{i_0j_0} + \hat{T}^{p_{r_0}}_{i_0j_0} + TAT^r_{j_0} + \bar{T}^{p_1}_{j_0i_1} + \hat{T}^{p_1}_{j_0i_1} + TAT^s_{i_1}\}, & \text{if } \gamma > 1. \end{cases} \tag{4.3.3}$$

Essentially, the algorithm computes the earliest time that the service can start at request r_1 when considering γ parameters attained their worst-case value by checking which of the following strategies results in a higher value: request r_1 opening time; γ flights before the departure of request attained their worst-case value while the flights between l_0 and l_1 took a nominal time; $\gamma - 1$ requests before r_0 and a single flight between l_0 and l_1 (request or positioning flights) attained their worst-case value; or both flights between l_0 and l_1 and $\gamma - 2$ previous flights had maximum duration. If $\gamma = 0$ or $\gamma = 1$, the algorithm uses the first two lines in (4.3.3) as boundary conditions.

The other time variables ($DT_{l\gamma}$ and $Fly_{l\gamma}$) are computed using a similar strategy. For $DT_{l\gamma}$ we must consider the split duty rule, that is, when computing the accumulated duty time, we should use one of the expressions in Table 21 depending on the time the crew remains grounded. For variable $Fly_{l\gamma}$, only the flying time parameter($T^{p_r}_{i_r j_r}$) should be considered in the calculations. Thus, $Fly_{l\gamma}$ have a similar expression to (4.3.1) but does not consider maintenance, turnaround times or other parameter not related to travel time. Finally, the overtime costs for each can be computed with a similar strategy as in the deterministic case, with the exception that to compute $OT_{l\gamma}$ we use $Fly_{l\gamma}$ and $DT_{l\gamma}$ with the respective γ in the expression. Furthermore, for an option that requires a rest, $DT_{l\gamma}$ and $Fly_{l\gamma}$ take into account only the flights and their respective deviations after the rest. For example, let $DT_{l_3\gamma}$ be the duty time when we consider case 3 and γ worst-case realizations, and let j_0 and i_3 be, respectively, the arrival airport at the parent request of l_3 and the departure airport of request related to label l_3. Then, to compute $DT_{l_3\gamma}$ we employ the following expression:

$$DT_{l_3\gamma} = \begin{cases} PRE + \bar{T}^{p_3}_{j_0 i_3} + TAT^s_{i_3}, & \text{if } \gamma = 0, \\ \max\{PRE + \bar{T}^{p_3}_{j_0 i_3} + TAT^s_{i_3}, & \\ \qquad PRE + \bar{T}^{p_3}_{j_0 i_3} + \hat{T}^{p_3}_{j_0 i_3} + TAT^s_{i_3}\}, & \text{if } \gamma \geq 1. \end{cases}$$

Similarly, for the knapsack uncertainty set, we compute for each $\delta \leq \Delta$ the total elapsed time ($Elap_{l\delta}$), accumulated duty time ($DT_{l\delta}$) and flying time ($Fly_{l\delta}$) using a dynamic programming strategy. For this set, the algorithm uses equations inspired in (3.3.3) as basis. For instance, if one wants to compute the total elapsed time using case 1, generating label l_1, the following expression should be used:

$$Elap_{l_1\delta} = \max\Big\{a_{r_1}, \max_{\gamma \leq \min\{\delta, \hat{T}^{p_{r_0}}_{i_0 j_0} + \hat{T}^{p_1}_{j_0 i_1}\}} \{Elap_{l_0(\delta-\gamma)} + \bar{T}^{p_{r_0}}_{i_0 j_0} + TAT^r_{j_0} + \bar{T}^{p_1}_{j_0 i_1} + TAT^s_{i_1} + \gamma\}\Big\}. \tag{4.3.4}$$

A solution is robust infeasible if there is no feasible labels (due to time windows constraints) in one of the buckets. In this case, the same feasibility cuts used in the deterministic problem, given by (4.3.1), are introduced into the model, cutting the solution off. Similarly, the constraints added to enforce the computed overtime costs of the route, given by (4.3.2), are the same as in the deterministic case, with the exception that we consider the overtime for the worst-case scenario ($OT_{r\Gamma}$ or $OT_{r\Delta}$) instead of the deterministic solution.

4.4 Computational results

In this section, we present the results of computational experiments carried out to verify the performance of the proposed solution approaches and analyze their relevance to decision-making in practice. The experiments were run on a PC with a processor Intel Core i7-4790 3.6 GHz CPU and 16 GB RAM. The algorithm was implemented in language C++, on top of the

Concert Library of the IBM CPLEX Optimization Studio v.12.10, and the cuts were inserted using the generic callback routines provided by the library.

4.4.1 Instance description

The instances used in these experiments are actual data provided by an airline company and correspond to four months of flight logs. The first month comprises 10 days of operation and a total of 112 requests (including customer requests and maintenance events); the second involves 10 days and 129 requests; the third consists of 8 days and 107 requests; and the fourth month, a higher demand period, has 16 days and 576 requests. As proposed in Munari and Alvarez (2019), we group the flights of each month in instance classes M1 to M4, such that each instance covers three days of operation, which is compatible with the company's usual planning horizon (up to three days). Table 22 summarizes the instance classes information, showing the total number of days (Day), instances (Ins) and the total (Tot) and average (Avg) number of customer flight requests (L), maintenance events (M) and total requests (n = L+M) for each class. It also presents the average number of airports (Air), number of aircraft (V) and types of aircraft (P) available in the instances of that respective month.

Table 22 – General instance information

M	Day	Ins	L		M		n		Air	V	P
			Tot	Avg	Tot	Avg	Tot	Avg			
M1	10	8	58	18.19	54	16.94	112	35.13	30.00	18.38	7
M2	10	8	42	13.27	87	27.48	129	40.75	29.88	22.00	7
M3	8	6	95	35.66	12	4.51	107	40.17	46.00	21.83	7
M4	16	14	362	68.82	214	40.68	576	109.5	82.79	49.71	7

Overall, the first three classes have relatively small instances. They differ in the proportion of customer and maintenance requests. Particularly, M1 is characterized by having a more balanced distribution, with a similar number of requests of each type. Instances in class M2, on the other hand, have considerably more maintenance events while instances in M3 are composed almost exclusively by customer flight requests. Additionally, instances in M3 tend to have more airports. These differences are expected to impact computational performance and cost composition of the solutions, with M3 being more complex and requiring more outsourcing as it contains more customers requests. Finally, class M4 is considerably larger than the previous ones, averaging over 100 requests per instance and more than twice as many aircraft and airports. Thus, instances of this month are expected to be considerably more challenging to be solved than the others.

While the data provided by the company may have served as the basis for setting most parameters, not all information was available and had to be estimated. Table 23 details this information, listing the model parameters and showing whether the data was provided by the company or estimated, and how it was estimated. Additionally, since we have no information about the crews' duty status, we assume that all crew members start to work on the planning horizon immediately after a rest.

Table 23 – Data source description

Parameter	**Data source**		**Estimation strategy**
	Provided	**Estimated**	
AV_v	x		
k_v	x		
i_r	x		
j_r	x		
t_v	x		
p_r		x	Aircraft type used to serve request r in the flight logs
υ_r	x		
υ_p	x		
d_{ij}		x	Great circle distance
$\bar{T}^p_{ij}$		x	$\frac{d_{ij}}{\upsilon_p}$
TL_r	x		
TAT^r_k		x	Estimated by the company as 20 min
c_v	x		
C_{vrs}		x	$c_{t_v}\bar{T}^{t_v}_{j_r i_s}$
$Cout_r$		x	$2c_{p_r}\bar{T}^{p_r}_{i_r j_r}$
cov_r		x	$1.5c_{p_r}$
ST_r	x		
Δ_L	x		
Δ_M	x		
$MaxDuty$	x		
$MaxFly$	x		
PRE	x		
POS	x		

4.4.2 Computational performance of the deterministic approaches

In this topic, we compare the performance of the proposed formulations. For this purpose, we present Table 24, which shows the average computational time (CPUt) and the number of instances solved to optimality (Opt) per class of instances, when solving the deterministic problem using the MTZ-based (MTZ) and commodity flow (CF) models. We present the results using the compact models standalone and the results of the full B&C algorithm. Since all approaches provided optimal solutions for all instances, we omit the solution cost information from Table 24. An analysis of these solutions regarding their costs is presented in Section 4.4.3.

Table 24 – Comparative results between the MTZ and CF formulations as compact models and within the proposed B&C method

	Compact				**B&C**			
	MTZ		**CF**		**MTZ**		**CF**	
	Opt	**CPUt(s)**	**No**	**CPUt(s)**	**Opt**	**CPUt(s)**	**Opt**	**CPUt(s)**
M1	8	0.14	8	0.31	8	0.11	8	0.31
M2	8	0.16	8	0.30	8	0.16	8	0.90
M3	6	0.27	6	1.20	6	0.65	6	1.68
M4	14	3.43	14	39.19	14	247.00	12	415.27
Total	**36**	**2.06**	**36**	**22.96**	**36**	**96.23**	**34**	**242.46**

As the results indicate, the MTZ model outperforms the commodity flow formulation for each instance classes and for both solution approaches, as noted by the average computational times. More importantly, there were two instances (M4_10to12 and M4_11to13) that the B&C algorithm with a CF formulation failed to prove optimality within the time limit, while it succeeded when using the MTZ model. With this strictly superior performance, we used the

MTZ formulation as basis for the B&C algorithm developed for the RO problem, whose results are presented in Section 4.4.4.

One may notice that the B&C algorithms took considerably longer to be solved than the standalone compact formulations. That is because the compact formulations and the B&C algorithms model different problems. The compact formulations only considers the routing constraints of the problem, while the B&C algorithms also ensures the crew requirements described in Section 4.1.2. In fact, the B&C algorithms start with a compact formulation and only when the solver finds an integer solution that is feasible for the routing constraints, it runs the separation algorithm to verify if it is also feasible for crew rules, and introduce cuts to remove the solution if it is not. Thus, it is expected that the B&C algorithms take longer to be solved than their respective compact formulation, since in the best-case scenario, where the standalone compact models and the B&C algorithms have the same optimal solution, the B&C algorithm must, in addition to following the same solution steps used to solve the standalone compact model, run the separation algorithm at least once.

4.4.3 Solution quality in the deterministic approach

In this topic, we evaluate the performance of the proposed B&C method in a deterministic scenario, and compare the obtained solutions against those implemented by the company. Additionally, we asses the impact of the customer requirements on the solutions by comparing the solutions from the compact model proposed in Munari and Alvarez (2019) with the solutions provided by our B&C algorithm, which not only contemplates all routing characteristics considered in the compact model, but also introduces the crew requirements discussed in Section 4.1.2. Furthermore, we want to evaluate the impact of considering the split duty rule in the problem, and its possible advantages for the company.

We present the results of the B&C method in Table 25. Under the header *Instances*, we have columns ID, n, L and M, which details the characteristics of all instances. Column ID identifies each instance in the format Mx_$y$to$z$, where x indicates the month, and y and z represent the first and last days covered by the instance, respectively. The next header shows the cost of the solution implemented by the company, based on the provided flight logs. In the first column of this header, we show the total positioning cost of the routes taken by the company ($Cpos$). Since overtime costs were estimated and the company does not employ outsourcing, we consider this value as the "real" value of the company's solution. Additionally, we also present an estimate solution cost for the routes provided by the company, when they are enforced in the proposed algorithm considering all constraints and cost types (Est). Note that these routes may correspond to an infeasible solution, as in practice the company may have done last-minute operational adjustments that violate the desired requirements for the routes. If such solution is infeasible regarding the constraints considered in our approaches, we write the term "Inf" instead of the estimated cost. The remaining columns present the total cost (Ctot) and computational time (CPUt) of the solutions with a compact formulation standalone and the B&C algorithm, both using the MTZ-based constraints. We show the information for the MTZ-based model and B&C algorithm because, as seen in Section 4.4.2, they perform better than their counterpart based on CF constraints. Particularly for the B&C algorithm, we present the results without considering the split duty rules (B&C w.o. SD) and using this strategy (B&C SD). It is worth noting that all instances were solved within the time limit (3600 s), and therefore we omit the optimality gap in Table 25. To help illustrate these results, we also present Figure 15, which shows the average solution costs for the routes provided by the company or by the proposed algorithms.

A first point we identified is that, in general, our B&C algorithm was able to provide competitive solutions that were cheaper than the positioning cost of the company's solution for the majority of instances in a relatively short time, taking less than 2 minutes on average. As expected, the instances from the fourth month, which present considerably higher demand, were the hardest to solve. Nevertheless, the B&C obtained optimal solutions for them with an average time of 247 seconds, and only two of them took more than 5 minutes to be solved. Considering that currently the decision maker has to design the routes manually, and our algorithm accounts for most of the relevant crew and routing requirements, we believe that it may strongly assist the company.

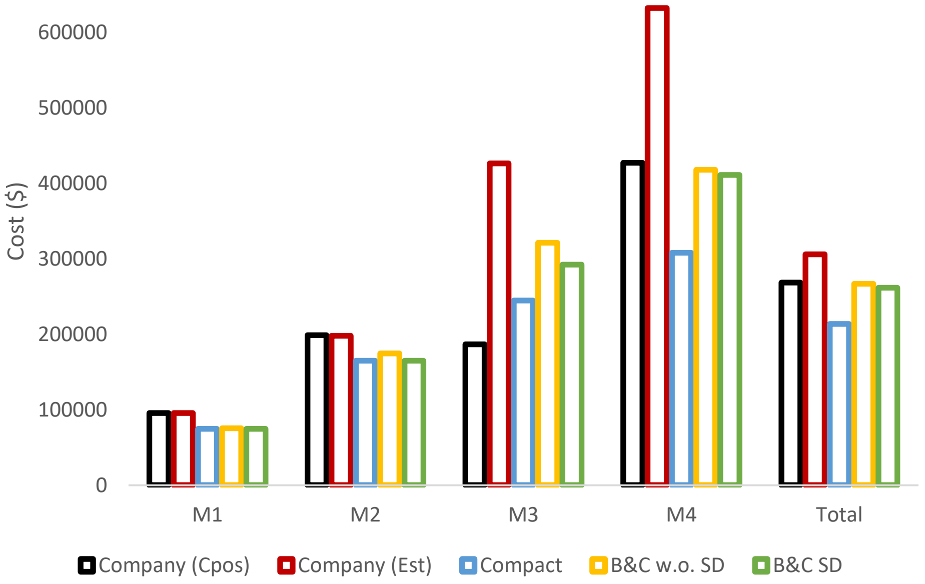


Figure 15 – Average costs for each month of instances

In some instances from M3 and M4, the solution provided by our algorithm was more expensive than the total positioning cost of the solution implemented by the company. This behaviour particularly more noticeable in the instances from M3, whose average cost was considerably higher than the company's solutions. However, when enforcing the company to follow the same constraints used in the B&C algorithm, we notice that many of them are not feasible, or require additional overtime costs which make them more expensive than the solution found by our algorithm.

When comparing the solutions with the standalone compact model, which only considers the routing constraints, and the B&C algorithm, which extends the compact model to consider the crew requirements, it is possible to notice that considering crew requirements had some negative impacts in the optimal solution, increasing its cost by 22.5% on average. This was expected since the B&C deals with more, and tighter, constraints. This difference was more

Table 25 – Computational results of all company's instances

Instance				Company Solution		Compact formulation (MTZ)		B&C w.o. SD (MTZ)		B&C SD (MTZ)	
ID	**L**	**M**	**n**	**Cpos($)**	**Est($)**	**Cot($)**	**CPUt(s)**	**Cot($)**	**CPUt(s)**	**Cot($)**	**CPUt(s)**
M1_1to3	14	22	37	87,527.54	87,527.54	68,274.49	0.11	72,755.73	0.14	68,274.49	0.11
M1_2to4	17	17	35	130,818.20	130,818.20	103,236.18	0.10	103,236.18	0.10	103,236.18	0.09
M1_3to5	24	16	41	127,816.68	127,816.68	84,257.95	0.23	84,257.95	0.18	84,257.95	0.18
M1_4to6	24	18	43	93,718.44	93,718.44	63,677.10	0.28	63,677.10	0.15	63,677.10	0.17
M1_5to7	22	17	40	89,792.18	89,792.18	71,632.73	0.19	71,632.73	0.19	71,632.73	0.21
M1_6to8	15	16	32	81,559.91	81,559.91	72,926.86	0.08	72,926.86	0.07	72,926.86	0.07
M1_7to9	14	14	29	74,651.34	74,651.34	61,871.79	0.05	61,871.79	0.04	61,871.79	0.05
M1_8to10	11	12	24	77,680.91	77,680.91	71,937.82	0.04	71,937.82	0.03	71,937.82	0.03
Avg M1	**17.63**	**16.50**	**35.13**	**95,445.65**	**95,445.65**	**74,726.87**	**0.13**	**75,287.02**	**0.11**	**74,726.87**	**0.11**
M2_1to3	8	19	28	166,836.06	141,218.51	102,880.69	0.04	149,265.71	0.04	102,880.69	0.03
M2_2to4	11	22	34	168,477.81	168,477.81	126,331.90	0.06	137,949.11	0.11	126,331.90	0.06
M2_3to5	13	24	38	156,857.73	156,857.73	126,357.93	0.12	131,363.31	0.54	126,357.93	0.10
M2_4to6	17	33	51	262,145.59	262,145.59	232,142.36	0.27	232,142.36	0.20	232,142.36	0.23
M2_5to7	18	31	50	179,238.59	Inf	158,376.20	0.31	158,376.20	0.27	158,376.20	0.31
M2_6to8	14	29	44	249,730.98	249,730.98	241,022.33	0.18	241,022.33	0.16	241,022.33	0.17
M2_7to9	10	28	39	181,115.44	181,115.44	130,213.48	0.13	130,829.93	0.08	130,213.48	0.20
M2_8to10	12	29	42	226,630.02	226,630.02	200,979.15	0.14	216,102.39	0.79	200,979.15	0.16
Avg M2	**12.88**	**26.88**	**40.75**	**198,879.03**	**198,025.15**	**164,788.01**	**0.16**	**174,631.42**	**0.27**	**164,788.01**	**0.16**
M3_1to3	49	11	61	277,902.02	282,101.92	228,692.93	0.73	262,018.97	2.82	243,932.78	0.78
M3_2to4	44	7	52	241,087.32	730,697.34	462,501.25	0.44	538,779.95	1.38	520,499.80	0.53
M3_3to5	36	5	42	180,030.30	Inf	120,789.34	0.23	213,290.12	0.37	204,764.81	0.26
M3_4to6	26	2	29	164,932.85	Inf	453,066.28	0.07	526,115.63	0.29	526,115.63	0.16
M3_5to7	28	2	31	230,174.19	465,013.52	186,768.79	0.1	256,682.31	1.05	218,272.58	2.03
M3_6to8	24	1	26	227,406.85	227,406.85	15,160.16	0.05	130,388.71	0.27	39,247.11	0.13
Avg M3	**34.50**	**4.67**	**40.17**	**186,854.47**	**426,304.91**	**244,496.46**	**0.27**	**321,212.61**	**1.03**	**292,138.78**	**0.65**
M4_1to3	67	41	109	311,877.44	Inf	461,045.84	4.38	462,133.28	4.29	462,133.28	4.81
M4_2to4	58	39	98	350,029.82	Inf	221,291.02	5.24	242,866.24	127.11	242,866.24	227.70
M4_3to5	62	36	99	341,188.47	341,188.47	212,073.68	5.03	237,273.05	11.46	237,273.05	10.03
M4_4to6	68	35	104	350,093.88	649,659.67	220,993.17	5.18	233,126.20	5.42	233,126.20	5.68
M4_5to7	70	39	110	389,702.09	546,518.00	257,494.41	4.54	372,691.53	81.89	372,291.54	182.21
M4_6to8	63	40	104	338,398.91	Inf	201,465.39	3.00	322,078.44	317.85	317,329.16	141.02
M4_7to9	72	41	114	518,951.14	Inf	247,021.57	4.13	308,729.67	502.51	296,886.99	78.09
M4_8to10	68	33	102	614,517.94	912,577.16	263,580.30	3.08	339,445.07	64.32	332,778.57	63.29
M4_9to11	76	34	111	569,620.72	Inf	319,241.10	4.46	710,695.06	1965.35	703,831.49	1554.09
M4_10to12	80	37	118	513,440.96	939,030.32	398,348.25	10.32	496,159.32	1538.58	484,849.23	952.67
M4_11to13	82	42	125	476,176.69	Inf	361,747.15	6.82	498,431.98	256.47	491,143.94	158.25
M4_12to14	71	49	121	417,695.68	Inf	313,540.69	6.61	679,998.20	59.86	666,998.52	67.54
M4_13to15	62	50	113	415,337.41	Inf	554,892.12	4.78	608,637.49	27.44	593,957.81	8.15
M4_14to16	56	48	105	373,520.17	404,252.74	276,302.49	2.47	336,551.51	4.86	318,407.91	4.51
Avg M4	**68.21**	**40.29**	**109.50**	**427,182.24**	**632,204.39**	**307,788.37**	**5.00**	**417,772.64**	**354.81**	**410,990.99**	**247.00**
Total	**39.06**	**26.08**	**66.14**	**268,241.17**	**305,927.49**	**213,670.41**	**2.06**	**267,027.59**	**141.86**	**261,745.15**	**96.23**

noticeable in months with higher customer demand, such as months 3 and 4, because the crew has less opportunities to rest than months with proportionally more maintenance events. Remarkably, all instances in months 1 and 2 actually had the same solution to the problem with and without crew requirements. Moreover, using the B&C algorithm to insert these constraints iteratively resulted in longer solution times than simply running the compact model with no crew constraints. Nevertheless, this increase was fairly small in the smaller instances (M1-M3), with increments of less than 1 second for all instances, and even the instances from M4 were solved within the time limit.

Finally, when comparing the results of columns “B&C w.o. SD” and “B&C SD”, it is possible to note that split duty rule employed by the company brings benefits. First, using this strategy can reduce the total cost of the solution in about 2%, because taking this strategy could save some overtime costs or allow the company to take some routes that would not be available otherwise. Not only that, but since this rule slightly relax the problem’s requirements, employing the split duty rule can actually speed up the computational times of the B&C algorithm, as some solutions that would be cut-off may be accepted, finishing the search quicker. This is noticeable in Table 25, where instances were solved more than 40% quicker when the split duty rule was employed, even though there is a slight increase in the time the algorithm takes to verify each node in the tree, as it requires one more verification step.

After comparing the solutions provided by our algorithm with those used by the company and study the impact of the crew requirements, we shift our focus to the solution cost and time composition. As noted in the instances description, each month has different characteristics regarding the proportion of customers request (M1-M3) or total demand (M4) and are expect to behave differently because of this. Particularly, it is expected that instances with more customer requests face more overtime and outsourcing costs, since there will be less resting opportunities for the crew. To assist in this analysis we present Table 26, which details the solution costs and times for each instance. Based on the solution provided by the B&C algorithm (with split duty rules), we show the total positioning (Pos), upgrade (Up), outsourcing (Out) and overtime ($Over$) costs and times. Additionally, we also present the aggregated total cost (Tot) and the total time the crew members worked in customer requests (Req) in each instance. To help visualize these values, we also present Figure 16, which illustrates the average cost and time composition of the solutions for each month’s instances.

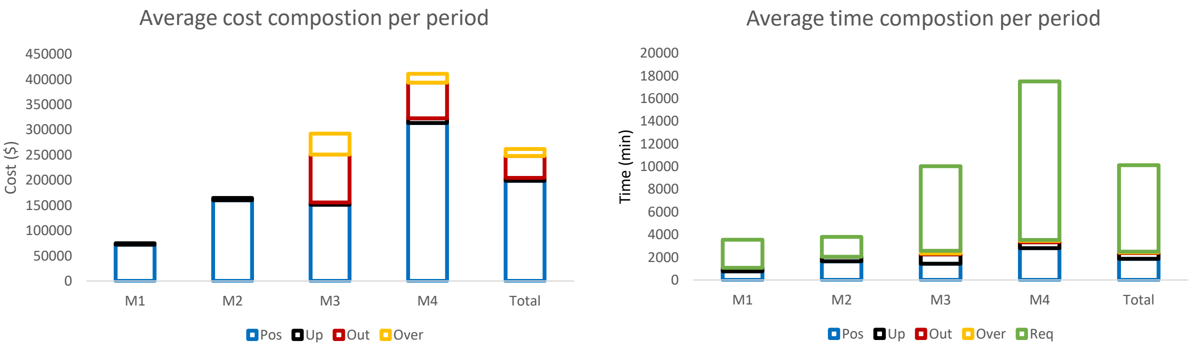


Figure 16 – Detailed costs and times for each month of instances

Table 26 – Total costs and times for each instance

Instance	**Cost ($)**					**Time (min)**				
ID	**Pos**	**Up**	**Out**	**Over**	**Tot**	**Pos**	**Up**	**Out**	**Over**	**Req**
M1_1to3	64413.69	3860.80	0.00	0.00	68274.49	686	240	0	0	1495
M1_2to4	101415.89	1820.29	0.00	0.00	103236.18	921	197	0	0	2058
M1_3to5	81364.02	2893.93	0.00	0.00	84257.95	906	289	0	0	2769
M1_4to6	61739.88	1937.22	0.00	0.00	63677.10	692	166	0	0	3711
M1_5to7	66465.02	5167.71	0.00	0.00	71632.73	734	416	0	0	3565
M1_6to8	70756.46	2170.40	0.00	0.00	72926.86	799	223	0	0	2808
M1_7to9	58959.51	2912.28	0.00	0.00	61871.79	684	473	0	0	1903
M1_8to10	69947.47	1990.35	0.00	0.00	71937.82	662	394	0	0	1624
Avg M1	**71882.74**	**2844.12**	**0.00**	**0.00**	**74726.87**	**760.50**	**299.75**	**0.00**	**0.00**	**2491.63**
M2_1to3	99768.30	3112.39	0.00	0.00	102880.69	1081	217	0	0	896
M2_2to4	119755.98	6575.92	0.00	0.00	126331.90	1280	308	0	0	1334
M2_3to5	119703.68	6654.25	0.00	0.00	126357.93	1287	407	0	0	1778
M2_4to6	228120.40	4021.96	0.00	0.00	232142.36	2224	467	0	0	2544
M2_5to7	158376.20	0.00	0.00	0.00	158376.20	1711	439	0	0	2698
M2_6to8	241022.33	0.00	0.00	0.00	241022.33	2311	573	0	0	2117
M2_7to9	127410.97	2802.51	0.00	0.00	130213.48	1361	250	0	0	1378
M2_8to10	187461.89	13517.26	0.00	0.00	200979.15	1848	596	0	0	1205
Avg M2	**160202.47**	**4585.54**	**0.00**	**0.00**	**164788.01**	**1637.88**	**407.13**	**0.00**	**0.00**	**1743.75**
M3_1to3	234230.12	5502.76	0.00	4199.90	243932.78	2237	842	0	21	9979
M3_2to4	182774.91	0.00	279726.34	57998.55	520499.80	1925	2176	109	290	9332
M3_3to5	127382.09	4333.36	0.00	73049.36	204764.81	1374	877	0	409	8496
M3_4to6	155934.44	7005.75	290126.08	73049.36	526115.63	1364	272	286	409	5742
M3_5to7	176296.64	11005.47	0.00	30970.47	218272.58	1410	356	0	207	6511
M3_6to8	30047.34	0.00	0.00	9199.77	39247.11	314	428	0	46	4775
Avg M3	**151110.92**	**4641.22**	**94975.40**	**41411.23**	**292138.78**	**1437.33**	**825.17**	**65.83**	**230.33**	**7472.50**
M4_1to3	252051.53	5553.53	204528.22	0.00	462133.28	2416	221	374	0	12785
M4_2to4	229543.56	13322.68	0.00	0.00	242866.24	2132	336	0	0	10981
M4_3to5	228016.23	9256.82	0.00	0.00	237273.05	2121	334	0	0	11437
M4_4to6	218071.50	5854.93	0.00	9199.77	233126.20	1969	179	0	46	13584
M4_5to7	333582.96	3909.45	0.00	34799.13	372291.54	2878	335	0	174	15979
M4_6to8	275839.79	6690.24	0.00	34799.13	317329.16	2390	222	0	174	14108
M4_7to9	267623.62	3664.01	0.00	25599.36	296886.99	2381	203	0	128	14871
M4_8to10	329818.71	2959.86	0.00	0.00	332778.57	2770	458	0	0	12346
M4_9to11	365867.53	6838.90	297325.90	33799.16	703831.49	3166	798	507	169	14593
M4_10to12	432191.14	19458.92	0.00	33199.17	484849.23	3769	1259	0	166	16813
M4_11to13	427212.07	11733.17	0.00	52198.70	491143.94	3862	1212	0	261	18635
M4_12to14	358477.41	5862.01	284259.56	18399.54	666998.52	3523	472	422	92	14655
M4_13to15	365512.98	19518.61	208926.22	0.00	593957.81	3311	689	168	0	13936
M4_14to16	304318.77	14089.14	0.00	0.00	318407.91	2552	691	0	0	11096
Avg M4	**313437.70**	**9193.73**	**71074.28**	**17285.28**	**410990.99**	**2802.86**	**529.21**	**105.07**	**86.43**	**13987.07**
Total	**198652.08**	**5999.91**	**43469.23**	**13623.93**	**261745.15**	**1862.53**	**500.42**	**51.83**	**72.00**	**7626.03**

A first information that can be extracted from these results is that the positioning costs are the main component of the optimal solutions for all months and represents, on average, 18.5% of the total flying time. This is a good result brought by our algorithm because, in general, air transportation companies spend more than 35% of their fleet time on positioning flights (Yao et al., 2008). In fact, even if we account for the outsourcing time, our solutions allow crew members to use almost 80% of their flight time working on customer requests, a value considerably higher than the industry average. This is considerably positive, as this reduction affects the main operational cost of the company, and thus providing a significant competitive advantage for the company.

We are also able to confirm the previous insight regarding the outsource and overtime costs, which are non-existent in the months characterized by fewer customer requests (M1-M2) and more frequent in the remaining periods. Even so, the solver tends to avoid these costs as much as possible, as their unit costs are considerably high. Outsourcing costs in particular are only used if the solver is unable to service a customer request on time, and are usually taken twice at most in a single instance. This outsourcing is usually a consequence of the assumption that, at the beginning of the planning horizon, each crew member has just finished their rest, so a presentation time of 40 minutes (PRE) is required, which makes customer orders with very small closing time windows. In the real-world application, it is possible that this presentation time would not be required (if the crew was in the middle of their duty at the start of the planning horizon), allowing the customer to be serviced on time. Additionally, even though we considered time windows as strong constraints in our model, the company can simply choose to start the service late and compensate the customer in some other less costly way.

Moreover, upgrade costs have little impact on the total solution costs, accounting for only 2.3% of the total value. This is advantageous since it allows the solver to replace the more expensive positioning, overtime and outsourcing costs with a cheaper strategy. Indeed, by looking into M3 and M4 data, we notice that even though the solution has a longer upgrade time, outsourcing and overtime increase the total costs much more than the upgrades, a consequence of these strategies having considerably higher unit costs.

4.4.4 RO approaches and impact of uncertainty

In this topic, we evaluate the impact of the RO approaches on the solutions with respect to cost, robustness and computational performance. For this purpose, we analyze the results of the model considering two different uncertainty sets for time travel, namely the cardinality constrained uncertainty set, with budget values $\Gamma \in \{0, 1, 5, 10\}$, and the knapsack uncertainty set with $\Delta \in \{0, 20, 40, 60\}$. When $\Gamma = 0$ or $\Delta = 0$, the algorithm is solving the deterministic case. For both uncertainty sets, we assume time deviations $\alpha \in \{0.1, 0.25, 0.5\}$.

To evaluate the robustness of a solution, we run a Monte Carlo simulation with 1000 scenarios, in which the travel times are randomly generated value between $\bar{T}^p_{ij}$ and $\bar{T}^p_{ij}(1 + \alpha)$, following a continuous uniform distribution. We evaluate the feasibility of each scenario by checking if it is possible to meet all requests, respecting the time windows, following the solution route. Note that once the route is fixed in a solution, the positioning, upgrade and outsourcing costs do not change as their parameter values are estimated a priori with the deterministic value, and thus only the overtime costs may change in a given scenario.

Table 27 summarizes the results for the B&C method using the cardinality constrained uncertainty set. For each Γ, α and month, we present the total number of instances tested (Ins), number of instances deemed infeasible (Inf) and solved to optimality (Opt) by the proposed B&C algorithm. We also present the average computational time (CPUt) and optimality gap (Gap) to solves instances of a specific group. Finally, we present the percentage of infeasible scenarios in the Monte Carlo simulation (Risk) and the percentage increase in the cost of the robust solution over the deterministic solution, named as the price-of-robustness (PoR), for each combination of α and Γ.

As expected, the algorithm tends to be more conservative as Γ increases, resulting in solutions with lower risks but higher costs. For example, in higher deviation scenarios ($\alpha = 0.5$), robust solutions may cost more than twice as much as the deterministic solution, but in return they are nearly immune to variability. The decision maker is free to choose whether they take a more expensive, and safer route, or prefers to risk a cheaper route. Regardless, the solutions provided by the robust optimization method are generally relevant, with trade-offs that can help decision makers take an informed decision. Thus, we can conclude that the proposed method can be useful as a decision support tool.

Despite the generally competitive results, the RO models had some limitations. The first one is that in more constrained problems, with higher deviation α and budget Γ, there is a greater chance that the algorithm will deem the instance infeasible for this combination of α and Γ. For example, more than half of the instances in M4 in a high deviation scenario ($\alpha = 0.5$) were proven to be infeasible when considering $\Gamma = 10$. Another limitation of the approach is that it struggles to solve larger instances, especially in more constrained instances. While the algorithm was able to solve all deterministic M4 instances, it did not find an optimal solution or prove infeasibility to some instances with deviation. This is particularly more frequent in instances with higher α and Γ, since the B&C tends to repetitively cut-off candidate solutions, as we have a more restricted problem.

An outlier in Table 27 is that in some set of instances with a fixed deviation level α, the average risk and PoR are lower in some instances with higher Γ. Notably, instances from M4, $\alpha = 0.5$ and $\Gamma = 10$ presented, on average, higher risks and lower costs. This behavior is consequence of some of the more expensive, and risky, instances turning infeasible when we considered higher budgets, lowering these average overall.

Analysing the average solutions per month, we notice that the deterministic solutions for instances in months M3 and M4 were generally the ones with worst performance. This is a consequence of the higher quantity of customer requests, which naturally have tighter time windows, in those instances and small travel time deviations can easily make the aircraft arrive too late to serve the demand. To avoid this problem, the RO approach generally uses more aircraft, with less requests per route and possibly more positioning, and outsourcing, which considerably increases the costs overall. This is particularly noticeable in instances from month 3 and 4 with $\alpha = 0.5$, where the robust solutions cost more than twice as much as the deterministic ones, in exchange for completely nullifying the risks (which were more than 70% originally).

When compared to instances from M3 and M4, the second month, which is characterize for having proportionally more maintenance events, presented less variability between deterministic

Table 27 – Results of the proposed B&C algorithm for the cardinality constrained uncertainty set

$\alpha = 10\%$								
Γ	**M**	**Ins**	**Inf**	**Opt**	**CPUt(s)**	**Gap(%)**	**Risk (%)**	**PoR(%)**
0	**M1**	8	0	8	0.11	0.0%	11.58%	-
	M2	8	0	8	0.16	0.0%	0.00%	-
	M3	6	0	6	0.65	0.0%	0.00%	-
	M4	14	0	14	247.00	0.0%	31.02%	-
1	**M1**	8	0	8	0.12	0.0%	0.00%	0.37%
	M2	8	0	8	0.17	0.0%	0.00%	0.16%
	M3	6	0	6	1.63	0.0%	0.00%	5.57%
	M4	14	1	9	1087.95	0.8%	16.67%	8.42%
5	**M1**	8	0	8	0.20	0.0%	0.00%	0.37%
	M2	8	0	8	0.20	0.0%	0.00%	0.16%
	M3	6	0	6	2.69	0.0%	0.00%	6.30%
	M4	14	4	5	1314.38	1.5%	16.67%	9.02%
10	**M1**	8	0	8	0.29	0.0%	0.00%	0.37%
	M2	8	0	8	0.28	0.0%	0.00%	0.16%
	M3	6	0	6	3.55	0.0%	0.00%	5.80%
	M4	14	5	4	616.41	1.4%	14.29%	8.09%
Total		**144**	**10**	**120**	**204.74**	**0.23%**	**5.64%**	**3.73%**

$\alpha = 25\%$								
Γ	**M**	**Ins**	**Inf**	**Opt**	**CPUt(s)**	**Gap(%)**	**Risk(%)**	**PoR(%)**
0	**M1**	8	0	8	0.11	0.0%	27.66%	-
	M2	8	0	8	0.16	0.0%	14.36%	-
	M3	6	0	6	0.65	0.0%	26.70%	-
	M4	14	0	14	247.00	0.0%	50.94%	-
1	**M1**	8	0	8	0.18	0.0%	0.05%	6.79%
	M2	8	0	8	0.15	0.0%	7.06%	1.27%
	M3	6	1	5	3.41	0.0%	0.00%	40.23%
	M4	14	1	5	2066.31	10.5%	10.00%	23.07%
5	**M1**	8	0	8	0.22	0.0%	0.00%	7.28%
	M2	8	0	8	0.27	0.0%	2.63%	2.21%
	M3	6	2	4	7.58	0.0%	0.00%	60.03%
	M4	14	4	6	1469.98	7.0%	0.00%	30.70%
10	**M1**	8	0	8	0.33	0.0%	0.00%	7.28%
	M2	8	0	8	0.32	0.0%	2.63%	2.21%
	M3	6	3	3	3.53	0.0%	0.00%	27.04%
	M4	14	7	4	1629.95	6.6%	0.00%	24.71%
Total		**144**	**18**	**111**	**339.38**	**1.51%**	**9.43%**	**19.40%**

$\alpha = 50\%$								
Γ	**M**	**Ins**	**Inf**	**Opt**	**CPUt(s)**	**Gap(%)**	**Risk(%)**	**PoR(%)**
0	**M1**	8	0	8	0.11	0.0%	47.90%	-
	M2	8	0	8	0.16	0.0%	31.81%	-
	M3	6	0	6	0.65	0.0%	79.48%	-
	M4	14	0	14	247.00	0.0%	71.51%	-
1	**M1**	8	0	8	0.39	0.0%	0.00%	23.16%
	M2	8	0	8	0.48	0.0%	11.50%	4.66%
	M3	6	3	3	19.82	0.0%	2.77%	81.67%
	M4	14	6	0	2064.51	19.4%	6.36%	71.87%
5	**M1**	8	0	8	0.55	0.0%	6.90%	21.61%
	M2	8	0	8	0.41	0.0%	7.04%	6.39%
	M3	6	4	2	43.09	0.0%	0.00%	122.47%
	M4	14	10	0	1028.96	10.6%	0.00%	95.98%
10	**M1**	8	1	7	1.31	0.0%	7.89%	16.74%
	M2	8	0	8	0.97	0.0%	7.04%	6.48%
	M3	6	4	2	59.60	0.0%	0.00%	128.83%
	M4	14	10	0	1288.34	15.7%	0.00%	122.99%
Total		**144**	**38**	**90**	**297.27**	**2.85%**	**16.96%**	**58.57%**

and robust solutions. Particularly, the deterministic solution for these instances were less sensible to deviation, presenting lower average risks than the other months. While more robust solutions tend to increase costs and reduce risks, they do so less intensively. At the most conservative configuration, with $\alpha = 0.5$ and $\Gamma = 10$, the RO approach presents a solution only 6.48% more expensive than the deterministic solutions, but it was unable to completely nullify the risks. This behavior is consequence of the higher frequency of maintenance events in these instances, which reduces the necessity and possibility of positioning, upgrades and outsourcing, as the maintenance must be executed in a specific aircraft. Thus, the risks in the more conservative route are result of maintenance events in which the designated aircraft is unable to service on time, and it is not possible to outsource this service.

Table 28 present the results of the B&C method with the knapsack uncertainty set, following a similar structure to Table 27, but showing the budget Δ instead of Γ. Generally speaking, one can reach similar conclusions as with the cardinality constrained uncertainty set. Increasing the budget Δ results in more conservative solutions, with higher costs but lower risks. Additionally, more instances are deemed infeasible when higher values for α and Δ are considered. The main difference in behaviour between the solutions provided by the knapsack and cardinality constrained uncertainty sets is a relatively more consistent performance among all deviation levels (α). While the number of instances that the B&C algorithm was able to find an optimal or prove infeasibility decreases with higher α with the cardinality constrained uncertainty set, the number of instances successfully solved remains almost the same to all α when considered the knapsack uncertainty set. This is due to Δ limiting the total deviation regardless of α, while in the cardinality constrained uncertainty set a fixed Γ may incorporate more deviation depending on α size, tightening the problem considerably more when deviation increases.

Furthermore, the algorithm seems to be able to perform better with the knapsack uncertainty set than with the cardinality constrained set, as seen for the lower average computational times and optimality gaps. Thus, using this uncertainty set might be preferable to the decision maker, as it offers solutions with similar robustness and costs as the cardinality constrained uncertainty set, but with lower computational times and an easier-to-estimate budget. The cardinality constrained uncertainty set, however, might be more suitable for instances with lower deviation levels (α=0.1), as it finds more feasible instances than the knapsack set in this environment.

To further explore cost behavior of the robust solutions, we present the results for instance M3_5to7 in Table 29. For each combination of α and Δ, the cost composition of the corresponding solution is given by the positioning (Pos), upgrade (Up), outsourcing (Out), overtime (Over) and total (Tot) costs. Additionally, we present the percentage of infeasible scenarios in the Monte Carlo simulation (Risk) for each solution. This instance was chosen for having a well defining behavior when α increases. To help visualize the cost behavior, Figure 17 shows the composition of the solution cost for each α and Δ.

Moreover, the algorithm proceeds to take more expensive strategies as α increases. When considering a deviation of $\alpha = 0.1$, the algorithm chooses to slightly increase the positioning costs, opting routes with less requests each. Now, when $\alpha = 0.25$, we notice an increase not only in the positioning costs but also in upgrades, a consequence of the more conservative routes. For this level of uncertainty, robust solutions have a positive impact on risk, decreasing the chance of

Table 28 – Results of the proposed B&C algorithm for the knapsack uncertainty set

$\alpha = 10\%$

Δ	M	**Ins**	**Inf**	**Opt**	**CPUt(s)**	**Gap(%)**	**Risk (%)**	**PoR(%)**
0	**M1**	8	0	8	0.14	0.0%	11.58%	-
	M2	8	0	8	0.19	0.0%	0.00%	-
	M3	6	0	6	0.60	0.0%	0.00%	-
	M4	14	0	14	268.34	0.0%	31.02%	-
20	**M1**	8	0	8	0.42	0.0%	0.00%	0.37%
	M2	8	0	8	0.46	0.0%	0.00%	0.00%
	M3	6	0	6	1.50	0.0%	0.00%	6.66%
	M4	14	2	9	471.21	0.8%	17.81%	6.10%
40	**M1**	8	0	8	0.81	0.0%	0.00%	0.37%
	M2	8	0	8	0.80	0.0%	0.00%	0.16%
	M3	6	0	6	5.02	0.0%	0.00%	6.76%
	M4	14	8	6	32.08	0.0%	0.00%	1.39%
60	**M1**	8	0	8	1.15	0.0%	0.00%	0.37%
	M2	8	0	8	1.22	0.0%	0.00%	0.64%
	M3	6	1	5	8.02	0.0%	0.00%	4.29%
	M4	14	11	3	2.33	0.0%	0.00%	0.96%
Total		**144**	**22**	**119**	**49.64**	**0.05%**	**3.78%**	**2.34%**

$\alpha = 25\%$

Δ	M	**Ins**	**Inf**	**Opt**	**CPUt(s)**	**Gap(%)**	**Risk(%)**	**PoR(%)**
0	**M1**	8	0	8	0.14	0.0%	27.66%	-
	M2	8	0	8	0.19	0.0%	14.36%	-
	M3	6	0	6	0.60	0.0%	26.70%	-
	M4	14	0	14	268.34	0.0%	50.94%	-
20	**M1**	8	0	8	0.55	0.0%	16.65%	3.57%
	M2	8	0	8	0.71	0.0%	14.36%	0.35%
	M3	6	0	6	9.02	0.0%	6.60%	38.99%
	M4	14	3	9	466.51	0.7%	13.44%	23.61%
40	**M1**	8	0	8	0.90	0.0%	0.00%	7.28%
	M2	8	0	8	0.85	0.0%	7.06%	1.27%
	M3	6	0	6	27.01	0.0%	6.55%	37.78%
	M4	14	8	6	43.76	0.0%	5.07%	18.15%
60	**M1**	8	0	8	1.33	0.0%	0.00%	7.28%
	M2	8	0	8	1.36	0.0%	7.04%	1.57%
	M3	6	1	5	8.29	0.0%	0.00%	29.56%
	M4	14	12	2	14.17	0.0%	10.13%	7.88%
Total		**144**	**24**	**118**	**52.73**	**0.04%**	**12.91%**	**14.77%**

$\alpha = 50\%$

Δ	M	**Ins**	**Inf**	**Opt**	**CPUt(s)**	**Gap(%)**	**Risk(%)**	**PoR(%)**
0	**M1**	8	0	8	0.14	0.0%	47.90%	-
	M2	8	0	8	0.19	0.0%	31.81%	-
	M3	6	0	6	0.60	0.0%	79.48%	-
	M4	14	0	14	268.34	0.0%	71.51%	-
20	**M1**	8	0	8	0.39	0.0%	23.11%	14.66%
	M2	8	0	8	0.40	0.0%	27.33%	4.84%
	M3	6	0	6	2.30	0.0%	23.47%	108.79%
	M4	14	4	9	752.21	0.8%	34.29%	118.08%
40	**M1**	8	0	8	0.90	0.0%	0.00%	18.25%
	M2	8	0	8	0.85	0.0%	14.86%	6.59%
	M3	6	0	6	4.53	0.0%	19.45%	92.23%
	M4	14	10	4	32.05	0.0%	14.03%	55.63%
60	**M1**	8	0	8	1.36	0.0%	0.00%	19.33%
	M2	8	0	8	1.25	0.0%	11.50%	4.66%
	M3	6	1	5	13.80	0.0%	2.20%	92.44%
	M4	14	13	1	6.78	0.0%	83.20%	68.45%
Total		**144**	**28**	**116**	**67.88**	**0.05%**	**30.26%**	**50.33%**

Table 29 – Costs and risks of the solutions for instance M3_5to7.

α	**Δ**	**Pos ($)**	**Up ($)**	**Out ($)**	**Over ($)**	**Tot ($)**	**Risk(%)**
10%	**0**	176296.64	11005.47	0.00	30970.47	218272.58	0.00%
	20	189696.44	11005.47	0.00	35200.47	235902.38	0.00%
	40	189696.44	11005.47	0.00	44350.41	245052.32	0.00%
	60	189696.44	11005.47	0.00	54500.33	255202.24	0.00%
25%	**0**	176296.64	11005.47	0.00	30970.47	218272.58	8.20%
	20	247694.99	24927.72	0.00	38638.10	311260.81	0.00%
	40	247694.99	24927.72	0.00	47788.04	320410.75	0.00%
	60	247694.99	24927.72	0.00	70938.22	343560.93	0.00%
50%	**0**	176296.64	11005.47	0.00	30970.47	218272.58	76.60%
	20	261170.48	24927.72	278908.04	38638.10	603644.34	0.00%
	40	261170.48	24927.72	278908.04	47788.04	612794.28	0.00%
	60	261170.48	24927.72	278908.04	58537.94	623544.18	0.00%

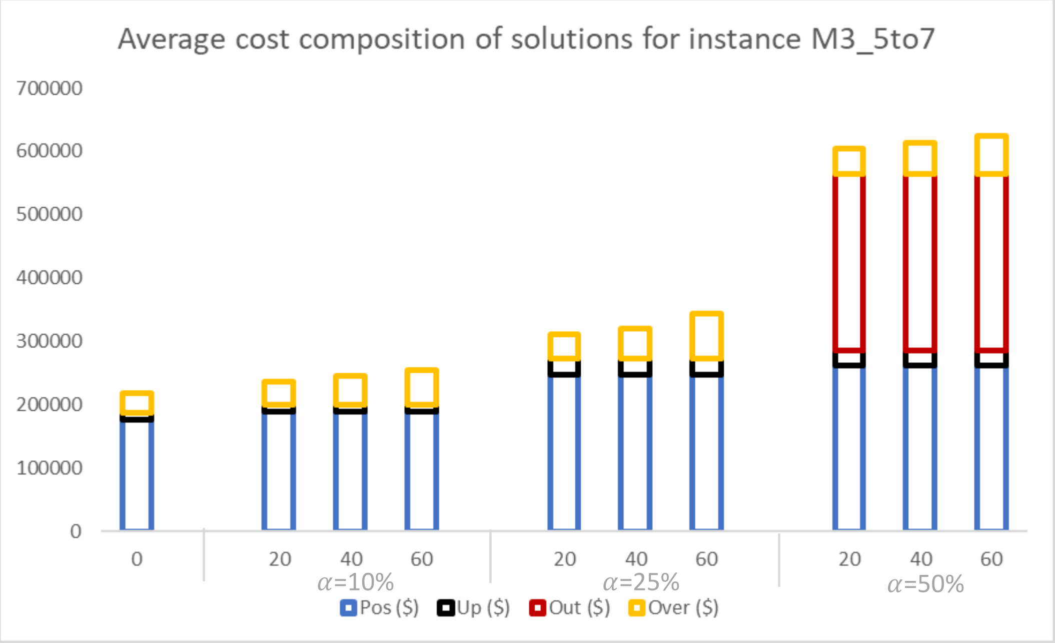


Figure 17 – Cost composition of the solutions for instance M3_5to7 with different configurations of α and Γ.

the solution being infeasible from 9.2% to 0.0%. Finally, with $\alpha = 0.5$, in addition to increase the positioning costs a little more, the solver chooses to pay for a costly outsourcing to ensure that a customer request is serviced. Although expensive, this strategy was very beneficial in terms of service level, reducing the risks from 76.6% to 0.0%. For this particular instance, changes in Δ with a fixed α have little effect in the solution, only changing the expected overtime costs.

5 Concluding Remarks and Future Work

We addressed traditional and real-world variants of the robust vehicle routing problem, with focus on effective modeling and solution approaches. We presented the foundations and the relevant literature review and then proposed novel deterministic and robust optimization formulations based on commodity flow (CF) constraints considering the traditional cardinality constrained uncertainty set. Furthermore, we developed new approaches based on the single and the multiple knapsack uncertainty sets, deriving robust counterparts using the linearization technique of dynamic programming equation. It is worth noting we found no work in literature that proposed a compact formulation for the robust VRP with time windows (VRPTW) under uncertainty on travel time for the knapsack uncertainty sets.

Computational experiments for the robust VRPTW using using benchmark instances adapted from the literature were conducted. The main objective of these experiments was to compare the proposed approaches for the cardinality constrained uncertainty set with the state-of-the-art compact model. The results show that our formulations has stronger linear relaxation and better results in instances with lower uncertainty levels and in some instance classes. Moreover, we implemented a tailored branch-and-cut (B&C) algorithm for both studied formulations to improve their results in the benchmark instances. We noted that although our commodity flow was improved by the B&C algorithm, the state-of-the-art model, based on MTZ-based constraints, presents more advantages in general, solving more instances to optimality within the time limit and having lower optimality gaps for the ones that were not solved.

We also developed new models and solution approaches for the addressed real-life variant, considering the deterministic problem and its robust extension. This variant is an aircraft routing problem motivated by the case of a company offering on-demand passenger transportation services. In the compact formulations, we considered all the routing constraints of the problem, such as different types of requests, time windows and possibility of *upgrades*. Additionally, we proposed a B&C algorithm based on labeling strategies that dynamically introduces crew requirements into the deterministic problem and its robust counterpart. The proposed algorithm considers most requirements related to routing and crew regulation, namely the minimum rest time, maximum duty time, maximum flying time in a single duty, presentation times at the start and end of a duty, the split duty rule and the possibility of working overtime. This separation algorithm was then extended to consider uncertainty in travel times following robust optimization paradigms, and considering two different uncertainty sets: the traditional cardinality-constrained set and the single knapsack set. To our knowledge, no work in the literature addresses an air transport problem considering all of the previous requirements in a deterministic or stochastic scenario.

Computational experiments using real-life data provided by the company showed that the deterministic approach obtained optimal solutions for all instances within reasonably short computational times. Moreover, the solution costs were, on average, considerably lower than the costs of the routes employed by the company. We also noted that the version of the algorithm with MTZ-based constraints outperformed the one based on commodity flow constraints. The experiments with the robust approaches were more computationally demanding, and they were unable to find the optimal solution for some larger instances. Nevertheless, the algorithm was successful in providing interesting solutions with trade-offs between cost and robustness for most instances, leaving it up to the decision maker to choose a solution aligned with their strategy. Thus, we can be concluded that the proposed algorithm can serve as a useful decision support tool for the company, and provide competitive solutions for real-world instances in a short time.

There are different possible directions for future works. One would be to develop branch-price-and-cut algorithms to solve the literature's RVRPTW instances as well as the robust aircraft routing problem, 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 the studied variants. 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 the decision making process.

References

Agra, A. et al. Layered formulation for the robust vehicle routing problem with time windows. In: Mahjoub, A. R.; Markakis, V.; Milis, I.; Paschos, V. T. (Ed.). *Combinatorial Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 249–260.

Agra, A. et al. The robust vehicle routing problem with time windows. *Computers & Operations Research*, v. 40, n. 3, p. 856 – 866, 2013. Transport Scheduling.

Augerat, P. *Approche polyedre du probleme de tournees de vehicules*. Tese (Doutorado) — Institut National Polytechinque de Grenoble, 1995.

Balinski, M. L.; Quandt, R. E. On an integer program for a delivery problem. *Operations Research*, v. 12, n. 2, p. 300–304, 1964.

Bartolini, E.; Goeke, D.; Schneider, M.; Ye, M. The robust traveling salesman problem with time windows under knapsack-constrained travel time uncertainty. *Transportation Science*, v. 55, n. 2, p. 371–394, 2021.

Ben-Tal, A.; Goryashko, A.; Guslitzer, E.; Nemirovski, A. Adjustable robust solutions of uncertain linear programs1. *Mathematical Programming*, v. 99, p. 351–376, 01 2004.

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

Bertsimas, D.; Sim, M. Robust discrete optimization and network flows. *Mathematical Programming*, v. 98, p. 43–71, 2003.

Bertsimas, D.; Sim, M. The price of robustness. *Operations research*, Informs, v. 52, n. 1, p. 35–53, 2004.

Bianchessi, N.; Irnich, S. Branch-and-cut for the split delivery vehicle routing problem with time windows. *Transportation Science*, v. 53, n. 2, p. 442–462, 2019.

Campos, R.; Alvarez, A.; Munari, P. Otimização robusta aplicada ao roteamento de aeronaves no transporte aÉreo de passageiros sob demanda. *Pesquisa Operacional para o Desenvolvimento*, v. 11, n. 3, p. 139–150, dez. 2019.

Ceria, S.; Stubbs, R. Incorporating estimation errors into portfolio selection: Robust portfolio construction. *Journal of Asset Management*, v. 7, 04 2006.

Charris, E.; Prins, C.; Santos, A. Local search based metaheuristics for the robust vehicle routing problem with discrete scenarios. *Applied Soft Computing*, v. 32, 07 2015.

Chen, Z.; Sim, M.; Xiong, P. Robust stochastic optimization made easy with rsome. *Management Science*, v. 66, n. 8, p. 3329–3339, 2020.

Dantzig, G. B.; Ramser, J. H. The truck dispatching problem. *Management science*, Informs, v. 6, n. 1, p. 80–91, 1959.

De La Vega, J.; Munari, P.; Morabito, R. Robust optimization for the vehicle routing problem with multiple deliverymen. *Central European Journal of Operations Research*, 2018. ISSN 1613-9178. Online First.

Desaulniers, G.; Madsen, O. B.; Ropke, S. The vehicle routing problem with time windows. In: Toth, P.; Vigo, D. (Ed.). *Vehicle routing: Problems, methods, and applications.* [S.l.: s.n.], 2014, (MOS/SIAM Ser Optim). p. 119–159.

Dunbar, M.; Froyland, G.; Wu, C.-L. An integrated scenario-based approach for robust aircraft routing, crew pairing and re-timing. *Comput. Oper. Res.*, Elsevier Science Ltd., GBR, v. 45, p. 68–86, maio 2014. ISSN 0305-0548.

Feillet, D. A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR: A Quarterly Journal of Operations Research*, v. 8, p. 407–424, 2010.

Gavish, B. The deliverying problem:new cutting planes procedures. In: *Apresentado em TIMS XXVI Conference, Compenhagen.* [S.l.: s.n.], 1984.

Gavish, B.; Graves, S. The traveling salesman problem and related problems. 05 1978.

Gendreau, M.; Jabali, O.; Rei, W. Future research directions in stochastic vehicle routing (50th anniversary invited article). *Transportation Science*, v. 50, n. 4, p. 1163–1173, 2016.

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

Gounaris, C. E.; Wiesemann, W.; Floudas, C. A. The robust capacitated vehicle routing problem under demand uncertainty. *Operations Research*, v. 61, n. 3, p. 677–693, 2013.

Gouveia, L. A result on projection for the vehicle routing problem. *European Journal of Operational Research*, v. 85, n. 3, p. 610–624, 1995. ISSN 0377-2217. Disponível em: <https://www.sciencedirect.com/science/article/pii/0377221794000258>.

Haouari, M.; Mansour, F. Z.; Sherali, H. D. A new compact formulation for the daily crew pairing problem. *Transportation Science*, v. 53, n. 3, p. 811–828, 2019.

Hu, C.; Lu, J.; Liu, X.; Zhang, G. Robust vehicle routing problem with hard time windows under demand and travel time uncertainty. *Computers Operations Research*, v. 94, 02 2018.

Irnich, S.; Toth, P.; Vigo, D. The family of vehicle routing problems. In: Toth, P.; Vigo, D. (Ed.). *Vehicle routing: Problems, methods, and applications.* [S.l.: s.n.], 2014, (MOS/SIAM Ser Optim). p. 1–33.

Jamili, A. A robust mathematical model and heuristic algorithms for integrated aircraft routing and scheduling, with consideration of fleet assignment problem. *Journal of Air Transport Management*, Elsevier, v. 58, p. 21–30, 2017.

Khooban, Z. Transportation. *Logistics Operations and Management*, p. 109–126, 12 2011.

Lacasse-Guay, E.; Desaulniers, G.; Soumis, F. Aircraft routing under different business processes. *Journal of Air Transport Management*, Elsevier, v. 16, n. 5, p. 258–263, 2010.

Langevin, A.; Desrochers, M.; Desrosiers, J.; Gelinas, S.; Soumis, F. A two-commodity flow formulation for the traveling salesman and the makespan problems with time windows. *Networks*, v. 23, n. 7, p. 631–640, 1993.

Laporte, G.; Toth, P.; Vigo, D. Vehicle routing: historical perspective and recent contributions. *EURO Journal on Transportation and Logistics*, Springer, v. 2, n. 1-2, p. 1–4, 2013.

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

Letchford, A. N.; Salazar-González, J.-J. Projection results for vehicle routing. *Mathematical Programming*, Springer, v. 105, n. 2, p. 251–274, 2006.

Letchford, A. N.; Salazar-González, J.-J. Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. *European Journal of Operational Research*, Elsevier, v. 244, n. 3, p. 730–738, 2015.

Li, Y.; Chung, S. H. Disaster relief routing under uncertainty: A robust optimization approach. *IISE Transactions*, Taylor & Francis, v. 51, n. 8, p. 869–886, 2019. Disponível em: <https://doi.org/10.1080/24725854.2018.1450540>.

Lysgaard, J.; Letchford, A.; Eglese, R. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, v. 100, p. 423–445, 06 2004.

Mercier, A.; Soumis, F. An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research*, v. 34, p. 2251–2265, 08 2007.

Miller, C.; Tucker, A.; Zemlin, R. Integer programming formulation of traveling salesman problems. *Journal of Association for Computing Machinery*, Informs, v. 7, n. 1, p. 326–9, 1960.

Munari, P.; Alvarez, A. Aircraft routing for on-demand air transportation with service upgrade and maintenance events: Compact model and case study. *Journal of Air Transport Management*, v. 75, p. 75 – 84, 2019. ISSN 0969-6997. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0969699717303538>.

Munari, P. et al. The robust vehicle routing problem with time windows: Compact formulation and branch-price-and-cut method. *Transportation Science*, 2019.

Ordonez, F. Robust vehicle routing. In: ______. *Risk and Optimization in an Uncertain World*. [s.n.], 2010. cap. Chapter 7, p. 153–178. Disponível em: <http://pubsonline.informs.org/doi/abs/10.1287/educ.1100.0078>.

Oyola, J.; Arntzen, H.; Woodruff, D. The stochastic vehicle routing problem, a literature review, part i: models. *EURO Journal on Transportation and Logistics*, v. 7, 10 2016.

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

Pessoa, A. A.; Poss, M.; Vanderbeck, F.; Sadykov, R.; Vanderbeck, F. Branch-and-cut-and-price for the robust capacitated vehicle routing problem with knapsack uncertainty. *Operations Research*, INFORMS, 2020. Disponível em: <https://hal.inria.fr/hal-01958184>.

Poggi, M.; Uchoa, E. New exact algorithms for the capacitated vehicle routing problem. In: Toth, P.; Vigo, D. (Ed.). *Vehicle routing: Problems, methods, and applications*. [S.l.: s.n.], 2014, (MOS/SIAM Ser Optim). p. 59–86.

Rahbari, A.; Nasiri, M. M.; Werner, F.; Musavi, M.; Jolai, F. The vehicle routing and scheduling problem with cross-docking for perishable products under uncertainty: Two robust bi-objective models. *Applied Mathematical Modelling*, v. 70, p. 605 – 625, 2019. ISSN 0307-904X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0307904X19300770>.

Rahimian, H.; Mehrotra, S. *Distributionally Robust Optimization: A Review*. [S.l.]: arXiv, 2019.

Shebalov, S.; Klabjan, D. Robust airline crew pairing: Move-up crews. *Transportation Science*, v. 40, n. 3, p. 300–312, 2006.

Sherali, H. D.; Bish, E. K.; Zhu, X. Airline fleet assignment concepts, models, and algorithms. *European Journal of Operational Research*, v. 172, n. 1, p. 1–30, 2006.

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

Solyali, O.; Cordeau, J.-F.; Laporte, G. Robust inventory routing under demand uncertainty. *Transportation Science*, v. 46, n. 3, p. 327–340, 2012. Disponível em: <https://doi.org/10.1287/trsc.1110.0387>.

Soyster, A. L. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, INFORMS, v. 21, n. 5, p. 1154–1157, 1973. ISSN 0030364X, 15265463. Disponível em: <http://www.jstor.org/stable/168933>.

Subramanyam, A.; Repoussis, P. P.; Gounaris, C. E. Robust optimization of a broad class of heterogeneous vehicle routing problems under demand uncertainty. *INFORMS Journal on Computing*, v. 0, n. 0, p. 22, 2020. Disponível em: <https://doi.org/10.1287/ijoc.2019.0923>.

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

Tajik, N.; Tavakkoli-Moghaddam, R.; Vahdani, B.; Mousavi, S. A robust optimization approach for pollution routing problem with pickup and delivery under uncertainty. *Journal of Manufacturing Systems*, v. 33, 04 2014.

Toth, P.; Vigo, D. *The vehicle routing problem.* [S.l.]: SIAM, 2002.

Toth, P.; Vigo, D. *Vehicle Routing: Problems, Methods and Applications.* Second. [S.l.]: MOS-SIAM Series in Optimization, 2014.

Yang, W.; Karaesmen, I. Z.; Keskinocak, P.; Tayur, S. Aircraft and crew scheduling for fractional ownership programs. *Annals of Operations Research*, v. 159, n. 1, p. 415–431, 2008. ISSN 02545330.

Yao, Y.; Ergun, Ö.; Johnson, E.; Schultz, W.; Singleton, J. M. Strategic planning in fractional aircraft ownership programs. *European Journal of Operational Research*, v. 189, n. 2, p. 526–539, 2008. ISSN 03772217.

Zwan, F. M. Van der; Wils, K.; Ghijs, S. S. A. *Development of an Aircraft Routing System for an Air Taxi Operator.* [S.l.], 2011.

APPENDIX A – Robust formulations for the VRPTW using the 3-knapsack uncertainty set

A.1 MTZ-based formulation

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij}, \tag{A.1.1}$$

$$\text{s.t} \sum_{\substack{i=1\\ i\neq j}}^{n+1} x_{ij} = 1, \qquad j \in N^*, \tag{A.1.2}$$

$$\sum_{\substack{i=0\\ i\neq h}}^{n} x_{ih} = \sum_{\substack{j=1\\ j\neq h}}^{n+1} x_{hj}, \qquad h \in N^*, \tag{A.1.3}$$

$$u_{j\delta_1\delta_2\delta_3} \geq u_{i\delta_1\delta_2\delta_3} + \bar{d}_j + M(x_{ij}-1), \qquad i,j \in N, \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \delta_3 \leq \Delta_3^d \tag{A.1.4}$$

$$u_{j\delta_1\delta_2\delta_3} \geq u_{i\delta_1-\hat{d}_j\delta_2\delta_3} + \bar{d}_j + \hat{d}_j + M(x_{ij}-1),$$
$$i,j \in N, j \in S_1, j \notin S_2, \notin S_3, \hat{d}_j\delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \delta_3 \leq \Delta_3^d, \tag{A.1.5}$$

$$u_{j\delta_1\delta_2\delta_3} \geq u_{i\delta_1\delta_2-\hat{d}_j\delta_3} + \bar{d}_j + \hat{d}_j + M(x_{ij}-1),$$
$$i,j \in N, j \notin S_1, j \in S_2, j \notin S_3, \delta_1 \leq \Delta_1^d, \hat{d}_j \leq \delta_2 \leq \Delta_2^d, \delta_3 \leq \Delta_3^d, \tag{A.1.6}$$

$$u_{j\delta_1\delta_2\delta_3} \geq u_{i\delta_1\delta_2\delta_3-\hat{d}_j} + \bar{d}_j + \hat{d}_j + M(x_{ij}-1),$$
$$i,j \in N, j \notin S_1, j \notin S_2, j \in S_3, \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \hat{d}_j \leq \delta_3 \leq \Delta_3^d, \tag{A.1.7}$$

$$u_{j\delta_1\delta_2\delta_3} \geq u_{i\delta_1-\hat{d}_j\delta_2-\hat{d}_j\delta_3} + \bar{d}_j + \hat{d}_j + M(x_{ij}-1),$$
$$i,j \in N, j \in S_1, j \in S_2, j \notin S_3, \hat{d}_j \leq \delta_1 \leq \Delta_1^d, \hat{d}_j \leq \delta_2 \leq \Delta_2^d, \delta_3 \leq \Delta_3^d, \tag{A.1.8}$$

$$u_{j\delta_1\delta_2\delta_3} \geq u_{i\delta_1-\hat{d}_j\delta_2\delta_3-\hat{d}_j} + \bar{d}_j + \hat{d}_j + M(x_{ij}-1),$$
$$i,j \in N, j \in S_1, j \notin S_2, j \in S_3, \hat{d}_j \leq \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \hat{d}_j \leq \delta_3 \leq \Delta_3^d, \tag{A.1.9}$$

$$u_{j\delta_1\delta_2\delta_3} \geq u_{i\delta_1\delta_2-\hat{d}_j\delta_3-\hat{d}_j} + \bar{d}_j + \hat{d}_j + M(x_{ij}-1),$$
$$i,j \in N, j \notin S_1, j \in S_2, j \in S_3, \delta_1 \leq \Delta_1^d, \hat{d}_j \leq \delta_2 \leq \Delta_2^d, \hat{d}_j \leq \delta_3 \leq \Delta_3^d, \tag{A.1.10}$$

$$u_{j\delta_1\delta_2\delta_3} \geq u_{i\delta_1\delta_3-\hat{d}_j\delta_2-\hat{d}_j\delta_3-\hat{d}_j} + \bar{d}_j + \hat{d}_j + M(x_{ij}-1),$$
$$i,j \in N, j \in S_1, j \in S_2, j \in S_3, \hat{d}_j \leq \delta_1 \leq \Delta_1^d, \hat{d}_j \leq \delta_2 \leq \Delta_2^d, \hat{d}_j \leq \delta_3 \leq \Delta_3^d, \tag{A.1.11}$$

$$u_{j\Delta_1^d\delta_2} \geq u_{i\Delta_1^d-\gamma\delta_2\delta_3} + \bar{d}_j + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{d}_j, j \in S_1, j \notin S_2, j \notin S_3, \delta_2 \leq \Delta_2^d, \delta_3 \leq \Delta_3^d, \tag{A.1.12}$$

$$u_{j\Delta_1^d\delta_2} \geq u_{i\delta_1\Delta_2^d-\gamma\delta_3} + \bar{d}_j + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{d}_j, j \notin S_1, j \in S_2, j \notin S_3, \delta_1 \leq \Delta_1^d, \delta_3 \leq \Delta_3^d, \tag{A.1.13}$$

$$u_{j\Delta_1^d\delta_2} \geq u_{i\delta_1\delta_2\Delta_3^d-\gamma} + \bar{d}_j + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{d}_j, j \notin S_1, j \notin S_2, j \in S_3, \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \tag{A.1.14}$$

$$u_{j\Delta_1^d\delta_2} \geq u_{i\Delta_1^d-\gamma\Delta_2^d-\gamma\delta_3} + \bar{d}_j + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{d}_j, j \in S_1, j \in S_2, j \notin S_3, \delta_3 \leq \Delta_3^d, \tag{A.1.15}$$
$$u_{j\Delta_1^d\delta_2} \geq u_{i\Delta_1^d-\gamma\delta_2\Delta_3^d-\gamma} + \bar{d}_j + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{d}_j, j \in S_1, j \notin S_2, j \in S_3, \delta_2 \leq \Delta_2^d, \tag{A.1.16}$$
$$u_{j\Delta_1^d\delta_2} \geq u_{i\delta_1\Delta_2^d-\gamma\Delta_3^d-\gamma} + \bar{d}_j + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{d}_j, j \notin S_1, j \in S_2, j \in S_3, \delta_1 \leq \Delta_1^d, \tag{A.1.17}$$
$$u_{j\Delta_1^d\delta_2} \geq u_{i\Delta_1^d-\gamma\Delta_2^d-\gamma\Delta_3^d-\gamma} + \bar{d}_j + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{d}_j, j \in S_1, j \in S_2, j \in S_3, \tag{A.1.18}$$
$$u_{j\Delta_1^d\Delta_2^d\Delta_3^d} \leq Q, \qquad j \in N, \tag{A.1.19}$$
$$w_{j\delta_1\delta_2\delta_3} \geq w_{i\delta_1-\gamma\delta_2\delta_3} + \bar{t}_{ij} + s_i + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{t}_{ij}, j \in S_1, j \notin S_2, j \notin S_3, \gamma \leq \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t, \delta_3 \leq \Delta_3^t, \tag{A.1.20}$$
$$w_{j\delta_1\delta_2\delta_3} \geq w_{i\delta_1\delta_2-\gamma\delta_3} + \bar{t}_{ij} + s_i + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{t}_{ij}, j \notin S_1, j \in S_2, j \notin S_3, \delta_1 \leq \Delta_1^t, \gamma \leq \delta_2 \leq \Delta_2^t, \delta_3 \leq \Delta_3^t, \tag{A.1.21}$$
$$w_{j\delta_1\delta_2\delta_3} \geq w_{i\delta_1\delta_2\delta_3-\gamma} + \bar{t}_{ij} + s_i + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{t}_{ij}, j \notin S_1, j \notin S_2, j \in S_3, \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t, \gamma \leq \delta_3 \leq \Delta_3^t, \tag{A.1.22}$$
$$w_{j\delta_1\delta_2\delta_3} \geq w_{i\delta_1-\gamma\delta_2-\gamma\delta_3} + \bar{t}_{ij} + s_i + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{t}_{ij}, j \in S_1, j \in S_2, j \notin S_3, \gamma \leq \delta_1 \leq \Delta_1^t, \gamma \leq \delta_2 \leq \Delta_2^t, \delta_3 \leq \Delta_3^t, \tag{A.1.23}$$
$$w_{j\delta_1\delta_2\delta_3} \geq w_{i\delta_1-\gamma\delta_2\delta_3-\gamma} + \bar{t}_{ij} + s_i + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{t}_{ij}, j \in S_1, j \notin S_2, j \in S_3, \gamma \leq \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t, \gamma \leq \delta_3 \leq \Delta_3^t, \tag{A.1.24}$$
$$w_{j\delta_1\delta_2\delta_3} \geq w_{i\delta_1\delta_2-\gamma\delta_3-\gamma} + \bar{t}_{ij} + s_i + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{t}_{ij}, j \notin S_1, j \in S_2, j \in S_3, \delta_1 \leq \Delta_1^t, \gamma \leq \delta_2 \leq \Delta_2^t, \gamma \leq \delta_3 \leq \Delta_3^t, \tag{A.1.25}$$
$$w_{j\delta_1\delta_2\delta_3} \geq w_{i\delta_1-\gamma\delta_2-\gamma\delta_3-\gamma} + \bar{t}_{ij} + s_i + \gamma + M(x_{ij}-1),$$
$$i,j \in N, \gamma \leq \hat{t}_{ij}, j \in S_1, j \in S_2, j \in S_3, \gamma \leq \delta_1 \leq \Delta_1^t, \gamma \leq \delta_2 \leq \Delta_2^t, \gamma \leq \delta_3 \leq \Delta_3^t, \tag{A.1.26}$$
$$a_j \leq w_{j\delta_1\delta_2\delta_3} \leq b_j, \qquad i,j \in N, \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t, \delta_3 \leq \Delta_3^t, \tag{A.1.27}$$
$$x_{ij} \in \{0,1\}, \qquad i \in N \tag{A.1.28}$$
$$u_{i\delta_1\delta_2\delta_3} \geq 0, \qquad i \in N, \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \delta_3 \leq \delta_3 \tag{A.1.29}$$
$$w_{i\delta_1\delta_2\delta_3} \geq 0, \qquad i \in N, \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t, \delta_3 \leq \Delta_3^t. \tag{A.1.30}$$

A.2 CF formulation

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij}, \tag{A.2.1}$$
$$\text{s.t} \sum_{\substack{i=1 \\ i\neq j}}^{n+1} x_{ij} = 1, \qquad j \in N^*, \tag{A.2.2}$$

$$\sum_{\substack{i=0\\i\neq h}}^{n} x_{ih} = \sum_{\substack{j=1\\j\neq h}}^{n+1} x_{hj}, \qquad h \in N^*, \tag{A.2.3}$$

$$\sum_{h\in N} f_{hi\delta_1\delta_2\delta_3} - \sum_{j\in N} f_{ij\delta_1\delta_2\delta_3} \geq \bar{d}_i, \qquad i \in N, \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \delta_3 \leq \Delta_3^d, \tag{A.2.4}$$

$$\sum_{h\in N} f_{hi\delta_1\delta_2\delta_3} - \sum_{j\in N} f_{ij\delta_1-\hat{d}_i\delta_2\delta_3} \geq \bar{d}_i + \hat{d}_i,$$

$$i \in N, i \in S_1, i \notin S_2, i \notin S_3, \hat{d}_i \leq \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \delta_3 \leq \Delta_3^d, \tag{A.2.5}$$

$$\sum_{h\in N} f_{hi\delta_1\delta_2\delta_3} - \sum_{j\in N} f_{ij\delta_1\delta_2-\hat{d}_i\delta_3} \geq \bar{d}_i + \hat{d}_i,$$

$$i \in N, i \notin S_1, i \in S_2, i \notin S_3, \delta_1 \leq \Delta_1^d, \hat{d}_i \leq \delta_2 \leq \Delta_2^d, \delta_3 \leq \Delta_3^d, \tag{A.2.6}$$

$$\sum_{h\in N} f_{hi\delta_1\delta_2\delta_3} - \sum_{j\in N} f_{ij\delta_1\delta_2\delta_3-\hat{d}_i} \geq \bar{d}_i + \hat{d}_i,$$

$$i \in N, i \notin S_1, i \notin S_2, i \in S_3, \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \hat{d}_i \leq \delta_3 \leq \Delta_3^d, \tag{A.2.7}$$

$$\sum_{h\in N} f_{hi\delta_1\delta_2\delta_3} - \sum_{j\in N} f_{ij\delta_1-\hat{d}_i\delta_2-\hat{d}_i\delta_3} \geq \bar{d}_i + \hat{d}_i,$$

$$i \in N, i \in S_1, i \in S_2, i \notin S_3, \hat{d}_i \leq \delta_1 \leq \Delta_1^d, \hat{d}_i \leq \delta_2 \leq \Delta_2^d, \delta_3 \leq \Delta_3^d, \tag{A.2.8}$$

$$\sum_{h\in N} f_{hi\delta_1\delta_2\delta_3} - \sum_{j\in N} f_{ij\delta_1-\hat{d}_i\delta_2\delta_3-\hat{d}_i} \geq \bar{d}_i + \hat{d}_i,$$

$$i \in N, i \in S_1, i \notin S_2, i \in S_3, \hat{d}_i \leq \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \hat{d}_i \leq \delta_3 \leq \Delta_3^d, \tag{A.2.9}$$

$$\sum_{h\in N} f_{hi\delta_1\delta_2\delta_3} - \sum_{j\in N} f_{ij\delta_1\delta_2-\hat{d}_i\delta_3-\hat{d}_i} \geq \bar{d}_i + \hat{d}_i,$$

$$i \in N, i \notin S_1, i \in S_2, i \in S_3, \delta_1 \leq \Delta_1^d, \hat{d}_i \leq \delta_2 \leq \Delta_2^d, \hat{d}_i \leq \delta_3 \leq \Delta_3^d, \tag{A.2.10}$$

$$\sum_{h\in N} f_{hi\delta_1\delta_2\delta_3} - \sum_{j\in N} f_{ij\delta_1-\hat{d}_i\delta_2-\hat{d}_i\delta_3-\hat{d}_i} \geq \bar{d}_i + \hat{d}_i,$$

$$i \in N, i \in S_1, i \in S_2, i \in S_3, \hat{d}_i \leq \delta_1 \leq \Delta_1^d, \hat{d}_i \leq \delta_2 \leq \Delta_2^d, \hat{d}_i \leq \delta_3 \leq \Delta_3^d, \tag{A.2.11}$$

$$\sum_{h\in N} f_{hi\Delta_1^d\delta_2\delta_3} - \sum_{j\in N} f_{ij\Delta_1^d-\gamma\delta_2\delta_3} \geq \bar{d}_i + \gamma, i$$

$$i \in N, i \in S_1, i \notin S_2, i \notin S_3, \delta_2 \leq \Delta_2^d, \delta_3 \leq \Delta_3^d, \gamma < \hat{d}_i, \tag{A.2.12}$$

$$\sum_{h\in N} f_{hi\delta_1\Delta_2^d\delta_3} - \sum_{j\in N} f_{ij\delta_1\Delta_2^d-\gamma\delta_3} \geq \bar{d}_i + \gamma, i$$

$$i \in N, i \notin S_1, i \in S_2, i \notin S_3, \delta_1 \leq \Delta_1^d, \delta_3 \leq \Delta_3^d, \gamma < \hat{d}_i, \tag{A.2.13}$$

$$\sum_{h\in N} f_{hi\delta_1\delta_2\Delta_3^d} - \sum_{j\in N} f_{ij\delta_1\delta_2\Delta_3^d-\gamma} \geq \bar{d}_i + \gamma, i$$

$$i \in N, i \notin S_1, i \notin S_2, i \in S_3, \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \gamma < \hat{d}_i, \tag{A.2.14}$$

$$\sum_{h\in N} f_{hi\Delta_1^d\Delta_2^d\delta_3} - \sum_{j\in N} f_{ij\Delta_1^d-\gamma\Delta_2^d-\gamma\delta_3} \geq \bar{d}_i + \gamma, i$$

$$i \in N, i \in S_1, i \in S_2, i \notin S_3, \delta_3 \leq \Delta_3^d, \gamma < \hat{d}_i, \tag{A.2.15}$$

$$\sum_{h\in N} f_{hi\Delta_1^d\delta_2\Delta_3^d} - \sum_{j\in N} f_{ij\Delta_1^d-\gamma\delta_2\Delta_3^d-\gamma} \geq \bar{d}_i + \gamma, i$$

$$i \in N, i \in S_1, i \notin S_2, i \in S_3, \delta_2 \leq \Delta_2^d, \gamma < \hat{d}_i, \tag{A.2.16}$$

$$\sum_{h\in N} f_{hi\delta_1\Delta_2^d\Delta_3^d} - \sum_{j\in N} f_{ij\delta_1\Delta_2^d-\gamma\Delta_3^d-\gamma} \geq \bar{d}_i + \gamma, i$$

$$i \in N, i \notin S_1, i \in S_2, i \in S_3, \delta_1 \leq \Delta_1^d, \gamma < \hat{d}_i, \tag{A.2.17}$$

$$\sum_{h \in N} f_{hi\Delta_1^d\Delta_2^d\Delta_3^d} - \sum_{j \in N} f_{ij\Delta_1^d-\gamma\Delta_2^d-\gamma\Delta_3^d-\gamma} \geq \bar{d}_i + \gamma, i$$
$$i \in N, i \notin S_1, i \in S_2, i \in S_3, \gamma < \hat{d}_i, \tag{A.2.18}$$

$$f_{ij\delta_1\delta_2\delta_3} \leq Qx_{ij}, \qquad i, j \in N, \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \delta_3 \leq \Delta_3^d, \tag{A.2.19}$$

$$\sum_{j \in N} g_{ij\delta_1\delta_2\delta_3} \geq \sum_{\substack{h \in N \\ \gamma \leq \hat{t}_{hi}}} (g_{hi\delta_1-\gamma\delta_2\delta_3} + (\bar{t}_{hi} + \gamma)x_{hi}) + s_i,$$
$$i \in N, i \in S_1, i \notin S_2, i \notin S_3, \gamma \leq \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t, \delta_3 \leq \Delta_3^t, \tag{A.2.20}$$

$$\sum_{j \in N} g_{ij\delta_1\delta_2\delta_3} \geq \sum_{\substack{h \in N \\ \gamma \leq \hat{t}_{hi}}} (g_{hi\delta_1\delta_2-\gamma\delta_3} + (\bar{t}_{hi} + \gamma)x_{hi}) + s_i,$$
$$i \in N, i \notin S_1, i \in S_2, i \notin S_3, \delta_1 \leq \Delta_1^t, \gamma \leq \delta_2 \leq \Delta_2^t, \delta_3 \leq \Delta_3^t, \tag{A.2.21}$$

$$\sum_{j \in N} g_{ij\delta_1\delta_2\delta_3} \geq \sum_{\substack{h \in N \\ \gamma \leq \hat{t}_{hi}}} (g_{hi\delta_1\delta_2\delta_3-\gamma} + (\bar{t}_{hi} + \gamma)x_{hi}) + s_i,$$
$$i \in N, i \notin S_1, i \notin S_2, i \in S_3, \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t, \gamma \leq \delta_3 \leq \Delta_3^t, \tag{A.2.22}$$

$$\sum_{j \in N} g_{ij\delta_1\delta_2\delta_3} \geq \sum_{\substack{h \in N \\ \gamma \leq \hat{t}_{hi}}} (g_{hi\delta_1-\gamma\delta_2-\gamma\delta_3} + (\bar{t}_{hi} + \gamma)x_{hi}) + s_i,$$
$$i \in N, i \in S_1, i \in S_2, i \notin S_3, \gamma \leq \delta_1 \leq \Delta_1^t, \gamma \leq \delta_2 \leq \Delta_2^t, \delta_3 \leq \Delta_3^t, \tag{A.2.23}$$

$$\sum_{j \in N} g_{ij\delta_1\delta_2\delta_3} \geq \sum_{\substack{h \in N \\ \gamma \leq \hat{t}_{hi}}} (g_{hi\delta_1-\gamma\delta_2\delta_3-\gamma} + (\bar{t}_{hi} + \gamma)x_{hi}) + s_i,$$
$$i \in N, i \in S_1, i \notin S_2, i \in S_3, \gamma \leq \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t, \gamma \leq \delta_3 \leq \Delta_3^t, \tag{A.2.24}$$

$$\sum_{j \in N} g_{ij\delta_1\delta_2\delta_3} \geq \sum_{\substack{h \in N \\ \gamma \leq \hat{t}_{hi}}} (g_{hi\delta_1\delta_2-\gamma\delta_3-\gamma} + (\bar{t}_{hi} + \gamma)x_{hi}) + s_i,$$
$$i \in N, i \notin S_1, iinS_2, i \in S_3, \delta_1 \leq \Delta_1^t, \gamma \leq \delta_2 \leq \Delta_2^t, \gamma \leq \delta_3 \leq \Delta_3^t, \tag{A.2.25}$$

$$\sum_{j \in N} g_{ij\delta_1\delta_2\delta_3} \geq \sum_{\substack{h \in N \\ \gamma \leq \hat{t}_{hi}}} (g_{hi\delta_1-\gamma\delta_2-\gamma\delta_3-\gamma} + (\bar{t}_{hi} + \gamma)x_{hi}) + s_i,$$
$$i \in N, i \in S_1, i \in S_2, i \in S_3, \gamma \leq \delta_1 \leq \Delta_1^t, \gamma \leq \delta_2 \leq \Delta_2^t, \gamma \leq \delta_3 \leq \Delta_3^t, \tag{A.2.26}$$

$$(a_j + s_j)x_{ij} \leq g_{ij\delta_1\delta_2\delta_3} \leq (b_j + s_j)x_{ij}, \qquad i, j \in N, \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t, \delta_3 \leq \Delta_3^t, \tag{A.2.27}$$

$$x_{ij} \in \{0, 1\}, \qquad (i, j) \in A, , \tag{A.2.28}$$

$$f_{ij\delta_1\delta_2\delta_3} \geq 0, \qquad (i, j) \in A, \delta_1 \leq \Delta_1^d, \delta_2 \leq \Delta_2^d, \delta_3 \leq \Delta_3^d, , \tag{A.2.29}$$

$$g_{ij\delta_1\delta_2\delta_3} \geq 0, \qquad (i, j) \in A, \delta_1 \leq \Delta_1^t, \delta_2 \leq \Delta_2^t, \delta_3 \leq \Delta_3^t. \tag{A.2.30}$$

APPENDIX B – Results from the compact models for the literature instances

B.1 Cardinality-Constrained Uncertainty set

In this section, we present in more details the results of our computational experiments in instances of the literature for the cardinality constrained uncertainty set using the compact models. We created tables for each model (MTZ and CF) showing the relevant information for each set of instances separately. In these tables, for every combination of Γ^t, Γ^d and deviation (*Dev*) studied, we present the average solution value (*Sol*) and the time taken to solve the problem (*Time*) of both the linear relaxation and integer problem, and the Price-of-Robustness (*PoR*), which represents how much the robust solution costs more than the deterministic solution, expressed in percentage. To better evaluate the computational results, we also present the percentage gap between the best solution obtained in the time limit and its lower bound (*Gap*), if the solution is optimal this difference is zero. Finally, we also present the number of instances (*Ins*) evaluated in that combination of Γ^q, Γ^t and deviations, how many of those were solved to optimality by the model within the time limit (*Opt*) and the number of infeasible instances (*Inf*) in that group.

B.1.1 MTZ-based formulation

Table 30 – Average results from the MTZ formulation for the RVRPTW instances of class C1

C1												
				Linear Relaxation		**Integer Solution**						
Γ	***Dev^d***	**Γ**	***Dev^t***	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	76.75	0.011	190.59	419.28	0.0%	0.8%	9	8	0
1	10%	0	0%	76.75	0.010	190.97	441.69	0.2%	1.9%	9	8	0
1	25%	0	0%	76.75	0.011	190.97	466.34	0.2%	0.9%	9	8	0
1	50%	0	0%	76.75	0.019	226.10	826.58	18.6%	3.0%	9	7	0
5	10%	0	0%	76.75	0.020	226.10	924.74	18.6%	5.0%	9	7	0
5	25%	0	0%	76.75	0.019	235.10	979.01	23.4%	5.0%	9	7	0
5	50%	0	0%	76.75	0.030	254.57	1364.81	33.6%	7.2%	9	6	0
10	10%	0	0%	76.75	0.032	226.10	956.45	18.6%	4.9%	9	7	0
10	25%	0	0%	76.75	0.035	252.91	1662.15	32.7%	10.1%	9	5	0
10	50%	0	0%	76.76	0.010	256.53	1662.17	34.6%	10.6%	9	5	0
0	0%	1	10%	76.76	0.011	190.59	444.49	0.0%	0.9%	9	8	0
0	0%	1	25%	76.78	0.011	190.59	459.81	0.0%	1.8%	9	8	0
0	0%	1	50%	76.77	0.022	195.84	434.27	2.7%	0.8%	9	8	0
0	0%	5	10%	76.78	0.022	190.59	532.16	0.0%	2.0%	9	8	0
0	0%	5	25%	76.81	0.022	190.59	519.16	0.0%	1.3%	9	8	0
0	0%	5	50%	76.77	0.041	206.30	547.15	8.2%	1.2%	9	8	0
0	0%	10	10%	76.79	0.041	190.59	657.14	0.0%	2.0%	9	8	0
0	0%	10	25%	76.83	0.043	190.59	766.38	0.0%	2.1%	9	8	0
0	0%	10	50%	76.76	0.012	206.30	695.62	8.2%	1.6%	9	8	0
1	10%	1	10%	76.76	0.01	190.97	448.43	0.00	1.2%	9	8	0
1	25%	1	25%	76.78	0.01	190.97	461.41	0.00	1.1%	9	8	0
1	50%	1	50%	76.77	0.04	226.66	851.16	0.19	3.0%	9	7	0
5	10%	5	10%	76.78	0.04	226.32	1217.40	0.19	5.0%	9	6	0
5	25%	5	25%	76.81	0.04	235.50	1138.43	0.24	5.1%	9	7	0
5	50%	5	50%	76.77	0.08	255.52	1339.80	0.34	9.5%	9	6	0
10	10%	10	10%	76.79	0.09	226.53	1257.82	0.19	7.2%	9	6	0
10	25%	10	25%	76.83	0.08	253.07	1759.80	0.33	11.0%	9	5	0
10	50%	10	50%	84.15	0.00	257.60	1713.04	0.35	12.4%	9	5	0
All				77.04	0.03	216.61	890.95	0.14	4.2%	252	198	0

Table 31 – Average results from the MTZ formulation for the RVRPTW instances of class R1

R1												
				Linear Relaxation		**Integer Solution**						
Γ	***Dev^d***	**Γ**	***Dev^t***	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	290.11	0.011	463.37	128.72	0.0%	0.0%	12	12	0
1	10%	0	0%	290.11	0.011	463.37	269.55	0.0%	0.0%	12	12	0
1	25%	0	0%	290.11	0.010	463.37	249.41	0.0%	0.0%	12	12	0
1	50%	0	0%	290.11	0.019	463.37	230.45	0.0%	0.0%	12	12	0
5	10%	0	0%	290.11	0.019	463.37	418.38	0.0%	0.8%	12	11	0
5	25%	0	0%	290.11	0.019	463.37	417.42	0.0%	0.9%	12	11	0
5	50%	0	0%	290.11	0.036	463.37	417.79	0.0%	0.5%	12	11	0
10	10%	0	0%	290.11	0.033	463.37	626.57	0.0%	0.8%	12	11	0
10	25%	0	0%	290.11	0.036	463.37	667.29	0.0%	0.7%	12	11	0
10	50%	0	0%	290.27	0.012	463.37	650.56	0.0%	0.8%	12	11	0
0	0%	1	10%	290.50	0.011	466.89	223.91	0.8%	0.0%	12	12	0
0	0%	1	25%	290.96	0.011	477.74	152.68	3.0%	0.0%	12	12	0
0	0%	1	50%	290.49	0.025	464.38	83.10	5.1%	0.0%	12	8	4
0	0%	5	10%	291.06	0.026	469.31	399.64	1.3%	0.6%	12	11	0
0	0%	5	25%	292.14	0.028	481.78	337.84	3.9%	0.7%	12	11	0
0	0%	5	50%	290.54	0.049	486.99	309.74	10.3%	1.0%	12	7	4
0	0%	10	10%	291.20	0.049	470.74	512.44	1.7%	1.2%	12	11	0
0	0%	10	25%	292.52	0.053	482.74	393.83	4.1%	1.0%	12	11	0
0	0%	10	50%	290.27	0.014	488.18	327.78	10.6%	1.5%	12	7	4
1	10%	1	10%	290.50	0.01	466.89	265.08	0.01	0.0%	12	12	0
1	25%	1	25%	290.96	0.01	477.74	282.02	0.03	0.0%	12	12	0
1	50%	1	50%	290.49	0.04	464.38	102.22	0.05	0.0%	12	8	4
5	10%	5	10%	291.06	0.04	469.31	476.71	0.01	1.0%	12	11	0
5	25%	5	25%	292.14	0.05	482.32	397.25	0.04	0.9%	12	11	0
5	50%	5	50%	290.54	0.10	487.64	335.35	0.11	1.9%	12	7	4
10	10%	10	10%	291.20	0.10	470.18	880.85	0.02	1.8%	12	10	0
10	25%	10	25%	292.52	0.10	482.32	624.16	0.04	1.5%	12	11	0
10	50%	10	50%	286.63	0.00	488.79	365.42	0.11	2.5%	12	7	4
All				290.61	0.03	471.86	376.65	0.03	0.7%	336	293	24

Table 32 – Average results from the MTZ formulation for the RVRPTW instances of class RC1

RC1												
				Linear Relaxation		**Integer Solution**						
Γ	***Dev^d***	**Γ**	***Dev^t***	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	95.04	0.011	350.24	1802.80	0.0%	10.6%	8	4	0
1	10%	0	0%	95.04	0.010	350.24	1802.48	0.0%	10.2%	8	4	0
1	25%	0	0%	95.04	0.010	350.24	1802.13	0.0%	10.7%	8	4	0
1	50%	0	0%	95.04	0.019	379.94	1904.19	9.4%	18.8%	8	4	0
5	10%	0	0%	95.04	0.020	380.36	1936.52	9.5%	19.3%	8	4	0
5	25%	0	0%	95.04	0.019	455.99	3070.53	32.0%	29.3%	8	2	0
5	50%	0	0%	95.04	0.034	461.11	3132.86	33.4%	28.6%	8	2	0
10	10%	0	0%	95.04	0.035	379.94	1992.88	9.4%	23.4%	8	4	0
10	25%	0	0%	95.04	0.035	456.73	3093.38	32.2%	29.9%	8	2	0
10	50%	0	0%	95.06	0.011	480.80	3150.44	39.6%	33.6%	8	1	0
0	0%	1	10%	95.08	0.011	350.91	1643.99	0.1%	9.8%	8	5	0
0	0%	1	25%	95.11	0.011	352.91	1503.28	0.7%	8.4%	8	5	0
0	0%	1	50%	95.06	0.021	368.19	1235.43	7.1%	7.0%	8	5	1
0	0%	5	10%	95.08	0.019	352.58	1802.92	0.6%	18.0%	8	4	0
0	0%	5	25%	95.13	0.020	377.04	1519.10	7.1%	9.4%	8	5	0
0	0%	5	50%	95.06	0.039	425.04	1388.24	24.6%	13.3%	8	4	1
0	0%	10	10%	95.09	0.042	352.60	1803.70	0.7%	16.8%	8	4	0
0	0%	10	25%	95.14	0.040	382.59	1594.29	8.6%	12.8%	8	5	0
0	0%	10	50%	95.06	0.012	450.13	2255.03	32.8%	18.6%	8	2	1
1	10%	1	10%	95.08	0.01	350.91	1802.16	0.00	13.3%	8	4	0
1	25%	1	25%	95.11	0.01	352.91	1460.91	0.01	9.5%	8	5	0
1	50%	1	50%	95.06	0.04	406.50	1605.99	0.19	12.8%	8	4	1
5	10%	5	10%	95.08	0.04	385.69	1964.36	0.11	19.9%	8	4	0
5	25%	5	25%	95.13	0.04	466.53	2955.14	0.35	27.8%	8	2	0
5	50%	5	50%	95.06	0.08	486.46	2036.16	0.44	22.1%	8	3	1
10	10%	10	10%	95.09	0.08	388.38	2118.40	0.12	22.9%	8	4	0
10	25%	10	25%	95.14	0.08	472.25	3150.23	0.37	31.2%	8	1	0
10	50%	10	50%	95.17	0.00	489.37	2464.13	0.45	26.5%	8	2	1
All				95.08	0.03	402.02	2071.13	0.16	18.4%	224	99	6

Table 33 – Average results from the MTZ formulation for the RVRPTW instances of class C2

C2												
				Linear Relaxation		**Integer Solution**						
Γ	***Dev^d***	**Γ**	***Dev^t***	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	144.57	0.010	214.45	13.28	0.0%	0.0%	8	8	0
1	10%	0	0%	144.57	0.010	214.45	8.75	0.0%	0.0%	8	8	0
1	25%	0	0%	144.57	0.010	214.45	17.26	0.0%	0.0%	8	8	0
1	50%	0	0%	144.57	0.018	214.45	25.59	0.0%	0.0%	8	8	0
5	10%	0	0%	144.57	0.019	214.45	25.56	0.0%	0.0%	8	8	0
5	25%	0	0%	144.57	0.018	214.45	65.01	0.0%	0.0%	8	8	0
5	50%	0	0%	144.57	0.031	214.45	66.76	0.0%	0.0%	8	8	0
10	10%	0	0%	144.57	0.032	214.45	136.57	0.0%	0.0%	8	8	0
10	25%	0	0%	144.57	0.035	214.45	99.75	0.0%	0.0%	8	8	0
10	50%	0	0%	144.57	0.010	214.45	84.05	0.0%	0.0%	8	8	0
0	0%	1	10%	144.58	0.011	214.45	15.99	0.0%	0.0%	8	8	0
0	0%	1	25%	144.58	0.010	214.45	16.29	0.0%	0.0%	8	8	0
0	0%	1	50%	144.57	0.018	214.63	21.15	0.1%	0.0%	8	8	0
0	0%	5	10%	144.58	0.018	214.45	255.45	0.0%	0.0%	8	8	0
0	0%	5	25%	144.59	0.020	214.63	82.65	0.1%	0.0%	8	8	0
0	0%	5	50%	144.58	0.037	214.63	155.87	0.1%	0.0%	8	8	0
0	0%	10	10%	144.58	0.035	214.45	462.98	0.0%	0.8%	8	7	0
0	0%	10	25%	144.60	0.036	214.63	180.64	0.1%	0.0%	8	8	0
0	0%	10	50%	144.57	0.012	214.63	91.81	0.1%	0.0%	8	8	0
1	10%	1	10%	144.58	0.01	214.45	14.94	0.00	0.0%	8	8	0
1	25%	1	25%	144.58	0.01	214.45	42.23	0.00	0.0%	8	8	0
1	50%	1	50%	144.57	0.03	214.63	15.49	0.00	0.0%	8	8	0
5	10%	5	10%	144.58	0.03	214.45	111.75	0.00	0.0%	8	8	0
5	25%	5	25%	144.59	0.03	214.63	461.90	0.00	1.0%	8	7	0
5	50%	5	50%	144.58	0.08	214.63	115.17	0.00	0.0%	8	8	0
10	10%	10	10%	144.58	0.07	214.45	193.15	0.00	0.0%	8	8	0
10	25%	10	25%	144.60	0.07	214.63	464.35	0.00	1.4%	8	7	0
10	50%	10	50%	166.74	0.00	214.63	322.33	0.00	0.0%	8	8	0
All				145.37	0.03	214.51	127.38	0.00	0.1%	224	221	0

Table 34 – Average results from the MTZ formulation for the RVRPTW instances of class R2

						R2						
				Linear Relaxation		**Integer Solution**						
Γ	***Dev***^{d}	**Γ**	***Dev***^{t}	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	279.82	0.011	382.15	41.89	0.0%	0.0%	11	11	0
1	10%	0	0%	279.82	0.010	382.15	52.97	0.0%	0.0%	11	11	0
1	25%	0	0%	279.82	0.010	382.15	46.32	0.0%	0.0%	11	11	0
1	50%	0	0%	279.82	0.019	382.15	47.22	0.0%	0.0%	11	11	0
5	10%	0	0%	279.82	0.020	382.15	180.29	0.0%	0.0%	11	11	0
5	25%	0	0%	279.82	0.020	382.15	171.29	0.0%	0.0%	11	11	0
5	50%	0	0%	279.82	0.034	382.15	135.54	0.0%	0.0%	11	11	0
10	10%	0	0%	279.82	0.033	382.15	356.32	0.0%	0.0%	11	11	0
10	25%	0	0%	279.82	0.033	382.15	423.91	0.0%	0.0%	11	11	0
10	50%	0	0%	279.83	0.011	382.15	368.22	0.0%	0.0%	11	11	0
0	0%	1	10%	279.85	0.011	382.15	48.50	0.0%	0.0%	11	11	0
0	0%	1	25%	279.88	0.010	383.55	67.03	0.4%	0.0%	11	11	0
0	0%	1	50%	279.83	0.020	385.88	43.99	0.9%	0.0%	11	11	0
0	0%	5	10%	279.85	0.020	383.55	143.83	0.4%	0.0%	11	11	0
0	0%	5	25%	279.88	0.021	383.73	161.54	0.5%	0.0%	11	11	0
0	0%	5	50%	279.83	0.041	387.14	138.75	1.3%	0.0%	11	11	0
0	0%	10	10%	279.85	0.039	383.55	351.87	0.4%	0.0%	11	11	0
0	0%	10	25%	279.88	0.040	383.98	292.54	0.5%	0.0%	11	11	0
0	0%	10	50%	279.83	0.013	387.14	343.75	1.3%	0.0%	11	11	0
1	10%	1	10%	279.85	0.01	382.15	58.98914	0.00	0.0%	11	11	0
1	25%	1	25%	279.88	0.01	383.55	63.11	0.00	0.0%	11	11	0
1	50%	1	50%	279.83	0.03	385.88	74.50	0.01	0.0%	11	11	0
5	10%	5	10%	279.85	0.04	383.55	442.51	0.00	0.0%	11	11	0
5	25%	5	25%	279.88	0.04	383.73	280.99	0.00	0.0%	11	11	0
5	50%	5	50%	279.83	0.08	387.14	490.56	0.01	0.0%	11	11	0
10	10%	10	10%	279.85	0.08	383.63	1050.34	0.00	0.3%	11	10	0
10	25%	10	25%	279.88	0.08	383.98	976.03	0.01	0.2%	11	10	0
10	50%	10	50%	262.45	0.00	387.14	1066.62	0.01	0.0%	11	11	0
All				279.22	0.03	383.67	282.84	0.00	0.0%	308	306	0

Table 35 – Average results from the MTZ formulation for the RVRPTW instances of class RC2

						RC2						
				Linear Relaxation		**Integer Solution**						
Γ	***Dev***^{d}	**Γ**	***Dev***^{t}	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	94.78	0.010	319.28	1500.82	0.0%	14.1%	8	5	0
1	10%	0	0%	94.78	0.010	319.28	1686.56	0.0%	14.3%	8	5	0
1	25%	0	0%	94.78	0.010	319.28	1791.88	0.0%	15.9%	8	5	0
1	50%	0	0%	94.78	0.020	319.28	1696.77	0.0%	14.7%	8	5	0
5	10%	0	0%	94.78	0.019	319.28	1859.49	0.0%	16.3%	8	4	0
5	25%	0	0%	94.78	0.019	319.28	1848.11	0.0%	16.8%	8	4	0
5	50%	0	0%	94.78	0.035	319.28	1859.37	0.0%	17.5%	8	4	0
10	10%	0	0%	94.78	0.035	319.34	1932.23	0.0%	19.6%	8	4	0
10	25%	0	0%	94.78	0.035	319.35	1928.81	0.0%	18.5%	8	4	0
10	50%	0	0%	94.78	0.010	319.34	1916.45	0.0%	19.5%	8	4	0
0	0%	1	10%	94.79	0.010	319.28	1810.25	0.0%	16.6%	8	4	0
0	0%	1	25%	94.79	0.011	319.75	1671.99	0.2%	13.4%	8	5	0
0	0%	1	50%	94.78	0.020	320.03	1514.86	0.3%	13.5%	8	5	0
0	0%	5	10%	94.79	0.019	319.66	1816.68	0.1%	17.7%	8	4	0
0	0%	5	25%	94.79	0.019	319.76	1833.63	0.2%	17.6%	8	4	0
0	0%	5	50%	94.78	0.038	320.59	1814.93	0.5%	15.9%	8	4	0
0	0%	10	10%	94.79	0.037	319.83	1864.38	0.2%	17.9%	8	4	0
0	0%	10	25%	94.79	0.036	319.84	1855.87	0.2%	19.0%	8	4	0
0	0%	10	50%	94.78	0.012	320.54	1853.01	0.5%	18.9%	8	4	0
1	10%	1	10%	94.79	0.01	319.28	1826.77	0.00	16.5%	8	4	0
1	25%	1	25%	94.79	0.01	319.50	1821.26	0.00	16.9%	8	4	0
1	50%	1	50%	94.78	0.03	319.89	1820.14	0.00	15.6%	8	4	0
5	10%	5	10%	94.79	0.03	319.88	1882.37	0.00	18.6%	8	4	0
5	25%	5	25%	94.79	0.04	319.89	1846.80	0.00	18.8%	8	4	0
5	50%	5	50%	94.78	0.08	320.05	1863.04	0.00	19.8%	8	4	0
10	10%	10	10%	94.79	0.08	319.85	2101.51	0.00	19.3%	8	4	0
10	25%	10	25%	94.79	0.07	320.40	2088.55	0.00	19.1%	8	4	0
10	50%	10	50%	82.71	0.00	320.90	1959.34	0.01	19.8%	8	4	0
All				94.35	0.03	319.71	1830.92	0.00	17.2%	224	118	0

B.1.2 CF formulation

Table 36 – Average results from the CF formulation for the RVRPTW instances of class C1

C1												
				Linear Relaxation		**Integer Solution**						
Γ	***Dev^d***	**Γ**	***Dev^t***	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	177.96	0.030	190.59	0.61	0.0%	0.0%	9	9	0
1	10%	0	0%	178.31	0.055	190.97	2.35	0.2%	0.0%	9	9	0
1	25%	0	0%	178.97	0.057	190.97	1.26	0.2%	0.0%	9	9	0
1	50%	0	0%	180.38	0.055	226.10	9.58	18.6%	0.0%	9	9	0
5	10%	0	0%	178.96	0.168	226.10	220.66	18.6%	0.0%	9	9	0
5	25%	0	0%	181.73	0.206	234.84	845.79	23.2%	0.4%	9	8	0
5	50%	0	0%	187.54	0.224	254.68	2004.90	33.6%	4.7%	9	4	0
10	10%	0	0%	179.52	0.418	226.10	856.80	18.6%	1.1%	9	8	0
10	25%	0	0%	184.22	0.447	253.09	2048.65	32.8%	7.5%	9	4	0
10	50%	0	0%	195.80	0.647	257.52	2028.48	35.1%	7.6%	9	4	0
0	0%	1	10%	177.98	0.077	190.59	1.82	0.0%	0.0%	9	9	0
0	0%	1	25%	178.00	0.067	190.59	1.54	0.0%	0.0%	9	9	0
0	0%	1	50%	178.07	0.077	195.84	1.21	2.7%	0.0%	9	9	0
0	0%	5	10%	177.99	0.255	190.59	8.06	0.0%	0.0%	9	9	0
0	0%	5	25%	178.05	0.264	190.59	9.16	0.0%	0.0%	9	9	0
0	0%	5	50%	178.35	0.275	206.30	10.21	8.2%	0.0%	9	9	0
0	0%	10	10%	178.00	0.964	190.59	57.64	0.0%	0.0%	9	9	0
0	0%	10	25%	178.07	0.831	190.59	51.45	0.0%	0.0%	9	9	0
0	0%	10	50%	178.46	0.786	206.30	75.17	8.2%	0.0%	9	9	0
1	10%	1	10%	178.32	0.09	190.97	1.68	0.00	0.0%	9	9	0
1	25%	1	25%	179.00	0.09	190.97	2.46	0.00	0.0%	9	9	0
1	50%	1	50%	180.42	0.09	226.66	14.40	0.19	0.0%	9	9	0
5	10%	5	10%	178.98	0.43	226.32	447.74	0.19	0.0%	9	9	0
5	25%	5	25%	181.76	0.46	235.24	1391.84	0.23	0.6%	9	7	0
5	50%	5	50%	187.61	0.46	255.50	2017.26	0.34	5.6%	9	4	0
10	10%	10	10%	179.55	2.93	229.26	1486.52	0.20	6.2%	9	6	0
10	25%	10	25%	184.29	3.05	253.94	2149.61	0.33	12.2%	9	4	0
10	50%	10	50%	195.89	3.47	259.67	2094.90	0.36	11.5%	9	4	0
All				181.15	0.61	216.84	637.20	0.14	2.0%	252	215	0

Table 37 – Average results from the CF formulation for the RVRPTW instances of class R1

R1												
				Linear Relaxation		**Integer Solution**						
Γ	***Dev^d***	**Γ**	***Dev^t***	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	396.37	0.027	463.37	30.99	0.0%	0.0%	12	12	0
1	10%	0	0%	396.37	0.052	463.37	50.02	0.0%	0.0%	12	12	0
1	25%	0	0%	396.37	0.061	463.37	37.81	0.0%	0.0%	12	12	0
1	50%	0	0%	396.37	0.060	463.37	50.52	0.0%	0.0%	12	12	0
5	10%	0	0%	396.37	0.179	463.37	380.87	0.0%	0.0%	12	12	0
5	25%	0	0%	396.37	0.205	463.37	261.67	0.0%	0.0%	12	12	0
5	50%	0	0%	396.37	0.194	463.37	251.17	0.0%	0.0%	12	12	0
10	10%	0	0%	396.37	0.419	463.37	1016.84	0.0%	0.3%	12	11	0
10	25%	0	0%	396.37	0.370	463.37	1050.72	0.0%	0.5%	12	10	0
10	50%	0	0%	396.37	0.478	463.62	1078.63	0.1%	0.4%	12	11	0
0	0%	1	10%	396.72	0.095	466.89	64.03	0.8%	0.0%	12	12	0
0	0%	1	25%	397.45	0.086	475.28	141.55	2.4%	0.0%	12	12	0
0	0%	1	50%	310.62	0.103	464.38	47.07	5.1%	0.0%	12	8	4
0	0%	5	10%	397.73	0.283	469.31	1041.32	1.3%	0.5%	12	10	0
0	0%	5	25%	400.71	0.311	480.51	1340.28	3.6%	0.6%	12	10	0
0	0%	5	50%	318.68	0.321	483.98	1183.89	9.6%	1.2%	12	6	4
0	0%	10	10%	398.43	1.067	470.39	2084.62	1.6%	2.1%	12	7	0
0	0%	10	25%	403.09	0.760	480.76	1894.77	3.7%	2.5%	12	7	0
0	0%	10	50%	323.88	0.720	487.21	1377.46	10.4%	3.1%	12	4	4
1	10%	1	10%	396.72	0.11	466.89	97.14	0.01	0.0%	12	12	0
1	25%	1	25%	397.45	0.11	475.28	181.82	0.02	0.0%	12	12	0
1	50%	1	50%	310.62	0.13	464.38	105.86	0.05	0.0%	12	8	4
5	10%	5	10%	397.73	0.54	469.40	1446.62	0.01	0.5%	12	11	0
5	25%	5	25%	400.71	0.50	480.76	1747.01	0.04	1.9%	12	7	0
5	50%	5	50%	318.68	0.51	484.38	1381.01	0.10	2.3%	12	4	4
10	10%	10	10%	398.43	5.21	472.32	2442.73	0.02	5.7%	12	4	0
10	25%	10	25%	403.09	4.36	481.55	2302.26	0.04	4.0%	12	6	0
10	50%	10	50%	323.88	5.13	487.43	1624.86	0.10	4.5%	12	4	4
All				380.65	0.80	471.25	882.63	0.03	1.1%	336	260	24

Table 38 – Average results from the CF formulation for the RVRPTW instances of class RC1

RC1												
				Linear Relaxation		**Integer Solution**						
Γ	***Dev***^{d}	**Γ**	***Dev***^{t}	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	290.87	0.037	350.24	3.14	0.0%	0.0%	8	8	0
1	10%	0	0%	290.97	0.065	350.24	5.62	0.0%	0.0%	8	8	0
1	25%	0	0%	291.15	0.068	350.24	5.63	0.0%	0.0%	8	8	0
1	50%	0	0%	291.54	0.065	379.94	1071.28	9.4%	2.0%	8	6	0
5	10%	0	0%	291.36	0.263	379.94	2107.46	9.4%	4.6%	8	4	0
5	25%	0	0%	292.72	0.241	456.58	3151.98	32.2%	17.5%	8	1	0
5	50%	0	0%	296.98	0.240	464.85	3153.68	34.6%	17.7%	8	1	0
10	10%	0	0%	292.02	0.453	388.66	2358.26	12.2%	8.9%	8	3	0
10	25%	0	0%	296.31	0.711	459.39	3157.01	33.0%	20.2%	8	1	0
10	50%	0	0%	313.37	0.568	482.61	3171.83	40.1%	20.9%	8	1	0
0	0%	1	10%	291.05	0.087	350.91	6.16	0.1%	0.0%	8	8	0
0	0%	1	25%	291.56	0.085	351.64	6.58	0.4%	0.0%	8	8	0
0	0%	1	50%	292.78	0.082	364.16	7.19	5.8%	0.0%	8	7	1
0	0%	5	10%	291.26	0.319	352.10	42.18	0.5%	0.0%	8	8	0
0	0%	5	25%	292.28	0.323	365.35	41.07	3.9%	0.0%	8	8	0
0	0%	5	50%	296.25	0.302	397.97	1195.21	16.1%	3.3%	8	5	1
0	0%	10	10%	291.38	1.021	352.10	149.06	0.5%	0.0%	8	8	0
0	0%	10	25%	292.78	0.865	366.83	266.81	4.3%	0.0%	8	8	0
0	0%	10	50%	298.96	0.974	423.00	2019.82	24.1%	10.1%	8	3	1
1	10%	1	10%	291.14	0.12	350.91	8.83	0.00	0.0%	8	8	0
1	25%	1	25%	291.81	0.11	351.64	11.05	0.00	0.0%	8	8	0
1	50%	1	50%	293.30	0.12	404.84	1727.97	0.19	4.4%	8	4	1
5	10%	5	10%	291.72	0.62	385.54	2360.07	0.11	7.1%	8	3	0
5	25%	5	25%	293.82	0.58	471.99	3151.11	0.37	21.8%	8	1	0
5	50%	5	50%	300.71	0.69	485.70	2700.27	0.44	23.6%	8	1	1
10	10%	10	10%	292.46	4.49	390.56	3049.27	0.13	10.0%	8	3	0
10	25%	10	25%	297.55	5.09	479.68	3161.60	0.39	25.3%	8	1	0
10	50%	10	50%	316.82	4.18	505.69	2700.69	0.51	26.9%	8	1	1
All				294.82	0.81	400.47	1456.82	0.16	8.0%	224	134	6

Table 39 – Average results from the CF formulation for the RVRPTW instances of class C2

C2												
				Linear Relaxation		**Integer Solution**						
Γ	***Dev***^{d}	**Γ**	***Dev***^{t}	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	181.59	0.025	214.45	1.94	0.0%	0.0%	8	8	0
1	10%	0	0%	181.59	0.052	214.45	2.84	0.0%	0.0%	8	8	0
1	25%	0	0%	181.59	0.056	214.45	2.73	0.0%	0.0%	8	8	0
1	50%	0	0%	181.59	0.060	214.45	2.93	0.0%	0.0%	8	8	0
5	10%	0	0%	181.59	0.197	214.45	32.50	0.0%	0.0%	8	8	0
5	25%	0	0%	181.59	0.208	214.45	28.70	0.0%	0.0%	8	8	0
5	50%	0	0%	181.59	0.211	214.45	36.06	0.0%	0.0%	8	8	0
10	10%	0	0%	181.59	0.407	214.45	141.64	0.0%	0.0%	8	8	0
10	25%	0	0%	181.59	0.369	214.45	220.43	0.0%	0.0%	8	8	0
10	50%	0	0%	181.59	0.487	214.45	248.92	0.0%	0.0%	8	8	0
0	0%	1	10%	181.59	0.072	214.45	3.84	0.0%	0.0%	8	8	0
0	0%	1	25%	181.60	0.072	214.45	4.23	0.0%	0.0%	8	8	0
0	0%	1	50%	181.61	0.079	214.63	4.17	0.1%	0.0%	8	8	0
0	0%	5	10%	181.60	0.237	214.45	37.15	0.0%	0.0%	8	8	0
0	0%	5	25%	181.61	0.250	214.63	39.95	0.1%	0.0%	8	8	0
0	0%	5	50%	181.64	0.258	214.63	51.19	0.1%	0.0%	8	8	0
0	0%	10	10%	181.60	0.769	214.45	253.81	0.0%	0.0%	8	8	0
0	0%	10	25%	181.62	0.608	214.63	295.70	0.1%	0.0%	8	8	0
0	0%	10	50%	181.66	0.646	214.63	282.42	0.1%	0.0%	8	8	0
1	10%	1	10%	181.59	0.10	214.45	6.76	0.00	0.0%	8	8	0
1	25%	1	25%	181.60	0.10	214.45	6.59	0.00	0.0%	8	8	0
1	50%	1	50%	181.61	0.10	214.63	7.67	0.00	0.0%	8	8	0
5	10%	5	10%	181.60	0.63	214.45	123.84	0.00	0.0%	8	8	0
5	25%	5	25%	181.61	0.58	214.63	147.35	0.00	0.0%	8	8	0
5	50%	5	50%	181.64	0.60	214.63	184.75	0.00	0.0%	8	8	0
10	10%	10	10%	181.60	3.06	214.63	909.46	0.00	0.7%	8	7	0
10	25%	10	25%	181.62	3.77	214.63	903.98	0.00	0.7%	8	7	0
10	50%	10	50%	181.66	3.76	214.63	991.18	0.00	1.5%	8	6	0
All				181.61	0.63	214.52	177.60	0.00	0.1%	224	220	0

Table 40 – Average results from the CF formulation for the RVRPTW instances of class R2

R2												
				Linear Relaxation		**Integer Solution**						
Γ	***Dev*^{d}**	**Γ**	***Dev*^{t}**	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	313.97	0.025	382.15	83.19	0.0%	0.0%	11	11	0
1	10%	0	0%	313.97	0.058	382.15	134.83	0.0%	0.0%	11	11	0
1	25%	0	0%	313.97	0.056	382.15	198.77	0.0%	0.0%	11	11	0
1	50%	0	0%	313.97	0.058	382.15	208.01	0.0%	0.0%	11	11	0
5	10%	0	0%	313.97	0.215	382.68	1495.33	0.1%	1.7%	11	7	0
5	25%	0	0%	313.97	0.187	382.21	1401.58	0.0%	1.1%	11	7	0
5	50%	0	0%	313.97	0.178	382.57	1439.65	0.1%	1.7%	11	7	0
10	10%	0	0%	313.97	0.433	383.44	2077.64	0.4%	3.4%	11	5	0
10	25%	0	0%	313.97	0.441	384.96	2076.17	0.8%	3.8%	11	5	0
10	50%	0	0%	313.97	0.514	383.63	2056.69	0.4%	3.1%	11	6	0
0	0%	1	10%	313.99	0.085	382.15	200.23	0.0%	0.0%	11	11	0
0	0%	1	25%	314.02	0.075	383.55	234.24	0.4%	0.0%	11	11	0
0	0%	1	50%	314.09	0.086	385.88	256.80	0.9%	0.0%	11	11	0
0	0%	5	10%	314.03	0.292	383.96	1515.05	0.5%	1.6%	11	7	0
0	0%	5	25%	314.14	0.304	384.43	1633.58	0.7%	1.7%	11	7	0
0	0%	5	50%	314.37	0.323	387.62	1724.86	1.4%	2.0%	11	7	0
0	0%	10	10%	314.05	0.692	385.96	2325.91	1.1%	4.1%	11	5	0
0	0%	10	25%	314.21	0.554	387.18	2362.29	1.4%	4.0%	11	5	0
0	0%	10	50%	314.58	0.537	388.94	2381.51	1.8%	3.8%	11	6	0
1	10%	1	10%	313.99	0.11	382.15	263.6415	0.00	0.0%	11	11	0
1	25%	1	25%	314.02	0.10	383.55	540.59	0.00	0.0%	11	11	0
1	50%	1	50%	314.09	0.10	385.88	439.49	0.01	0.0%	11	11	0
5	10%	5	10%	314.03	0.46	385.17	2096.42	0.01	2.8%	11	6	0
5	25%	5	25%	314.14	0.43	384.42	2188.39	0.01	2.8%	11	5	0
5	50%	5	50%	314.37	0.42	388.80	2195.27	0.02	3.5%	11	5	0
10	10%	10	10%	314.05	3.84	390.18	2828.66	0.02	7.2%	11	3	0
10	25%	10	25%	314.21	3.47	396.66	2825.72	0.04	8.4%	11	4	0
10	50%	10	50%	314.58	3.77	392.79	2894.77	0.03	7.2%	11	4	0
All				314.09	0.64	385.26	1431.40	0.01	2.3%	308	211	0

Table 41 – Average results from the CF formulation for the RVRPTW instances of class RC2

RC2												
				Linear Relaxation		**Integer Solution**						
Γ	***Dev*^{d}**	**Γ**	***Dev*^{t}**	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	173.10	0.038	319.28	1812.10	0.0%	5.0%	8	4	0
1	10%	0	0%	173.10	0.071	319.28	1816.86	0.0%	5.0%	8	4	0
1	25%	0	0%	173.10	0.072	319.28	1819.94	0.0%	5.4%	8	4	0
1	50%	0	0%	173.10	0.075	319.28	1816.55	0.0%	6.3%	8	4	0
5	10%	0	0%	173.10	0.238	319.28	2070.55	0.0%	7.7%	8	4	0
5	25%	0	0%	173.10	0.227	319.28	2272.78	0.0%	8.1%	8	3	0
5	50%	0	0%	173.10	0.212	319.96	2080.60	0.2%	7.9%	8	4	0
10	10%	0	0%	173.10	0.502	320.81	2339.06	0.5%	10.5%	8	3	0
10	25%	0	0%	173.10	0.551	324.03	2358.31	1.7%	11.3%	8	3	0
10	50%	0	0%	173.10	0.544	320.90	2381.56	0.5%	10.3%	8	3	0
0	0%	1	10%	173.13	0.087	319.28	1824.78	0.0%	5.9%	8	4	0
0	0%	1	25%	173.17	0.094	319.75	1818.91	0.2%	6.1%	8	4	0
0	0%	1	50%	173.24	0.085	319.89	1815.58	0.2%	5.9%	8	4	0
0	0%	5	10%	173.16	0.339	319.36	1958.85	0.0%	8.3%	8	4	0
0	0%	5	25%	173.24	0.345	320.31	1964.52	0.3%	7.9%	8	4	0
0	0%	5	50%	173.44	0.321	322.36	1936.89	1.1%	8.9%	8	4	0
0	0%	10	10%	173.17	0.559	323.14	2376.68	1.4%	10.7%	8	3	0
0	0%	10	25%	173.31	0.511	323.01	2301.66	1.4%	10.0%	8	4	0
0	0%	10	50%	173.70	0.557	323.30	2136.37	1.5%	10.3%	8	4	0
1	10%	1	10%	173.13	0.11	319.28	1829.14	0.00	6.7%	8	4	0
1	25%	1	25%	173.17	0.11	319.95	1828.79	0.00	6.6%	8	4	0
1	50%	1	50%	173.24	0.11	319.95	1814.25	0.00	6.6%	8	4	0
5	10%	5	10%	173.16	0.47	321.94	2303.15	0.01	10.2%	8	3	0
5	25%	5	25%	173.24	0.44	323.20	2307.97	0.01	10.2%	8	3	0
5	50%	5	50%	173.44	0.50	324.16	2318.84	0.02	10.2%	8	4	0
10	10%	10	10%	173.17	3.61	323.34	2816.88	0.01	13.7%	8	3	0
10	25%	10	25%	173.31	3.73	330.59	2856.85	0.04	15.2%	8	3	0
10	50%	10	50%	173.70	4.08	330.11	2661.01	0.04	16.6%	8	3	0
All				173.22	0.66	321.58	2129.98	0.01	8.8%	224	102	0

B.2 Single knapsack Uncertainty set

In this section, we present in more details the results of our computational experiments in instances of the literature for the cardinality constrained uncertainty set using the compact models. We created tables for each model (MTZ and CF) showing the relevant information for each set of instances separately. In these tables, for every combination of Δ^t, Δ^d and deviation (Dev) studied, we present the average solution value (Sol) and the time taken to solve the problem ($Time$) of both the linear relaxation and integer problem, and the Price-of-Robustness (PoR), which represents how much the robust solution costs more than the deterministic solution, expressed in percentage. To better evaluate the computational results, we also present the percentage gap between the best solution obtained in the time limit and its lower bound (Gap), if the solution is optimal this difference is zero. Finally, we also present the number of instances (Ins) evaluated in that combination of Δ^q, Δ^t and deviations, how many of those were solved to optimality by the model within the time limit (Opt) and the number of infeasible instances (Inf) in that group.

B.2.1 MTZ-based formulation

Table 42 – Average results from the MTZ formulation for the RVRPTW instances of class C1 with the single knapsack uncertainty set

				C1								
				Linear Relaxation		**Integer Solution**						
Δ	Dev^d	Δ	Dev^t	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	76.89	0.068	190.02	758.05	0.0%	1.5%	5	4	4
20	10%	0	0%	76.89	0.073	224.58	1454.73	18.2%	11.1%	5	3	4
20	25%	0	0%	76.89	0.070	224.58	1509.07	18.2%	10.1%	5	3	4
20	50%	0	0%	76.89	0.137	224.58	1467.64	18.2%	9.9%	5	3	4
40	10%	0	0%	76.89	0.135	224.58	1526.61	18.2%	12.1%	5	3	4
40	25%	0	0%	76.89	0.131	235.08	2160.85	23.7%	14.9%	5	2	4
40	50%	0	0%	76.89	0.205	234.84	2160.74	23.6%	15.7%	5	2	4
60	10%	0	0%	76.89	0.216	224.58	1591.79	18.2%	12.2%	5	3	4
60	25%	0	0%	76.89	0.205	254.16	2166.14	33.8%	19.7%	5	2	4
60	50%	0	0%	76.91	0.436	254.94	2165.00	34.2%	19.3%	5	2	4
0	0%	20	10%	76.92	0.507	190.46	1459.69	0.2%	7.4%	5	3	4
0	0%	20	25%	76.92	0.526	190.02	1463.85	0.0%	7.3%	5	3	4
0	0%	20	50%	76.92	1.848	190.10	1462.17	0.0%	7.7%	5	3	4
0	0%	40	10%	76.93	3.075	190.22	1524.71	0.1%	9.7%	5	3	4
0	0%	40	25%	76.94	3.461	190.94	1599.89	0.5%	9.5%	5	3	4
0	0%	40	50%	76.92	3.652	191.22	1650.06	0.6%	9.7%	5	3	4
0	0%	60	10%	76.93	7.776	191.94	1714.03	1.0%	10.4%	5	3	4
0	0%	60	25%	76.96	9.108	191.88	1868.69	1.0%	10.6%	5	3	4
0	0%	60	50%	76.91	0.539	193.98	1906.88	2.1%	11.6%	5	3	4
20	10%	20	10%	76.92	0.62	225.56	1635.90	0.19	14.0%	5	3	4
20	25%	20	25%	76.92	0.59	225.00	1704.40	0.18	13.6%	5	3	4
20	50%	20	50%	76.92	1.83	224.98	1726.47	0.18	13.3%	5	3	4
40	10%	40	10%	76.93	2.79	230.64	2164.70	0.21	17.2%	5	2	4
40	25%	40	25%	76.94	3.35	238.34	2168.93	0.25	19.7%	5	2	4
40	50%	40	50%	76.92	3.51	240.28	2170.28	0.26	20.2%	5	2	4
60	10%	60	10%	76.93	7.54	228.54	2168.88	0.20	18.3%	5	2	4
60	25%	60	25%	76.96	8.58	259.86	2233.10	0.37	23.6%	5	2	4
60	50%	60	50%	90.21	0.01	259.58	2200.48	0.37	23.6%	5	2	4
All				77.39	2.18	219.48	1777.99	0.16	13.4%	140	75	112

Table 43 – Average results from the MTZ formulation for the RVRPTW instances of class R1 with the single knapsack uncertainty set

				R1								
				Linear Relaxation		**Integer Solution**						
Δ	Dev^d	Δ	Dev^t	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	299.35	0.074	513.24	9.72	0.0%	0.0%	5	5	7
20	10%	0	0%	299.35	0.060	513.24	277.01	0.0%	0.0%	5	5	7
20	25%	0	0%	299.35	0.064	513.24	467.39	0.0%	0.0%	5	5	7
20	50%	0	0%	299.35	0.131	513.24	457.49	0.0%	0.0%	5	5	7
40	10%	0	0%	299.35	0.141	513.24	832.56	0.0%	1.1%	5	4	7
40	25%	0	0%	299.35	0.128	513.24	921.98	0.0%	1.2%	5	4	7
40	50%	0	0%	299.35	0.214	513.24	899.51	0.0%	1.0%	5	4	7
60	10%	0	0%	299.35	0.207	513.24	1103.33	0.0%	1.4%	5	4	7
60	25%	0	0%	299.35	0.211	514.54	1142.37	0.3%	1.8%	5	4	7
60	50%	0	0%	299.87	0.651	513.24	1042.80	0.0%	1.6%	5	4	7
0	0%	20	10%	299.98	0.687	513.24	918.22	0.0%	1.8%	5	4	7
0	0%	20	25%	299.99	0.708	523.76	1097.05	1.7%	1.4%	5	4	7
0	0%	20	50%	300.12	3.958	523.76	1168.95	1.7%	1.3%	5	4	7
0	0%	40	10%	300.48	4.942	516.14	1521.99	0.7%	4.4%	5	3	7
0	0%	40	25%	300.71	5.146	533.36	1549.96	3.9%	5.3%	5	3	7
0	0%	40	50%	300.25	8.495	553.12	1496.02	7.3%	4.9%	5	3	7
0	0%	60	10%	300.83	12.853	524.54	1556.82	2.7%	7.8%	5	3	7
0	0%	60	25%	301.36	12.998	540.24	1861.34	5.5%	8.4%	5	3	7
0	0%	60	50%	299.87	0.733	557.68	1817.51	8.4%	7.0%	5	3	7
20	10%	20	10%	299.98	0.75	519.18	1453.06	0.01	3.4%	5	3	7
20	25%	20	25%	299.99	0.73	523.84	1042.87	0.02	1.8%	5	4	7
20	50%	20	50%	300.12	4.10	526.54	1144.60	0.02	2.5%	5	4	7
40	10%	40	10%	300.48	4.89	517.50	1661.36	0.01	5.2%	5	3	7
40	25%	40	25%	300.71	4.81	534.52	1562.07	0.04	6.3%	5	3	7
40	50%	40	50%	300.25	8.15	554.32	1540.66	0.08	6.0%	5	3	7
60	10%	60	10%	300.83	12.07	524.16	1542.41	0.03	7.9%	5	3	7
60	25%	60	25%	301.36	12.57	537.56	2036.26	0.05	9.2%	5	3	7
60	50%	60	50%	291.02	0.01	555.30	1999.32	0.08	6.8%	5	3	7
All				299.73	3.59	525.45	1218.74	0.02	3.6%	140	103	196

Table 44 – Average results from the MTZ formulation for the RVRPTW instances of class RC1 with the single knapsack uncertainty set

						RC1						
				Linear Relaxation		**Integer Solution**						
Δ	***Dev*^{d}**	**Δ**	***Dev*^{t}**	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	95.63	0.072	372.72	1254.27	0.0%	5.6%	5	4	0
20	10%	0	0%	95.63	0.065	389.78	1541.00	5.2%	16.0%	5	3	0
20	25%	0	0%	95.63	0.065	390.12	1499.64	5.3%	16.6%	5	3	0
20	50%	0	0%	95.63	0.139	389.78	1552.64	5.2%	16.8%	5	3	0
40	10%	0	0%	95.63	0.138	389.78	1756.75	5.2%	16.7%	5	3	0
40	25%	0	0%	95.63	0.130	466.52	2883.36	26.8%	33.6%	5	1	0
40	50%	0	0%	95.63	0.206	466.84	2881.97	26.9%	30.2%	5	1	0
60	10%	0	0%	95.63	0.204	395.18	1809.60	7.0%	23.7%	5	3	0
60	25%	0	0%	95.63	0.221	473.16	2887.90	28.8%	33.5%	5	1	0
60	50%	0	0%	95.66	0.391	473.82	2886.68	28.9%	34.1%	5	1	0
0	0%	20	10%	95.70	0.439	374.18	1595.31	0.4%	18.2%	5	3	0
0	0%	20	25%	95.73	0.448	374.14	1670.94	0.3%	18.6%	5	3	0
0	0%	20	50%	95.66	2.426	374.34	1785.18	0.4%	16.8%	5	3	1
0	0%	40	10%	95.70	2.884	379.08	2338.40	1.9%	23.6%	5	3	0
0	0%	40	25%	95.76	3.311	378.72	2147.17	1.8%	20.6%	5	3	0
0	0%	40	50%	95.66	5.370	381.98	2058.02	2.7%	23.2%	5	3	1
0	0%	60	10%	95.70	6.685	383.06	2887.42	3.1%	30.1%	5	1	0
0	0%	60	25%	95.77	8.221	393.46	2886.85	6.1%	30.8%	5	1	0
0	0%	60	50%	95.66	0.494	393.58	2883.99	6.0%	27.0%	5	1	1
20	10%	20	10%	95.70	0.54	393.00	2092.70	0.06	21.4%	5	3	0
20	25%	20	25%	95.73	0.53	397.92	2519.77	0.08	22.0%	5	3	0
20	50%	20	50%	95.66	2.44	394.32	1899.69	0.07	21.6%	5	3	1
40	10%	40	10%	95.70	2.83	407.70	2890.43	0.11	31.4%	5	1	0
40	25%	40	25%	95.76	3.10	488.70	2891.78	0.33	41.9%	5	1	0
40	50%	40	50%	95.66	5.07	493.00	2887.94	0.34	38.1%	5	1	1
60	10%	60	10%	95.70	6.32	413.66	2894.03	0.12	35.6%	5	1	0
60	25%	60	25%	95.77	7.41	495.44	2911.01	0.35	46.3%	5	1	0
60	50%	60	50%	95.83	0.01	497.84	2895.42	0.36	41.0%	5	1	1
All				95.69	2.15	415.07	2324.64	0.12	26.3%	140	59	6

Table 45 – Average results from the MTZ formulation for the RVRPTW instances of class C2 with the single knapsack uncertainty set

						C2						
				Linear Relaxation		**Integer Solution**						
Δ	***Dev*^{d}**	**Δ**	***Dev*^{t}**	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	145.32	0.069	214.38	43.32	0.0%	0.0%	5	5	0
20	10%	0	0%	145.32	0.077	214.38	361.07	0.0%	0.0%	5	5	0
20	25%	0	0%	145.32	0.059	214.38	482.34	0.0%	0.0%	5	5	0
20	50%	0	0%	145.32	0.184	214.38	454.98	0.0%	0.0%	5	5	0
40	10%	0	0%	145.32	0.172	214.38	791.25	0.0%	2.3%	5	4	0
40	25%	0	0%	145.32	0.181	214.66	1009.08	0.1%	1.8%	5	4	0
40	50%	0	0%	145.32	0.314	214.38	777.21	0.0%	1.8%	5	4	0
60	10%	0	0%	145.32	0.298	214.38	1367.24	0.0%	2.4%	5	4	0
60	25%	0	0%	145.32	0.333	214.38	982.92	0.0%	2.7%	5	4	0
60	50%	0	0%	145.33	0.373	214.38	915.40	0.0%	2.5%	5	4	0
0	0%	20	10%	145.33	0.438	214.38	1447.77	0.0%	3.4%	5	3	0
0	0%	20	25%	145.33	0.495	214.66	830.34	0.1%	2.7%	5	4	0
0	0%	20	50%	145.33	1.923	215.40	875.00	0.5%	2.1%	5	4	0
0	0%	40	10%	145.33	3.173	216.38	1508.85	0.9%	5.9%	5	3	0
0	0%	40	25%	145.34	3.676	218.84	1492.13	2.1%	7.5%	5	3	0
0	0%	40	50%	145.33	4.134	217.98	1492.85	1.7%	7.0%	5	3	0
0	0%	60	10%	145.34	7.679	220.92	1576.15	3.1%	8.6%	5	3	0
0	0%	60	25%	145.35	9.776	222.40	1678.94	3.8%	10.6%	5	3	0
0	0%	60	50%	145.33	0.502	221.98	1817.67	3.6%	10.7%	5	3	0
20	10%	20	10%	145.33	0.62	215.40	1147.43	0.00	3.6%	5	4	0
20	25%	20	25%	145.33	0.62	214.66	963.74	0.00	3.3%	5	4	0
20	50%	20	50%	145.33	2.28	214.38	966.43	0.00	3.0%	5	4	0
40	10%	40	10%	145.33	3.26	219.64	1535.33	0.02	7.9%	5	3	0
40	25%	40	25%	145.34	3.67	217.18	1504.96	0.01	6.7%	5	3	0
40	50%	40	50%	145.33	4.85	219.64	1514.70	0.02	8.0%	5	3	0
60	10%	60	10%	145.34	7.95	218.54	1667.81	0.02	8.3%	5	3	0
60	25%	60	25%	145.35	10.01	220.32	1650.85	0.03	10.8%	5	3	0
60	50%	60	50%	180.79	0.01	219.96	1770.99	0.03	10.4%	5	3	0
All				146.60	2.40	216.67	1165.24	0.01	4.8%	140	103	0

Table 46 – Average results from the MTZ formulation for the RVRPTW instances of class R2 with the single knapsack uncertainty set

R2												
				Linear Relaxation		**Integer Solution**						
Δ	***Dev***^{d}	**Δ**	***Dev***^{t}	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	281.73	0.071	402.64	38.04	0.0%	0.0%	5	5	0
20	10%	0	0%	281.73	0.063	402.72	1224.88	0.0%	0.5%	5	4	0
20	25%	0	0%	281.73	0.063	402.64	985.02	0.0%	0.0%	5	5	0
20	50%	0	0%	281.73	0.134	402.64	1070.06	0.0%	0.0%	5	5	0
40	10%	0	0%	281.73	0.134	402.64	1484.64	0.0%	1.8%	5	3	0
40	25%	0	0%	281.73	0.142	402.64	1465.83	0.0%	2.0%	5	3	0
40	50%	0	0%	281.73	0.209	403.24	1464.75	0.2%	2.3%	5	3	0
60	10%	0	0%	281.73	0.207	402.72	1518.46	0.0%	2.8%	5	3	0
60	25%	0	0%	281.73	0.206	403.14	1528.92	0.1%	3.2%	5	3	0
60	50%	0	0%	281.76	0.432	403.14	1481.59	0.1%	3.4%	5	3	0
0	0%	20	10%	281.77	0.484	405.68	1521.16	0.8%	4.1%	5	3	0
0	0%	20	25%	281.77	0.461	402.94	1547.05	0.1%	3.3%	5	3	0
0	0%	20	50%	281.76	2.455	403.24	1513.15	0.2%	3.6%	5	3	0
0	0%	40	10%	281.78	3.643	410.60	1938.43	2.1%	7.1%	5	3	0
0	0%	40	25%	281.81	3.859	411.50	2149.81	2.4%	7.4%	5	3	0
0	0%	40	50%	281.76	5.123	411.56	1981.89	2.4%	7.2%	5	3	0
0	0%	60	10%	281.78	9.102	419.60	2318.30	4.6%	11.6%	5	2	0
0	0%	60	25%	281.83	10.470	409.96	2455.47	2.0%	9.8%	5	2	0
0	0%	60	50%	281.76	0.513	423.36	2323.78	5.5%	11.1%	5	2	0
20	10%	20	10%	281.77	0.53	406.88	1634.383	0.01	5.4%	5	3	0
20	25%	20	25%	281.77	0.53	406.62	1584.18	0.01	5.1%	5	3	0
20	50%	20	50%	281.76	2.33	406.46	1572.61	0.01	5.1%	5	3	0
40	10%	40	10%	281.78	3.47	411.18	2186.63	0.02	7.5%	5	3	0
40	25%	40	25%	281.81	3.56	410.90	2239.58	0.02	8.7%	5	2	0
40	50%	40	50%	281.76	4.82	411.70	2158.89	0.02	7.5%	5	3	0
60	10%	60	10%	281.78	8.53	412.98	2368.06	0.03	10.5%	5	2	0
60	25%	60	25%	281.83	8.83	415.12	2431.08	0.03	11.1%	5	2	0
60	50%	60	50%	243.53	0.00	424.58	2524.81	0.06	11.4%	5	2	0
All				280.40	2.51	408.32	1739.69	0.02	5.5%	140	84	0

Table 47 – Average results from the MTZ formulation for the RVRPTW instances of class RC2 with the single knapsack uncertainty set

RC2												
				Linear Relaxation		**Integer Solution**						
Δ	***Dev***^{d}	**Δ**	***Dev***^{t}	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	95.77	0.062	332.56	1452.77	0.0%	15.5%	5	3	0
20	10%	0	0%	95.77	0.063	333.02	1746.18	0.1%	16.9%	5	3	0
20	25%	0	0%	95.77	0.060	332.68	1857.97	0.0%	17.6%	5	3	0
20	50%	0	0%	95.77	0.127	332.56	1867.54	0.0%	17.3%	5	3	0
40	10%	0	0%	95.77	0.126	332.94	2162.07	0.1%	24.0%	5	2	0
40	25%	0	0%	95.77	0.133	332.56	2005.92	0.0%	18.8%	5	3	0
40	50%	0	0%	95.77	0.212	332.56	2162.85	0.0%	22.8%	5	2	0
60	10%	0	0%	95.77	0.205	333.36	2174.17	0.3%	25.1%	5	2	0
60	25%	0	0%	95.77	0.207	333.98	2163.94	0.5%	24.3%	5	2	0
60	50%	0	0%	95.77	0.365	333.06	2164.71	0.2%	25.0%	5	2	0
0	0%	20	10%	95.78	0.408	332.68	2170.15	0.0%	25.4%	5	2	0
0	0%	20	25%	95.79	0.430	334.86	2163.87	0.7%	25.3%	5	2	0
0	0%	20	50%	95.77	2.019	333.84	2164.84	0.4%	24.3%	5	2	0
0	0%	40	10%	95.78	2.515	336.38	2227.68	1.2%	27.4%	5	2	0
0	0%	40	25%	95.79	3.055	338.76	2228.78	1.9%	27.6%	5	2	0
0	0%	40	50%	95.77	4.459	336.30	2213.57	1.2%	27.8%	5	2	0
0	0%	60	10%	95.78	5.726	337.42	2313.93	1.6%	28.7%	5	2	0
0	0%	60	25%	95.79	7.297	337.90	2085.84	1.3%	20.6%	4	2	0
0	0%	60	50%	95.77	0.447	338.82	2398.01	2.0%	29.9%	5	2	0
20	10%	20	10%	95.78	0.49	333.60	2173.65	0.00	26.3%	5	2	0
20	25%	20	25%	95.79	0.52	333.54	2167.59	0.00	26.4%	5	2	0
20	50%	20	50%	95.77	1.90	333.26	2167.91	0.00	26.1%	5	2	0
40	10%	40	10%	95.78	2.29	337.70	2228.75	0.02	28.6%	5	2	0
40	25%	40	25%	95.79	2.86	337.96	2432.61	0.02	28.6%	5	2	0
40	50%	40	50%	95.77	4.40	338.56	2248.72	0.02	28.7%	5	2	0
60	10%	60	10%	95.78	5.44	342.08	2474.23	0.03	30.2%	5	2	0
60	25%	60	25%	95.79	6.72	337.40	2882.87	0.02	30.0%	5	2	0
60	50%	60	50%	95.56	0.00	339.24	2889.47	0.02	30.9%	5	1	0
All				95.77	1.88	335.34	2192.52	0.01	25.0%	139	60	0

B.2.2 CF formulation

Table 48 – Average results from the CF formulation for the RVRPTW instances of class C1 with the single knapsack uncertainty set

						C1						
				Linear Relaxation		**Integer Solution**						
Δ	***Dev***^{d}	**Δ**	***Dev***^{t}	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	176.14	0.051	190.02	0.74	0.0%	0.0%	5	5	4
20	10%	0	0%	177.60	6.874	229.18	2099.34	20.6%	9.4%	5	3	4
20	25%	0	0%	178.62	8.592	225.58	1710.38	18.7%	5.2%	5	3	4
20	50%	0	0%	179.68	4.650	224.58	1025.26	18.2%	0.0%	5	5	4
40	10%	0	0%	178.08	26.531	239.12	2187.74	25.9%	16.0%	5	2	4
40	25%	0	0%	180.42	40.375	247.24	2258.50	30.1%	17.0%	5	2	4
40	50%	0	0%	182.38	24.351	244.16	2253.37	28.5%	15.1%	5	2	4
60	10%	0	0%	178.39	50.554	243.08	2204.13	28.0%	17.5%	5	2	4
60	25%	0	0%	181.95	80.049	257.30	2990.12	35.4%	22.0%	5	1	4
60	50%	0	0%	184.76	49.746	259.26	2975.51	36.4%	19.3%	5	1	4
0	0%	20	10%	176.16	59.761	378.04	2165.00	98.8%	20.1%	5	2	4
0	0%	20	25%	176.16	77.187	566.74	2167.42	199.8%	35.3%	5	2	4
0	0%	20	50%	176.16	77.168	378.72	2165.62	101.0%	20.3%	5	2	4
0	0%	40	10%	176.16	335.527	754.76	2193.43	298.6%	51.1%	5	2	4
0	0%	40	25%	176.17	573.585	754.76	2194.06	298.6%	51.1%	5	2	4
0	0%	40	50%	176.18	579.033	754.76	2237.04	298.6%	51.1%	5	2	4
0	0%	60	10%	176.17	1031.407	754.76	2309.20	298.6%	51.1%	5	2	4
0	0%	60	25%	179.00	1658.581	660.85	2023.52	247.0%	48.4%	4	2	4
0	0%	60	50%	184.71	1068.661	504.33	1873.73	164.7%	32.1%	3	2	4
20	10%	20	10%	177.61	106.32	255.06	2600.00	0.34	21.2%	5	2	4
20	25%	20	25%	178.64	109.23	421.60	2535.27	1.23	29.4%	5	2	4
20	50%	20	50%	179.69	108.20	595.08	2650.34	2.15	39.8%	5	2	4
40	10%	40	10%	178.10	644.05	775.64	3589.75	3.09	55.7%	5	1	4
40	25%	40	25%	180.44	902.17	952.06	3600.84	4.02	68.1%	5	0	4
40	50%	40	50%	182.39	820.21	954.04	3600.98	4.03	71.1%	5	0	4
60	10%	60	10%	187.30	2065.57	952.06	3851.71	4.02	76.0%	5	0	4
60	25%	60	25%	198.55	2237.86	1130.40	3879.65	4.92	89.6%	4	0	4
60	50%	60	50%	202.82	2246.79	1130.40	3748.54	4.92	86.7%	3	0	4
All				180.73	535.47	536.91	2467.54	1.83	36.4%	134	51	112

Table 49 – Average results from the CF formulation for the RVRPTW instances of class R1 with the single knapsack uncertainty set

						R1						
				Linear Relaxation		**Integer Solution**						
Δ	***Dev***^{d}	**Δ**	***Dev***^{t}	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	448.45	0.041	513.24	21.55	0.0%	0.0%	5	5	7
20	10%	0	0%	448.45	1.516	516.38	1454.75	0.8%	2.2%	5	3	7
20	25%	0	0%	448.45	1.461	513.24	1458.07	0.0%	1.1%	5	3	7
20	50%	0	0%	448.45	0.841	513.24	1187.70	0.0%	0.9%	5	4	7
40	10%	0	0%	448.45	3.168	513.24	1531.97	0.0%	3.0%	5	3	7
40	25%	0	0%	448.45	9.345	521.10	1537.61	1.8%	5.5%	5	3	7
40	50%	0	0%	448.45	6.371	515.02	1507.26	0.4%	3.2%	5	3	7
60	10%	0	0%	448.45	9.077	518.84	1571.86	1.3%	5.2%	5	3	7
60	25%	0	0%	448.45	19.171	516.88	1602.58	0.8%	5.2%	5	3	7
60	50%	0	0%	448.45	28.598	520.22	1566.51	1.6%	6.0%	5	3	7
0	0%	20	10%	448.82	94.417	680.56	2196.86	40.1%	20.1%	5	2	7
0	0%	20	25%	448.87	63.522	696.38	2191.32	42.9%	20.9%	5	2	7
0	0%	20	50%	448.87	66.850	582.43	770.16	2.8%	4.4%	5	2	9
0	0%	40	10%	449.28	892.742	976.28	2795.26	100.0%	46.5%	5	2	7
0	0%	40	25%	449.63	989.579	987.94	2935.28	101.9%	52.6%	5	1	7
0	0%	40	50%	450.22	1034.991	834.53	1139.25	48.6%	20.6%	5	2	9
0	0%	60	10%	512.72	1748.592	977.54	3132.50	100.2%	53.2%	5	1	7
0	0%	60	25%	513.51	653.684	928.20	2832.02	78.6%	42.4%	4	1	8
0	0%	60	50%	553.27	65.542	836.53	1454.24	48.9%	22.6%	5	1	9
20	10%	20	10%	448.82	79.00	555.40	2238.06	0.10	15.2%	5	2	7
20	25%	20	25%	448.87	65.83	560.12	2185.74	0.10	14.7%	5	2	7
20	50%	20	50%	448.87	88.96	588.83	772.13	0.04	5.9%	5	2	9
40	10%	40	10%	449.28	670.53	976.28	2571.22	1.00	41.3%	5	2	7
40	25%	40	25%	449.63	1010.29	991.52	2910.34	1.03	48.1%	5	1	7
40	50%	40	50%	450.22	978.43	834.53	1122.54	0.49	21.6%	5	2	9
60	10%	60	10%	512.72	1643.43	978.62	2914.77	1.00	53.4%	5	1	7
60	25%	60	25%	513.51	1785.75	820.93	2423.78	0.46	23.4%	3	1	9
60	50%	60	50%	553.27	89.13	836.53	1455.26	0.49	22.7%	5	1	9
All				465.53	432.17	707.31	1838.59	0.37	20.1%	137	61	211

Table 50 – Average results from the CF formulation for the RVRPTW instances of class RC1 with the single knapsack uncertainty set

						RC1						
				Linear Relaxation		Integer Solution						
Δ	***Dev*^{d}**	**Δ**	***Dev*^{t}**	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	301.22	0.053	372.72	4.43	0.0%	0.0%	5	5	0
20	10%	0	0%	301.69	6.324	395.28	2800.67	7.0%	7.7%	5	2	0
20	25%	0	0%	301.93	4.775	395.50	2154.65	6.9%	5.7%	5	3	0
20	50%	0	0%	302.29	2.566	399.24	1767.80	8.2%	6.6%	5	3	0
40	10%	0	0%	302.30	21.610	415.04	2952.10	12.8%	17.7%	5	1	0
40	25%	0	0%	302.62	18.689	490.90	3020.42	33.8%	28.2%	5	1	0
40	50%	0	0%	303.24	11.214	483.66	2986.83	31.9%	24.8%	5	1	0
60	10%	0	0%	303.12	41.627	421.28	2977.26	13.9%	19.1%	5	1	0
60	25%	0	0%	303.68	52.913	508.56	3278.22	38.9%	30.5%	5	1	0
60	50%	0	0%	304.46	27.919	498.56	3192.18	36.1%	28.3%	5	1	0
0	0%	20	10%	301.41	41.049	410.84	3024.75	10.4%	17.0%	5	1	0
0	0%	20	25%	301.48	48.421	410.28	3042.84	10.4%	16.5%	5	1	0
0	0%	20	50%	301.52	45.501	393.98	2343.37	9.5%	14.6%	5	1	1
0	0%	40	10%	301.50	257.968	1600.82	3601.09	355.2%	71.2%	5	0	0
0	0%	40	25%	301.65	328.025	1603.26	3601.01	355.7%	71.7%	5	0	0
0	0%	40	50%	301.78	552.729	1530.88	2880.86	354.6%	68.3%	5	0	1
0	0%	60	10%	301.54	748.311	1609.50	3601.32	357.0%	73.0%	5	0	0
0	0%	60	25%	301.89	996.315	1552.93	3601.73	320.3%	71.5%	4	0	0
0	0%	60	50%	319.43	587.000	1186.05	2401.23	220.7%	53.8%	3	0	1
20	10%	20	10%	301.86	74.45	1020.36	3083.59	1.78	45.5%	5	1	0
20	25%	20	25%	302.17	61.83	783.90	3181.36	1.29	39.8%	5	1	0
20	50%	20	50%	302.55	52.96	819.60	2337.50	1.47	37.3%	5	1	1
40	10%	40	10%	302.54	524.46	1600.82	3600.80	3.55	71.2%	5	0	0
40	25%	40	25%	302.95	575.85	1623.42	3600.94	3.60	74.6%	5	0	0
40	50%	40	50%	303.65	423.57	1545.80	2880.77	3.58	70.2%	5	0	1
60	10%	60	10%	310.34	1750.32	1884.40	3770.18	4.17	86.6%	5	0	0
60	25%	60	25%	320.77	1985.28	1884.40	3730.01	4.17	89.7%	5	0	0
60	50%	60	50%	312.27	1939.85	1884.40	2400.86	3.72	82.4%	3	0	1
All				304.21	399.34	990.23	2922.10	1.76	43.7%	135	25	6

Table 51 – Average results from the CF formulation for the RVRPTW instances of class C2 with the single knapsack uncertainty set

						C2						
				Linear Relaxation		Integer Solution						
Δ	***Dev*^{d}**	**Δ**	***Dev*^{t}**	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	183.34	0.043	214.38	2.48	0.0%	0.0%	5	5	0
20	10%	0	0%	183.34	8.754	214.66	1533.99	0.1%	2.5%	5	4	0
20	25%	0	0%	183.34	8.142	214.54	1309.26	0.1%	0.3%	5	4	0
20	50%	0	0%	183.34	4.781	214.38	846.94	0.0%	0.0%	5	5	0
40	10%	0	0%	183.34	31.464	227.42	2163.33	6.1%	11.0%	5	2	0
40	25%	0	0%	183.34	44.528	227.22	2163.19	6.0%	10.9%	5	2	0
40	50%	0	0%	183.34	24.848	221.74	1670.16	3.4%	8.0%	5	3	0
60	10%	0	0%	183.34	71.084	223.28	2164.19	4.1%	10.3%	5	2	0
60	25%	0	0%	183.34	93.527	229.58	2167.34	7.1%	12.1%	5	2	0
60	50%	0	0%	183.34	59.177	217.14	2163.79	1.3%	8.1%	5	2	0
0	0%	20	10%	183.35	114.563	438.72	2202.65	105.2%	27.1%	5	2	0
0	0%	20	25%	183.35	134.442	253.66	2242.04	18.3%	21.1%	5	2	0
0	0%	20	50%	183.35	126.862	438.72	2245.12	105.2%	27.1%	5	2	0
0	0%	40	10%	194.75	1607.497	1061.34	2924.01	395.2%	74.1%	5	1	0
0	0%	40	25%	194.75	1660.570	1061.34	2960.23	395.2%	74.1%	5	1	0
0	0%	40	50%	194.75	1657.968	1061.34	2937.05	395.2%	74.1%	5	1	0
0	0%	60	10%	194.75	2063.248	1061.34	3032.79	395.2%	76.7%	5	1	0
0	0%	60	25%	203.73	1454.393	920.23	2428.61	328.6%	57.0%	3	1	0
0	0%	60	50%	203.73	456.092	743.85	1827.05	246.5%	42.4%	2	1	0
20	10%	20	10%	183.35	149.05	239.06	2214.35	0.12	16.8%	5	2	0
20	25%	20	25%	183.35	136.49	239.06	2238.00	0.12	17.1%	5	2	0
20	50%	20	50%	183.35	151.86	239.06	2245.28	0.12	17.3%	5	2	0
40	10%	40	10%	194.75	1793.82	849.68	2796.10	2.97	57.2%	5	2	0
40	25%	40	25%	194.75	1763.36	1061.34	2998.36	3.95	73.9%	5	1	0
40	50%	40	50%	194.75	1699.58	849.68	2979.51	2.97	57.5%	5	1	0
60	10%	60	10%	203.73	2292.11	1061.34	3055.71	3.95	76.7%	5	1	0
60	25%	60	25%	203.73	2359.98	920.23	2498.14	3.29	61.6%	3	1	0
60	50%	60	50%	203.73	2056.71	743.85	1902.12	2.46	42.4%	2	1	0
All				189.84	786.61	551.72	2211.14	1.57	34.2%	130	56	0

Table 52 – Average results from the CF formulation for the RVRPTW instances of class R2 with the single knapsack uncertainty set

R2												
				Linear Relaxation		**Integer Solution**						
Δ	***Dev*^{d}**	**Δ**	***Dev*^{t}**	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	335.56	0.048	402.64	70.24	0.0%	0.0%	5	5	0
20	10%	0	0%	335.56	2.372	407.38	2136.13	1.3%	5.8%	5	3	0
20	25%	0	0%	335.56	2.863	407.30	1946.55	1.3%	5.8%	5	3	0
20	50%	0	0%	335.56	2.599	406.02	2076.30	0.9%	4.8%	5	3	0
40	10%	0	0%	335.56	9.039	409.98	2186.72	2.0%	9.1%	5	2	0
40	25%	0	0%	335.56	10.982	410.52	2224.25	2.1%	9.8%	5	2	0
40	50%	0	0%	335.56	10.047	407.38	2190.19	1.3%	9.1%	5	2	0
60	10%	0	0%	335.56	23.285	414.00	2325.15	3.1%	10.9%	5	2	0
60	25%	0	0%	335.56	30.351	411.74	2857.19	2.5%	10.8%	5	2	0
60	50%	0	0%	335.56	34.713	412.04	2344.34	2.5%	10.9%	5	2	0
0	0%	20	10%	335.58	103.771	587.10	2892.17	47.4%	24.6%	5	1	0
0	0%	20	25%	335.59	101.351	446.44	2893.91	11.2%	18.0%	5	1	0
0	0%	20	50%	335.59	104.726	584.46	2891.61	46.4%	24.0%	5	1	0
0	0%	40	10%	335.60	1248.892	1088.34	3011.71	177.7%	64.7%	5	1	0
0	0%	40	25%	335.61	1110.017	1088.34	3050.74	177.7%	64.7%	5	1	0
0	0%	40	50%	335.62	1262.655	1088.34	3004.19	177.7%	64.7%	5	1	0
0	0%	60	10%	383.59	2638.823	984.17	3601.14	140.0%	66.7%	3	1	0
0	0%	60	25%	423.08	155.161	463.30	3305.82	0.0%	0.0%	1	1	0
0	0%	60	50%	423.11	173.569	862.70	3602.50	110.2%	38.4%	2	0	0
20	10%	20	10%	335.58	118.97	421.14	2922.213	0.05	14.5%	5	1	0
20	25%	20	25%	335.59	118.52	753.86	2914.33	0.94	34.9%	5	1	0
20	50%	20	50%	335.59	106.89	752.80	2913.85	0.94	34.5%	5	1	0
40	10%	40	10%	335.60	797.10	1088.34	3357.53	1.78	59.8%	5	1	0
40	25%	40	25%	335.61	1280.20	1088.34	3282.86	1.78	69.9%	5	1	0
40	50%	40	50%	335.62	1144.70	1088.34	3151.38	1.78	59.8%	5	1	0
60	10%	60	10%	383.59	2619.19	1049.63	3738.65	1.60	68.9%	4	0	0
60	25%	60	25%	383.60	2704.51	1244.60	3651.42	1.93	82.0%	2	0	0
60	50%	60	50%	423.11	118.95	480.80	3601.99	0.04	5.1%	1	0	0
All				350.10	572.65	687.50	2790.90	0.71	31.1%	123	40	0

Table 53 – Average results from the CF formulation for the RVRPTW instances of class RC2 with the single knapsack uncertainty set

RC2												
				Linear Relaxation		**Integer Solution**						
Δ	***Dev*^{d}**	**Δ**	***Dev*^{t}**	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	191.38	0.053	332.56	1459.76	0.0%	5.2%	5	3	0
20	10%	0	0%	191.38	5.343	335.46	2505.23	0.9%	21.8%	5	2	0
20	25%	0	0%	191.38	4.231	333.72	2304.98	0.4%	11.1%	5	2	0
20	50%	0	0%	191.38	3.662	339.36	2245.84	2.1%	11.4%	5	2	0
40	10%	0	0%	191.38	22.271	346.80	2344.55	4.4%	27.7%	5	2	0
40	25%	0	0%	191.38	43.845	342.90	2264.61	3.2%	25.7%	5	2	0
40	50%	0	0%	191.38	14.878	336.16	2275.15	1.1%	21.2%	5	2	0
60	10%	0	0%	191.38	53.377	342.48	2885.91	3.1%	31.6%	5	1	0
60	25%	0	0%	191.38	39.406	350.12	2886.14	5.4%	33.2%	5	1	0
60	50%	0	0%	191.38	23.068	337.64	2884.33	1.6%	29.4%	5	1	0
0	0%	20	10%	191.43	86.603	346.00	2893.07	4.2%	33.1%	5	1	0
0	0%	20	25%	191.45	107.834	661.86	2892.88	109.5%	42.2%	5	1	0
0	0%	20	50%	191.48	123.126	645.72	2902.31	95.8%	40.8%	5	1	0
0	0%	40	10%	191.44	694.032	1579.56	2989.97	384.0%	72.7%	5	1	0
0	0%	40	25%	191.48	894.960	1579.56	3153.71	384.0%	72.8%	5	1	0
0	0%	40	50%	191.53	950.104	1579.56	3076.57	384.0%	72.7%	5	1	0
0	0%	60	10%	204.77	2130.241	1884.40	3686.23	446.1%	91.3%	3	0	0
0	0%	60	25%	224.44	2097.886	1376.67	3701.86	305.1%	63.8%	3	0	0
0	0%	60	50%	283.55	477.955	1153.00	3691.50	237.3%	58.8%	2	0	0
20	10%	20	10%	191.43	135.80	366.70	2908.13	0.10	35.7%	5	1	0
20	25%	20	25%	191.45	117.37	664.00	2918.74	1.02	44.5%	5	1	0
20	50%	20	50%	191.48	128.18	963.22	2927.50	2.02	50.8%	5	1	0
40	10%	40	10%	191.44	1022.71	1579.56	3208.37	3.84	72.8%	5	1	0
40	25%	40	25%	191.48	1159.42	1579.56	3381.09	3.84	74.4%	5	1	0
40	50%	40	50%	191.53	989.90	1579.56	3272.04	3.84	72.8%	5	1	0
60	10%	60	10%	224.37	2246.76	1884.40	3761.45	4.54	91.1%	4	0	0
60	25%	60	25%	246.62	2886.98	1884.40	3681.25	4.40	92.2%	2	0	0
60	50%	60	50%	283.55	2994.36	1884.40	3761.83	4.40	92.2%	2	0	0
All				202.81	694.80	949.62	2959.46	1.85	49.8%	126	30	0

APPENDIX C – Results from the B&C algorithm for the literature instances

C.1 Cardinality-Constrained Uncertainty set

In this section, we present in more details the results of our computational experiments in instances of the literature for the cardinality constrained uncertainty set using the B&C algorithm. We created tables for each model (MTZ and CF) showing the relevant information for each set of instances separately. In these tables, for every combination of Γ^t, Γ^d and deviation (*Dev*) studied, we present the average solution value (*Sol*) and the time taken to solve the problem (T) of both the linear relaxation and integer problem, and the Price-of-Robustness (*PoR*), which represents how much the robust solution costs more than the deterministic solution, expressed in percentage. To better evaluate the computational results, we also present the percentage gap between the best solution obtained in the time limit and its lower bound (*Gap*), if the solution is optimal this difference is zero. Finally, we also present the number of instances (*Ins*) evaluated in that combination of Γ^q, Γ^t and deviations, how many of those were solved to optimality by the model within the time limit (*Opt*) and the number of infeasible instances (*Inf*) in that group.

C.1.1 MTZ-based formulation

Table 54 – Average results from the MTZ formulation for the RVRPTW instances of class C1

						C1						
				Linear Relaxation		**Integer Solution**						
Γ	***Dev^d***	**Γ**	***Dev^t***	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	76.75	0.011	190.59	0.15	0.0%	0.0%	9	9	0
1	10%	0	0%	76.75	0.010	190.97	0.12	0.2%	0.0%	9	9	0
1	25%	0	0%	76.75	0.011	190.97	0.11	0.2%	0.0%	9	9	0
1	50%	0	0%	76.75	0.019	226.10	0.74	18.6%	0.0%	9	9	0
5	10%	0	0%	76.75	0.020	226.10	0.66	18.6%	0.0%	9	9	0
5	25%	0	0%	76.75	0.019	234.84	16.39	23.2%	0.0%	9	9	0
5	50%	0	0%	76.75	0.030	254.57	486.39	33.6%	0.1%	9	8	0
10	10%	0	0%	76.75	0.032	234.07	84.46	22.8%	0.0%	9	9	0
10	25%	0	0%	76.75	0.035	252.39	1260.77	32.4%	1.1%	9	6	0
10	50%	0	0%	76.76	0.010	256.18	448.63	34.4%	0.1%	9	8	0
0	0%	1	10%	76.76	0.011	190.59	0.16	0.0%	0.0%	9	9	0
0	0%	1	25%	76.78	0.011	190.59	0.11	0.0%	0.0%	9	9	0
0	0%	1	50%	76.77	0.022	195.84	0.12	2.7%	0.0%	9	9	0
0	0%	5	10%	76.78	0.022	190.59	0.10	0.0%	0.0%	9	9	0
0	0%	5	25%	76.81	0.022	190.59	0.10	0.0%	0.0%	9	9	0
0	0%	5	50%	76.77	0.041	206.30	0.18	8.2%	0.0%	9	9	0
0	0%	10	10%	76.79	0.041	190.59	0.12	0.0%	0.0%	9	9	0
0	0%	10	25%	76.83	0.043	190.59	0.12	0.0%	0.0%	9	9	0
0	0%	10	50%	76.76	0.012	206.30	0.17	8.2%	0.0%	9	9	0
1	10%	1	10%	76.76	0.01	190.97	0.17	0.00	0.0%	9	9	0
1	25%	1	25%	76.78	0.01	190.97	0.15	0.00	0.0%	9	9	0
1	50%	1	50%	76.77	0.04	226.66	0.75	0.19	0.0%	9	9	0
5	10%	5	10%	76.78	0.04	226.32	0.80	0.19	0.0%	9	9	0
5	25%	5	25%	76.81	0.04	235.24	10.06	0.23	0.0%	9	9	0
5	50%	5	50%	76.77	0.08	255.22	492.96	0.34	0.2%	9	8	0
10	10%	10	10%	76.79	0.09	234.07	70.78	0.23	0.0%	9	9	0
10	25%	10	25%	76.83	0.08	253.03	1255.82	0.33	1.2%	9	6	0
10	50%	10	50%	84.15	0.00	257.47	511.05	0.35	0.0%	9	8	0
All				77.04	0.03	217.10	165.79	0.14	0.1%	252	242	0

Table 55 – Average results from the MTZ formulation for the RVRPTW instances of class R1

						R1						
				Linear Relaxation		**Integer Solution**						
Γ	***Dev^d***	**Γ**	***Dev^t***	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	290.11	0.011	463.37	63.52	0.0%	0.0%	12	12	0
1	10%	0	0%	290.11	0.011	463.37	235.56	0.0%	0.0%	12	12	0
1	25%	0	0%	290.11	0.010	463.37	203.29	0.0%	0.0%	12	12	0
1	50%	0	0%	290.11	0.019	463.37	318.13	0.0%	0.0%	12	12	0
5	10%	0	0%	290.11	0.019	463.37	177.98	0.0%	0.0%	12	12	0
5	25%	0	0%	290.11	0.019	463.37	368.98	0.0%	0.2%	12	11	0
5	50%	0	0%	290.11	0.036	463.37	214.18	0.0%	0.0%	12	12	0
10	10%	0	0%	290.11	0.033	463.37	208.42	0.0%	0.0%	12	12	0
10	25%	0	0%	290.11	0.036	463.37	360.47	0.0%	0.2%	12	11	0
10	50%	0	0%	290.27	0.012	463.37	29.41	0.0%	0.0%	12	12	0
0	0%	1	10%	290.50	0.011	466.89	94.92	0.8%	0.0%	12	12	0
0	0%	1	25%	290.96	0.011	477.74	142.12	3.0%	0.0%	12	12	0
0	0%	1	50%	290.49	0.025	464.38	94.99	5.1%	0.0%	12	8	4
0	0%	5	10%	291.06	0.026	469.31	104.39	1.3%	0.0%	12	12	0
0	0%	5	25%	292.14	0.028	481.78	293.74	3.9%	0.0%	12	12	0
0	0%	5	50%	290.54	0.049	487.69	399.42	10.5%	1.2%	12	7	4
0	0%	10	10%	291.20	0.049	470.08	325.24	1.5%	0.3%	12	11	0
0	0%	10	25%	292.52	0.053	481.78	339.48	3.9%	0.1%	12	11	0
0	0%	10	50%	290.27	0.014	488.18	358.77	10.6%	1.1%	12	7	4
1	10%	1	10%	290.50	0.01	466.89	266.50	0.01	0.0%	12	12	0
1	25%	1	25%	290.96	0.01	477.74	517.04	0.03	0.0%	12	12	0
1	50%	1	50%	290.49	0.04	464.38	383.74	0.05	0.0%	12	8	4
5	10%	5	10%	291.06	0.04	469.31	309.51	0.01	0.0%	12	12	0
5	25%	5	25%	292.14	0.05	481.78	623.96	0.04	0.4%	12	11	0
5	50%	5	50%	290.54	0.10	487.51	918.09	0.10	1.8%	12	6	4
10	10%	10	10%	291.20	0.10	470.18	361.99	0.02	0.6%	12	11	0
10	25%	10	25%	292.52	0.10	482.48	685.12	0.04	0.6%	12	11	0
10	50%	10	50%	286.63	0.00	488.18	509.05	0.11	1.1%	12	7	4
All				290.61	0.03	471.78	318.14	0.03	0.3%	336	300	24

Table 56 – Average results from the MTZ formulation for the RVRPTW instances of class RC1

RC1												
				Linear Relaxation		**Integer Solution**						
Γ	***Dev***^{d}	**Γ**	***Dev***^{t}	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	95.04	0.011	350.24	3.88	0.0%	0.0%	8	8	0
1	10%	0	0%	95.04	0.010	350.24	3.36	0.0%	0.0%	8	8	0
1	25%	0	0%	95.04	0.010	350.24	3.90	0.0%	0.0%	8	8	0
1	50%	0	0%	95.04	0.019	379.94	20.53	9.4%	0.0%	8	8	0
5	10%	0	0%	95.04	0.020	379.94	35.37	9.4%	0.0%	8	8	0
5	25%	0	0%	95.04	0.019	455.46	173.89	31.8%	0.0%	8	8	0
5	50%	0	0%	95.04	0.034	461.11	23.48	33.4%	0.0%	8	8	0
10	10%	0	0%	95.04	0.035	420.09	1946.83	21.6%	6.1%	8	4	0
10	25%	0	0%	95.04	0.035	456.16	5.01	32.0%	0.0%	8	8	0
10	50%	0	0%	95.06	0.011	479.01	1476.00	39.1%	1.5%	8	5	0
0	0%	1	10%	95.08	0.011	350.91	3.26	0.1%	0.0%	8	8	0
0	0%	1	25%	95.11	0.011	352.91	5.15	0.7%	0.0%	8	8	0
0	0%	1	50%	95.06	0.021	368.19	7.71	7.1%	0.0%	8	7	1
0	0%	5	10%	95.08	0.019	352.58	3.41	0.6%	0.0%	8	8	0
0	0%	5	25%	95.13	0.020	377.04	32.15	7.1%	0.0%	8	8	0
0	0%	5	50%	95.06	0.039	424.87	1419.19	24.6%	5.5%	8	4	1
0	0%	10	10%	95.09	0.042	352.58	3.70	0.6%	0.0%	8	8	0
0	0%	10	25%	95.14	0.040	382.59	28.58	8.6%	0.0%	8	8	0
0	0%	10	50%	95.06	0.012	450.36	2258.27	32.9%	9.8%	8	2	1
1	10%	1	10%	95.08	0.01	350.91	2.68	0.00	0.0%	8	8	0
1	25%	1	25%	95.11	0.01	352.91	2.73	0.01	0.0%	8	8	0
1	50%	1	50%	95.06	0.04	406.50	995.47	0.19	1.2%	8	5	1
5	10%	5	10%	95.08	0.04	385.65	344.77	0.11	0.0%	8	8	0
5	25%	5	25%	95.13	0.04	466.35	413.32	0.35	0.0%	8	8	0
5	50%	5	50%	95.06	0.08	486.10	1814.73	0.44	3.0%	8	3	1
10	10%	10	10%	95.09	0.08	425.24	1374.60	0.21	5.2%	8	4	0
10	25%	10	25%	95.14	0.08	467.69	244.72	0.35	0.0%	8	8	0
10	50%	10	50%	95.17	0.00	491.57	1812.10	0.46	3.3%	8	3	1
All				95.08	0.03	404.55	516.39	0.17	1.3%	224	189	6

Table 57 – Average results from the MTZ formulation for the RVRPTW instances of class C2

C2												
				Linear Relaxation		**Integer Solution**						
Γ	***Dev***^{d}	**Γ**	***Dev***^{t}	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	144.57	0.010	214.45	0.95	0.0%	0.0%	8	8	0
1	10%	0	0%	144.57	0.010	214.45	1.06	0.0%	0.0%	8	7	0
1	25%	0	0%	144.57	0.010	214.45	1.45	0.0%	0.0%	8	8	0
1	50%	0	0%	144.57	0.018	214.45	1.27	0.0%	0.0%	8	8	0
5	10%	0	0%	144.57	0.019	214.45	1.91	0.0%	0.0%	8	8	0
5	25%	0	0%	144.57	0.018	214.45	0.92	0.0%	0.0%	8	8	0
5	50%	0	0%	144.57	0.031	214.45	1.27	0.0%	0.0%	8	8	0
10	10%	0	0%	144.57	0.032	214.45	1.86	0.0%	0.0%	8	8	0
10	25%	0	0%	144.57	0.035	214.45	1.19	0.0%	0.0%	8	8	0
10	50%	0	0%	144.57	0.010	214.45	1.36	0.0%	0.0%	8	8	0
0	0%	1	10%	144.58	0.011	214.45	0.85	0.0%	0.0%	8	8	0
0	0%	1	25%	144.58	0.010	214.45	0.93	0.0%	0.0%	8	8	0
0	0%	1	50%	144.57	0.018	214.63	0.97	0.1%	0.0%	8	8	0
0	0%	5	10%	144.58	0.018	214.45	1.25	0.0%	0.0%	8	8	0
0	0%	5	25%	144.59	0.020	214.63	1.65	0.1%	0.0%	8	8	0
0	0%	5	50%	144.58	0.037	214.63	1.35	0.1%	0.0%	8	8	0
0	0%	10	10%	144.58	0.035	214.45	1.35	0.0%	0.0%	8	8	0
0	0%	10	25%	144.60	0.036	214.63	1.39	0.1%	0.0%	8	8	0
0	0%	10	50%	144.57	0.012	214.63	1.38	0.1%	0.0%	8	8	0
1	10%	1	10%	144.58	0.01	214.45	1.87	0.00	0.0%	8	8	0
1	25%	1	25%	144.58	0.01	214.45	1.28	0.00	0.0%	8	8	0
1	50%	1	50%	144.57	0.03	214.63	1.72	0.00	0.0%	8	8	0
5	10%	5	10%	144.58	0.03	214.45	1.29	0.00	0.0%	8	8	0
5	25%	5	25%	144.59	0.03	214.63	1.96	0.00	0.0%	8	8	0
5	50%	5	50%	144.58	0.08	214.63	1.49	0.00	0.0%	8	8	0
10	10%	10	10%	144.58	0.07	214.45	1.90	0.00	0.0%	8	8	0
10	25%	10	25%	144.60	0.07	214.63	1.23	0.00	0.0%	8	8	0
10	50%	10	50%	166.74	0.00	214.63	1.76	0.00	0.0%	8	8	0
All				145.37	0.03	214.51	1.39	0.00	0.0%	224	223	0

Table 58 – Average results from the MTZ formulation for the RVRPTW instances of class R2

						R2						
				Linear Relaxation		**Integer Solution**						
Γ	Dev^d	**Γ**	Dev^t	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	279.82	0.011	382.15	18.70	0.0%	0.0%	11	11	0
1	10%	0	0%	279.82	0.010	382.15	27.61	0.0%	0.0%	11	11	0
1	25%	0	0%	279.82	0.010	382.15	29.35	0.0%	0.0%	11	11	0
1	50%	0	0%	279.82	0.019	382.15	33.68	0.0%	0.0%	11	11	0
5	10%	0	0%	279.82	0.020	382.15	28.70	0.0%	0.0%	11	11	0
5	25%	0	0%	279.82	0.020	382.15	34.79	0.0%	0.0%	11	11	0
5	50%	0	0%	279.82	0.034	382.15	40.03	0.0%	0.0%	11	11	0
10	10%	0	0%	279.82	0.033	382.15	34.06	0.0%	0.0%	11	11	0
10	25%	0	0%	279.82	0.033	382.15	25.83	0.0%	0.0%	11	11	0
10	50%	0	0%	279.83	0.011	382.15	26.79	0.0%	0.0%	11	11	0
0	0%	1	10%	279.85	0.011	382.15	17.23	0.0%	0.0%	11	11	0
0	0%	1	25%	279.88	0.010	383.55	19.68	0.4%	0.0%	11	11	0
0	0%	1	50%	279.83	0.020	385.88	19.47	0.9%	0.0%	11	11	0
0	0%	5	10%	279.85	0.020	383.55	23.96	0.4%	0.0%	11	11	0
0	0%	5	25%	279.88	0.021	383.73	22.90	0.5%	0.0%	11	11	0
0	0%	5	50%	279.83	0.041	387.14	31.53	1.3%	0.0%	11	11	0
0	0%	10	10%	279.85	0.039	383.55	22.14	0.4%	0.0%	11	11	0
0	0%	10	25%	279.88	0.040	383.98	23.45	0.5%	0.0%	11	11	0
0	0%	10	50%	279.83	0.013	387.14	35.96	1.3%	0.0%	11	11	0
1	10%	1	10%	279.85	0.01	382.15	36.645	0.00	0.0%	11	11	0
1	25%	1	25%	279.88	0.01	383.55	38.49	0.00	0.0%	11	11	0
1	50%	1	50%	279.83	0.03	385.88	27.79	0.01	0.0%	11	11	0
5	10%	5	10%	279.85	0.04	383.55	36.23	0.00	0.0%	11	11	0
5	25%	5	25%	279.88	0.04	383.73	38.66	0.00	0.0%	11	11	0
5	50%	5	50%	279.83	0.08	387.14	55.07	0.01	0.0%	11	11	0
10	10%	10	10%	279.85	0.08	383.55	26.79	0.00	0.0%	11	11	0
10	25%	10	25%	279.88	0.08	383.98	35.11	0.01	0.0%	11	11	0
10	50%	10	50%	262.45	0.00	387.14	68.53	0.01	0.0%	11	11	0
All				279.22	0.03	383.67	31.40	0.00	0.0%	308	308	0

Table 59 – Average results from the MTZ formulation for the RVRPTW instances of class RC2

						RC2						
				Linear Relaxation		**Integer Solution**						
Γ	Dev^d	**Γ**	Dev^t	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	94.78	0.010	319.28	1442.09	0.0%	4.5%	8	5	0
1	10%	0	0%	94.78	0.010	319.28	1455.93	0.0%	6.1%	8	5	0
1	25%	0	0%	94.78	0.010	319.28	1451.95	0.0%	5.6%	8	5	0
1	50%	0	0%	94.78	0.020	319.28	1429.73	0.0%	6.2%	8	5	0
5	10%	0	0%	94.78	0.019	319.28	1413.22	0.0%	5.6%	8	5	0
5	25%	0	0%	94.78	0.019	319.36	1485.67	0.0%	5.9%	8	5	0
5	50%	0	0%	94.78	0.035	319.28	1404.33	0.0%	5.8%	8	5	0
10	10%	0	0%	94.78	0.035	319.28	1398.92	0.0%	5.9%	8	5	0
10	25%	0	0%	94.78	0.035	319.28	1410.36	0.0%	5.5%	8	5	0
10	50%	0	0%	94.78	0.010	319.28	1416.31	0.0%	5.4%	8	5	0
0	0%	1	10%	94.79	0.010	319.28	1446.96	0.0%	4.5%	8	5	0
0	0%	1	25%	94.79	0.011	319.76	1447.30	0.2%	4.7%	8	5	0
0	0%	1	50%	94.78	0.020	320.03	1447.14	0.3%	4.6%	8	5	0
0	0%	5	10%	94.79	0.019	319.66	1418.02	0.1%	5.0%	8	5	0
0	0%	5	25%	94.79	0.019	319.76	1417.95	0.2%	5.0%	8	5	0
0	0%	5	50%	94.78	0.038	320.05	1418.88	0.3%	5.4%	8	5	0
0	0%	10	10%	94.79	0.037	319.76	1426.41	0.2%	5.0%	8	5	0
0	0%	10	25%	94.79	0.036	319.76	1428.30	0.2%	5.1%	8	5	0
0	0%	10	50%	94.78	0.012	320.53	1444.24	0.5%	4.8%	8	5	0
1	10%	1	10%	94.79	0.01	319.95	1441.03	0.00	6.1%	8	5	0
1	25%	1	25%	94.79	0.01	319.50	1399.29	0.00	6.0%	8	5	0
1	50%	1	50%	94.78	0.03	319.89	1423.57	0.00	6.3%	8	5	0
5	10%	5	10%	94.79	0.03	319.36	1416.06	0.00	5.8%	8	5	0
5	25%	5	25%	94.79	0.04	319.76	1461.96	0.00	6.0%	8	5	0
5	50%	5	50%	94.78	0.08	322.45	1434.42	0.01	6.9%	8	5	0
10	10%	10	10%	94.79	0.08	319.76	1412.55	0.00	5.6%	8	5	0
10	25%	10	25%	94.79	0.07	320.01	1428.87	0.00	6.2%	8	5	0
10	50%	10	50%	82.71	0.00	322.41	1426.30	0.01	6.9%	8	5	0
All				94.35	0.03	319.80	1430.28	0.00	5.6%	224	140	0

C.1.2 CF formulation

Table 60 – Average results from the CF formulation for the RVRPTW instances of class C1

				C1								
				Linear Relaxation		**Integer Solution**						
Γ	***Dev^d***	**Γ**	***Dev^t***	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	177.96	0.030	190.59	0.18	0.0%	0.0%	9	9	0
1	10%	0	0%	178.31	0.055	190.97	0.23	0.2%	0.0%	9	9	0
1	25%	0	0%	178.97	0.057	190.97	0.24	0.2%	0.0%	9	9	0
1	50%	0	0%	180.38	0.055	226.10	1.14	18.6%	0.0%	9	9	0
5	10%	0	0%	178.96	0.168	226.10	2.57	18.6%	0.0%	9	9	0
5	25%	0	0%	181.73	0.206	234.84	57.09	23.2%	0.0%	9	9	0
5	50%	0	0%	187.54	0.224	254.61	645.26	33.6%	0.3%	9	8	0
10	10%	0	0%	179.52	0.418	234.07	178.41	22.8%	0.0%	9	9	0
10	25%	0	0%	184.22	0.447	252.74	1534.30	32.6%	1.5%	9	6	0
10	50%	0	0%	195.80	0.647	256.31	755.95	34.5%	0.4%	9	8	0
0	0%	1	10%	177.98	0.077	190.59	0.19	0.0%	0.0%	9	9	0
0	0%	1	25%	178.00	0.067	190.59	0.16	0.0%	0.0%	9	9	0
0	0%	1	50%	178.07	0.077	195.84	0.19	2.7%	0.0%	9	9	0
0	0%	5	10%	177.99	0.255	190.59	0.27	0.0%	0.0%	9	9	0
0	0%	5	25%	178.05	0.264	190.59	0.27	0.0%	0.0%	9	9	0
0	0%	5	50%	178.35	0.275	206.30	0.82	8.2%	0.0%	9	9	0
0	0%	10	10%	178.00	0.964	190.59	0.40	0.0%	0.0%	9	9	0
0	0%	10	25%	178.07	0.831	190.59	0.40	0.0%	0.0%	9	9	0
0	0%	10	50%	178.46	0.786	206.30	1.50	8.2%	0.0%	9	9	0
1	10%	1	10%	178.32	0.09	190.97	0.29	0.00	0.0%	9	9	0
1	25%	1	25%	179.00	0.09	190.97	0.29	0.00	0.0%	9	9	0
1	50%	1	50%	180.42	0.09	226.66	2.46	0.19	0.0%	9	9	0
5	10%	5	10%	178.98	0.43	226.32	2.95	0.19	0.0%	9	9	0
5	25%	5	25%	181.76	0.46	235.24	155.72	0.23	0.0%	9	9	0
5	50%	5	50%	187.61	0.46	255.33	1255.46	0.34	1.1%	9	6	0
10	10%	10	10%	179.55	2.93	234.07	522.01	0.23	0.1%	9	8	0
10	25%	10	25%	184.29	3.05	253.77	2003.69	0.33	2.6%	9	4	0
10	50%	10	50%	195.89	3.47	257.50	1374.11	0.35	1.1%	9	6	0
All				181.15	0.61	217.15	303.45	0.14	0.3%	252	235	0

Table 61 – Average results from the CF formulation for the RVRPTW instances of class R1

				R1								
				Linear Relaxation		**Integer Solution**						
Γ	***Dev^d***	**Γ**	***Dev^t***	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	396.37	0.027	463.37	31.21	0.0%	0.0%	12	12	0
1	10%	0	0%	396.37	0.052	463.37	57.78	0.0%	0.0%	12	12	0
1	25%	0	0%	396.37	0.061	463.37	59.31	0.0%	0.0%	12	12	0
1	50%	0	0%	396.37	0.060	463.37	67.83	0.0%	0.0%	12	12	0
5	10%	0	0%	396.37	0.179	463.37	90.52	0.0%	0.0%	12	12	0
5	25%	0	0%	396.37	0.205	463.37	50.13	0.0%	0.0%	12	12	0
5	50%	0	0%	396.37	0.194	463.37	81.59	0.0%	0.0%	12	12	0
10	10%	0	0%	396.37	0.419	463.37	122.98	0.0%	0.0%	12	12	0
10	25%	0	0%	396.37	0.370	463.37	131.70	0.0%	0.0%	12	12	0
10	50%	0	0%	396.37	0.478	463.37	140.05	0.0%	0.0%	12	12	0
0	0%	1	10%	396.72	0.095	466.89	33.98	0.8%	0.0%	12	12	0
0	0%	1	25%	397.45	0.086	477.74	112.77	3.0%	0.0%	12	12	0
0	0%	1	50%	310.62	0.103	464.38	35.94	5.1%	0.0%	12	8	4
0	0%	5	10%	397.73	0.283	469.31	136.87	1.3%	0.0%	12	12	0
0	0%	5	25%	400.71	0.311	481.78	418.78	3.9%	0.0%	12	12	0
0	0%	5	50%	318.68	0.321	486.99	960.05	10.3%	1.3%	12	6	4
0	0%	10	10%	398.43	1.067	470.03	426.19	1.5%	0.0%	12	12	0
0	0%	10	25%	403.09	0.760	481.78	733.35	3.9%	0.2%	12	11	0
0	0%	10	50%	323.88	0.720	488.18	1119.00	10.6%	2.3%	12	5	4
1	10%	1	10%	396.72	0.11	466.89	47.92	0.01	0.0%	12	12	0
1	25%	1	25%	397.45	0.11	477.74	182.82	0.03	0.0%	12	12	0
1	50%	1	50%	310.62	0.13	464.38	140.98	0.05	0.0%	12	8	4
5	10%	5	10%	397.73	0.54	469.31	203.74	0.01	0.0%	12	12	0
5	25%	5	25%	400.71	0.50	481.78	1289.12	0.04	0.2%	12	11	0
5	50%	5	50%	318.68	0.51	487.39	1277.81	0.10	2.8%	12	4	4
10	10%	10	10%	398.43	5.21	470.13	647.56	0.02	0.3%	12	11	0
10	25%	10	25%	403.09	4.36	481.78	1268.65	0.04	0.3%	12	11	0
10	50%	10	50%	323.88	5.13	488.18	1297.84	0.11	3.1%	12	4	4
All				380.65	0.80	471.72	398.80	0.03	0.4%	336	295	24

Table 62 – Average results from the CF formulation for the RVRPTW instances of class RC1

				RC1								
				Linear Relaxation		**Integer Solution**						
Γ	***Dev*^{d}**	**Γ**	***Dev*^{t}**	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	290.87	0.037	350.24	4.25	0.0%	0.0%	8	8	0
1	10%	0	0%	290.97	0.065	350.24	3.32	0.0%	0.0%	8	8	0
1	25%	0	0%	291.15	0.068	350.24	3.89	0.0%	0.0%	8	8	0
1	50%	0	0%	291.54	0.065	379.94	35.93	9.4%	0.0%	8	8	0
5	10%	0	0%	291.36	0.263	379.94	55.86	9.4%	0.0%	8	8	0
5	25%	0	0%	292.72	0.241	455.51	819.86	31.8%	0.1%	8	7	0
5	50%	0	0%	296.98	0.240	462.23	461.51	33.7%	0.5%	8	7	0
10	10%	0	0%	292.02	0.453	419.80	1968.98	21.5%	6.5%	8	4	0
10	25%	0	0%	296.31	0.711	456.16	27.68	32.0%	0.0%	8	8	0
10	50%	0	0%	313.37	0.568	478.86	1846.92	39.0%	2.4%	8	4	0
0	0%	1	10%	291.05	0.087	350.91	6.00	0.1%	0.0%	8	8	0
0	0%	1	25%	291.56	0.085	352.91	8.19	0.7%	0.0%	8	8	0
0	0%	1	50%	292.78	0.082	368.19	230.90	7.1%	0.0%	8	7	1
0	0%	5	10%	291.26	0.319	352.58	25.87	0.6%	0.0%	8	8	0
0	0%	5	25%	292.28	0.323	377.04	961.41	7.1%	1.5%	8	6	0
0	0%	5	50%	296.25	0.302	426.31	2015.64	25.0%	9.8%	8	3	1
0	0%	10	10%	291.38	1.021	352.58	17.94	0.6%	0.0%	8	8	0
0	0%	10	25%	292.78	0.865	382.59	1422.55	8.6%	1.9%	8	5	0
0	0%	10	50%	298.96	0.974	452.74	2283.21	33.6%	15.7%	8	2	1
1	10%	1	10%	291.14	0.12	350.91	4.86	0.00	0.0%	8	8	0
1	25%	1	25%	291.81	0.11	352.91	9.48	0.01	0.0%	8	8	0
1	50%	1	50%	293.30	0.12	406.70	1265.86	0.19	1.4%	8	5	1
5	10%	5	10%	291.72	0.62	385.65	859.94	0.11	0.6%	8	7	0
5	25%	5	25%	293.82	0.58	469.34	2829.67	0.36	2.9%	8	3	0
5	50%	5	50%	300.71	0.69	494.47	1957.53	0.47	5.6%	8	3	1
10	10%	10	10%	292.46	4.49	425.33	1862.55	0.21	6.9%	8	3	0
10	25%	10	25%	297.55	5.09	469.96	1124.73	0.36	1.0%	8	7	0
10	50%	10	50%	316.82	4.18	493.47	2071.15	0.47	4.7%	8	3	1
All				294.82	0.81	405.28	863.77	0.17	2.2%	224	172	6

Table 63 – Average results from the CF formulation for the RVRPTW instances of class C2

				C2								
				Linear Relaxation		**Integer Solution**						
Γ	***Dev*^{d}**	**Γ**	***Dev*^{t}**	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	181.59	0.025	214.45	1.67	0.0%	0.0%	8	8	0
1	10%	0	0%	181.59	0.052	214.45	1.85	0.0%	0.0%	8	8	0
1	25%	0	0%	181.59	0.056	214.45	1.76	0.0%	0.0%	8	8	0
1	50%	0	0%	181.59	0.060	214.45	1.01	0.0%	0.0%	8	8	0
5	10%	0	0%	181.59	0.197	214.45	2.73	0.0%	0.0%	8	8	0
5	25%	0	0%	181.59	0.208	214.45	2.69	0.0%	0.0%	8	8	0
5	50%	0	0%	181.59	0.211	214.45	2.74	0.0%	0.0%	8	8	0
10	10%	0	0%	181.59	0.407	214.45	3.96	0.0%	0.0%	8	8	0
10	25%	0	0%	181.59	0.369	214.45	3.40	0.0%	0.0%	8	8	0
10	50%	0	0%	181.59	0.487	214.45	4.40	0.0%	0.0%	8	8	0
0	0%	1	10%	181.59	0.072	214.45	1.35	0.0%	0.0%	8	8	0
0	0%	1	25%	181.60	0.072	214.45	1.49	0.0%	0.0%	8	8	0
0	0%	1	50%	181.61	0.079	214.63	1.73	0.1%	0.0%	8	8	0
0	0%	5	10%	181.60	0.237	214.45	2.72	0.0%	0.0%	8	8	0
0	0%	5	25%	181.61	0.250	214.63	2.93	0.1%	0.0%	8	8	0
0	0%	5	50%	181.64	0.258	214.63	2.85	0.1%	0.0%	8	8	0
0	0%	10	10%	181.60	0.769	214.45	2.93	0.0%	0.0%	8	8	0
0	0%	10	25%	181.62	0.608	214.63	2.99	0.1%	0.0%	8	8	0
0	0%	10	50%	181.66	0.646	214.63	2.90	0.1%	0.0%	8	8	0
1	10%	1	10%	181.59	0.10	214.45	2.27	0.00	0.0%	8	8	0
1	25%	1	25%	181.60	0.10	214.45	3.33	0.00	0.0%	8	8	0
1	50%	1	50%	181.61	0.10	214.63	2.69	0.00	0.0%	8	8	0
5	10%	5	10%	181.60	0.63	214.45	4.19	0.00	0.0%	8	8	0
5	25%	5	25%	181.61	0.58	214.63	4.15	0.00	0.0%	8	8	0
5	50%	5	50%	181.64	0.60	214.63	3.08	0.00	0.0%	8	8	0
10	10%	10	10%	181.60	3.06	214.45	7.36	0.00	0.0%	8	8	0
10	25%	10	25%	181.62	3.77	214.63	7.59	0.00	0.0%	8	8	0
10	50%	10	50%	181.66	3.76	214.63	8.68	0.00	0.0%	8	8	0
All				181.61	0.63	214.51	3.27	0.00	0.0%	224	224	0

Table 64 – Average results from the CF formulation for the RVRPTW instances of class R2

						R2						
				Linear Relaxation		**Integer Solution**						
Γ	***Dev***^{d}	**Γ**	***Dev***^{t}	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	313.97	0.025	382.15	66.45	0.0%	0.0%	11	11	0
1	10%	0	0%	313.97	0.058	382.15	99.15	0.0%	0.0%	11	11	0
1	25%	0	0%	313.97	0.056	382.15	119.73	0.0%	0.0%	11	11	0
1	50%	0	0%	313.97	0.058	382.15	155.75	0.0%	0.0%	11	11	0
5	10%	0	0%	313.97	0.215	382.15	220.93	0.0%	0.0%	11	11	0
5	25%	0	0%	313.97	0.187	382.15	194.51	0.0%	0.0%	11	11	0
5	50%	0	0%	313.97	0.178	382.15	314.87	0.0%	0.0%	11	11	0
10	10%	0	0%	313.97	0.433	382.15	195.37	0.0%	0.0%	11	11	0
10	25%	0	0%	313.97	0.441	382.15	224.58	0.0%	0.0%	11	11	0
10	50%	0	0%	313.97	0.514	382.15	279.05	0.0%	0.0%	11	11	0
0	0%	1	10%	313.99	0.085	382.15	89.98	0.0%	0.0%	11	11	0
0	0%	1	25%	314.02	0.075	383.55	122.09	0.4%	0.0%	11	11	0
0	0%	1	50%	314.09	0.086	385.88	139.14	0.9%	0.0%	11	11	0
0	0%	5	10%	314.03	0.292	383.55	125.79	0.4%	0.0%	11	11	0
0	0%	5	25%	314.14	0.304	383.73	225.32	0.5%	0.0%	11	11	0
0	0%	5	50%	314.37	0.323	387.14	440.56	1.3%	0.0%	11	11	0
0	0%	10	10%	314.05	0.692	383.55	320.93	0.4%	0.0%	11	11	0
0	0%	10	25%	314.21	0.554	383.98	517.92	0.5%	0.2%	11	10	0
0	0%	10	50%	314.58	0.537	387.70	669.63	1.5%	0.6%	11	10	0
1	10%	1	10%	313.99	0.11	382.15	209.1398	0.00	0.0%	11	11	0
1	25%	1	25%	314.02	0.10	383.55	145.36	0.00	0.0%	11	11	0
1	50%	1	50%	314.09	0.10	385.88	227.65	0.01	0.0%	11	11	0
5	10%	5	10%	314.03	0.46	383.55	250.92	0.00	0.0%	11	11	0
5	25%	5	25%	314.14	0.43	383.73	320.47	0.00	0.0%	11	11	0
5	50%	5	50%	314.37	0.42	387.14	822.89	0.01	0.1%	11	10	0
10	10%	10	10%	314.05	3.84	383.55	564.92	0.00	0.0%	11	11	0
10	25%	10	25%	314.21	3.47	384.24	680.68	0.01	0.4%	11	10	0
10	50%	10	50%	314.58	3.77	387.39	891.97	0.01	0.5%	11	9	0
All				314.09	0.64	383.71	308.42	0.00	0.1%	308	302	0

Table 65 – Average results from the CF formulation for the RVRPTW instances of class RC2

						RC2						
				Linear Relaxation		**Integer Solution**						
Γ	***Dev***^{d}	**Γ**	***Dev***^{t}	**Sol**	**T(s)**	**Sol**	**T(s)**	**PoR**	**Gap**	**Total**	**Opt**	**Inf**
0	0%	0	0%	173.10	0.038	319.28	1812.35	0.0%	4.8%	8	4	0
1	10%	0	0%	173.10	0.071	319.28	1816.81	0.0%	5.4%	8	4	0
1	25%	0	0%	173.10	0.072	319.28	1815.85	0.0%	4.9%	8	4	0
1	50%	0	0%	173.10	0.075	319.28	1813.39	0.0%	5.4%	8	4	0
5	10%	0	0%	173.10	0.238	319.28	1826.73	0.0%	5.6%	8	4	0
5	25%	0	0%	173.10	0.227	319.28	1826.09	0.0%	5.7%	8	4	0
5	50%	0	0%	173.10	0.212	319.28	1819.87	0.0%	6.1%	8	4	0
10	10%	0	0%	173.10	0.502	319.28	1829.96	0.0%	5.7%	8	4	0
10	25%	0	0%	173.10	0.551	319.28	1822.20	0.0%	5.8%	8	4	0
10	50%	0	0%	173.10	0.544	319.28	1840.59	0.0%	6.0%	8	4	0
0	0%	1	10%	173.13	0.087	319.28	1818.41	0.0%	4.9%	8	4	0
0	0%	1	25%	173.17	0.094	319.50	1819.30	0.1%	5.1%	8	4	0
0	0%	1	50%	173.24	0.085	320.04	1810.85	0.3%	5.2%	8	4	0
0	0%	5	10%	173.16	0.339	319.36	1818.24	0.0%	5.1%	8	4	0
0	0%	5	25%	173.24	0.345	319.84	1818.29	0.2%	5.4%	8	4	0
0	0%	5	50%	173.44	0.321	320.81	1818.42	0.6%	5.8%	8	4	0
0	0%	10	10%	173.17	0.559	319.50	1825.46	0.1%	5.5%	8	4	0
0	0%	10	25%	173.31	0.511	320.01	1825.46	0.3%	5.5%	8	4	0
0	0%	10	50%	173.70	0.557	322.29	1825.65	1.1%	6.3%	8	4	0
1	10%	1	10%	173.13	0.11	319.28	1812.78	0.00	5.5%	8	4	0
1	25%	1	25%	173.17	0.11	319.76	1806.83	0.00	6.1%	8	4	0
1	50%	1	50%	173.24	0.11	319.95	1806.10	0.00	5.5%	8	4	0
5	10%	5	10%	173.16	0.47	319.36	1824.17	0.00	6.3%	8	4	0
5	25%	5	25%	173.24	0.44	320.03	1830.58	0.00	6.5%	8	4	0
5	50%	5	50%	173.44	0.50	322.45	1829.92	0.01	7.3%	8	4	0
10	10%	10	10%	173.17	3.61	319.85	1858.59	0.00	6.3%	8	4	0
10	25%	10	25%	173.31	3.73	321.20	1858.86	0.01	6.6%	8	4	0
10	50%	10	50%	173.70	4.08	322.41	1857.36	0.01	6.9%	8	4	0
All				173.22	0.66	319.92	1824.61	0.00	5.8%	224	112	0