# Robust vehicle routing problem with hard time windows under demand and travel time uncertainty

C. Hu[a,b]*, J. Lu[b], X. Liu[a,c], G. Zhang[b]

[a] *Decision Systems & e-Service Intelligence Lab, Center for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia*

[b] *Department of Industrial Engineering & Management, Shanghai Jiao Tong University, P.R. China*

[c] *Department of Industrial & System Engineering, National University of Singapore, Singapore*

**Abstract:** Due to an increase in customer-oriented service strategies designed to meet more complex and exacting customer requirements, meeting a scheduled time window has become an important part of designing vehicle routes for logistics activities. However, practically, the uncertainty in travel times and customer demand often means vehicles miss these time windows, increasing service costs and decreasing customer satisfaction. In an effort to find a solution that meets the needs of real-world logistics, we examine the vehicle routing problem with hard time windows under demand and travel time uncertainty. To address the problem, we build a robust optimization model based on novel route-dependent uncertainty sets. However, due to the complex nature of the problem, the robust model is only able to tackle small-sized instances using standard solvers. Therefore, to tackle large instances, we design a two-stage algorithm based on a modified adaptive variable neighborhood search heuristic. The first stage of the algorithm minimizes the total number of vehicle routes, while the second stage minimizes the total travel distance. Extensive computational experiments are conducted with modified versions of Solomon's benchmark instances. The numerical results show that the proposed two-stage algorithm is able to find optimal solutions for small-sized instances and good-quality robust solutions for large-sized instances with little increase to the total travel distance and/or the number of vehicles used. A detailed analysis of the results also reveals several managerial insights for decision-makers in the logistics industry.

*Keywords:* Vehicle routing, time windows, uncertainty, robust optimization, adaptive variable neighborhood search

*Corresponding author at: Faculty of Engineering and Information Technology, University of Technology Sydney, Broadway, NSW 2007, Australia. E-mail address: huchenlian@sjtu.edu.cn, chenlian.hu@student.uts.edu.au (C.Hu)

# 1 Introduction

The vehicle routing problem (VRP) is a combinatorial optimization problem, first introduced by Dantzig and Ramser (1959), that aims to find the optimal set of routes for a fleet of vehicles delivering goods or services to a given set of customers. Due to its broad application in a variety of practical contexts (e.g., logistics, internet routing, and freight transportation), the VRP and its numerous variants have been studied extensively (Baldacci et al., 2012; Kumar and Panneerselvam, 2012; Toth and Vigo, 2014).

The vehicle routing problem with time windows (VRPTW), as an important variant of the VRP, assumes that each customer must be served within a given time window. For example, some customers might only accept deliveries between 11:00 am and 1:00 pm. Due to increasing and more exacting customer requirements, incorporating customer time windows into today's logistic and freight transportation activities has become more and more popular over the last two decades, and the VRPTW and its variants are likely to continue to be a hot research topic among researchers for the foreseeable future (Bräysy and Gendreau, 2005a,b; Baldacci et al., 2012; Miranda and Conceição, 2016). Several popular real-world applications of the VRPTW include waste collection, postal deliveries, school bus routing, and security patrol services (Bräysy and Gendreau, 2005a). The VRPTW assumes that all the input data, such as customer demands and travel times, are both deterministic and are known in advance. However, the solutions derived by deterministic models are often infeasible when applied to real-world situations because the level of customer demand and the travel times are uncertain (Taş et al., 2013). For example, in school bus routing problems, a single bus collects students from several pre-assigned stops and must arrive at the school within a specified time window. However, not all students use the bus on a daily basis and the likelihood that a student will take the bus on any given day varies significantly in practice (Caceres et al., 2017). Thus, the number of students at each stop is highly uncertain. Moreover, the levels of traffic congestion in urban areas means the travel time between any two stops can vary greatly. Hence, ignoring the uncertainty of travel times or student demand may mean students are late for school or that the bus does not have room for all the students. Real-world waste collection problems are also plagued by these two types of uncertainty. The amount of waste at each collection point can vary daily and vehicle speeds are influenced by the prevailing traffic conditions. Indeed, uncertainty in demand and travel times are common issues in many real-world VRPTW applications. And when vehicles fail to serve customers within an agreed upon time window, a reputation for poor service and customer dissatisfaction is often the result.

In an attempt to find an efficient routing strategy for such real-world applications, we study the VRPTW with demand and travel time uncertainty. The robust optimization theory outlined by Ben-Tal et al. (2009) is used to deal with the uncertainties. Robust optimization treats the uncertain parameters of an optimization problem as random variables. Early studies on robust optimization focused on deriving a robust counterpart for the original optimization problem using predefined uncertainty sets, such as ellipsoids or polyhedrons, to obtain a feasible least-cost solution for any realization of the uncertain parameters over the uncertainty sets (Ben-Tal and Nemirovski, 2002; Bertsimas and Sim, 2003, 2004). However, more recent works on robust optimization have developed several solution frameworks that can exploit the additional distribution information available in the uncertain parameters to produce less conservative solutions (Bertsimas et al., 2011). One advantage of robust optimization is that the consistency between the tractability of the robust counterpart and the original problem can be maintained as long as the uncertainty sets are designed properly, e.g., by using polyhedral uncertainty sets (Bertsimas and Sim, 2004). Additionally, the uncertainty sets can be derived with only partial information about the distributions of the uncertain parameters, such as supports and moments (Han et al., 2013). Thus, robust optimization is advantageous because of its computational tractability and its ability to tackle real-world applications where only partial distribution information or a small amount of historical data about the uncertain parameters is available.

Our aim through this research is to explore the joint impact of demand and travel time uncertainty on route planning, and produce a cost-effective, robust *a priori* routing strategy that is insensitive to uncertainty. The resulting contributions of this paper are as follows:

(1) a robust optimization model based on novel route-dependent uncertainty sets for the VRPTW with demand and travel time uncertainty;

(2) a two-stage algorithm based on a modified adaptive variable neighborhood search (AVNS) that is able to deal with large-sized instances; and

(3) an extensive computational study using adapted benchmark instances that shows our two-stage algorithm can generate optimal solutions for small-sized instances and good-quality robust solutions for large-sized instances with little increase to the total travel distance and/or the number of vehicles used.

The remainder of this paper is structured as follows. The relevant literature is reviewed in Section 2. Section 3 describes the VRPTW and a robust version of the VRPTW with demand and travel time uncertainty. The corresponding mathematical models for these two problems are also provided. In Section 4, we describe the details of the proposed two-stage algorithm. Section 5 presents an extensive computational study using the modified versions of Solomon's benchmark instances (Solomon, 1987) and a comprehensive analysis of the results. Section 6 concludes the paper and offers several managerial insights.

## 2 Related work

The VRPTW is one of several well-known extensions to the VRP. The literature related to this study includes the VRPTW and its two important variants: stochastic versions of the VRPTW and robust versions of the VRPTW. For a comprehensive background on the VRP, we refer readers to Toth and Vigo (2014).

In the VRPTW, each customer must be served by one vehicle within a scheduled time window. Customer time windows can be "hard", where earliness or tardiness are not allowed, or "soft", where earliness and tardiness are allowed but with a penalty (Miranda and Conceição, 2016). Over the last two decades, scholars have intensively proposed methods of solving the VRPTW with both exact and heuristic algorithms. The most successful exact algorithms rely on a column generation technique to compute the lower bound and a branch-and-price or a branch-and-cut-and-price algorithm to find an optimal integer solution (Jepsen et al., 2008; Desaulniers et al., 2008; Baldacci et al., 2011). Some of the more well-known heuristics include Tabu-search (TS) (Cordeau et al., 2001), large neighborhood search (LNS) (Pisinger and Ropke, 2007), variable neighborhood search (VNS) (Mladenović and Hansen, 1997), iterated local search (Ibaraki et al., 2005) and evolutionary algorithms (Mester and Bräysy, 2005). Detailed surveys of the various methods for solving the VRPTW are provided in Bräysy and Gendreau (2005a,b). As extensions to the VRPTW, a number of more complex routing problems have been explored. One interesting extension is the VRPTW with multiple depots and heterogeneous vehicles. In this scenario, the vehicles have different capacities and multiple vehicle depots are available (Dondo and Cerdá, 2007). The pickup and delivery problem with time windows is another interesting extension. Here, a number of customer requests, that each involves shipping some goods from a pickup point to a delivery point, need to be satisfied by a fleet of vehicles. Two successful heuristics for solving this problem are the reactive TS heuristic (Nanry and Barnes, 2000) and the adaptive LNS (ALNS) heuristic (Ropke and Pisinger, 2006).

More recently, researchers have begun to associate issues with data uncertainty and the VRPTW, giving rise to stochastic versions of the VRPTW (Ritzinger et al., 2016; Gendreau et al., 2016) and robust versions of the VRPTW (Agra et al., 2012, 2013; Braaten et al., 2017). Reviews of these two important extensions follow.

Stochastic versions of the VRPTW extend the VRPTW by assuming that the key problem parameters, such as customer demands, travel times, and service times, are stochastic. Over the last 20 years, many researchers have studied different stochastic versions of the VRPTW with the majority focusing on stochastic demands and stochastic travel times. Of the studies that focus on the VRPTW with stochastic demands, several different mathematical models and algorithms have been proposed and explored: Ong et al. (1997) proposed a chance-constrained model; Chang (2005) proposed a two-stage stochastic programming with recourse model and developed a heuristic based on the integer L-shaped method; Lei et al. (2011) also formulated a two-stage stochastic recourse model and devised an ALNS heuristic; and Zhang et al. (2016) proposed three probabilistic models to address on-time deliveries from different perspectives of the carrier and customers. The VRPTW with stochastic travel times was introduced more recently and has attracted greater researcher attention than the VRPTW with stochastic demands ever since. Ando and Taniguchi (2006) used a genetic algorithm to solve the VRPTW with stochastic travel times. Russell and Urban (2008) addressed the VRPTW with stochastic travel times based on the assumption that the travel time between any two customers obeys an Erlang distribution. Li et al. (2010) considered the VRPTW with both stochastic travel times and service times and introduced a chance-constrained model and a stochastic recourse model. Taş et al. (2013) addressed the VRP with soft time windows and stochastic travel times by assuming the travel times obey a gamma distribution. Taş et al. (2014) also studied the VRP with soft time windows and considered time-dependent and stochastic travel times. A TS heuristic or an ALNS heuristic is used to solve the problem depending on whether or not service times are a factor. More recently, Ehmke et al. (2015) solved the VRPTW with stochastic travel times by a chance-constraint approach. Recent surveys of stochastic versions of the VRPTW can be found in Ritzinger et al. (2016) and Gendreau et al. (2016).

Our literature review of stochastic versions of the VRPTW reveals that researchers often solve these problems using a two-stage stochastic programming with recourse model or a chance-constrained programming model. However, both these methods have two distinct disadvantages. First, both methods assume that the precise distributions of the uncertain parameters are known or can be estimated in advance, which is rare in real-world VRPTW applications. Second, since both methods suffer from the curse of dimensionality, their computational tractability is seriously affected by the dimensions of the problem at hand. As a result, both stochastic programming and chance-constrained programming are suitable for small- and medium-sized VRPTW applications where there is enough historical data to fairly accurately estimate the distributions of the uncertain parameters.

Consequently, a few researchers have turned to robust optimization as a new framework for overcoming these disadvantages. Robust optimization treats uncertain parameters as random variables that take their values from predefined uncertainty sets. As such, this framework has great merit in terms of its computational tractability and its ability to solve real-world VRPTW applications with only partial distribution information or little historical data about the uncertain parameters (Gounaris et al., 2014). For a detailed introduction to robust optimization, we refer readers to Ben-Tal et al. (2009).

Unlike stochastic versions of the VRPTW, robust versions of the VRPTW, and even robust versions of the VRP that do not consider time windows, have received much less research attention. According to our review, Sungur et al. (2008) were the first to discuss a robust version of the VRP with demand uncertainty. Gounaris et al. (2013) also considered a robust version of the VRP under demand uncertainty. They developed robust counterparts of several known deterministic VRP formulations and proposed a branch-and-cut solution procedure that is able to optimally solve instances of up to 50 nodes (i.e., customers). Gounaris et al. (2014) presented an adaptive memory programming meta-heuristic that can provide high-quality solutions for the same problem discussed in Gounaris et al. (2013). Researchers have also used robust optimization to deal with some variants of the VRP that consider other types of uncertainty. Ordóñez (2010) showed that the Miller-Tucker-Zemlin model could be modified to handle various robust versions of the VRP that include uncertain travel times, travel costs, and customer demands. Lee et al. (2012) tackled a robust version of the VRP with deadlines under demand and travel time uncertainty using an exact solution algorithm based on a column generation method. However, their algorithm is limited to solving instances of up to 25 nodes and takes around an hour to provide a routing strategy. Solano-Charris et al. (2015) considered a robust version of the VRP with uncertain travel costs, relying on two greedy heuristics, a local search, and four meta-heuristics to solve the problem. Adulyasak and Jaillet (2015) considered the VRP with deadlines given uncertain travel times in the context of both stochastic and robust optimization. They show that a robust model produces very competitive solutions compared to a stochastic model in a great number of scenarios, and is much less sensitive to uncertainty. Despite some interest in robust versions of the VRP, very few researchers have investigated robust versions of the VRPTW. Agra et al. (2012) proposed a layered formulation for a robust version of the VRPTW with travel time uncertainty. In a continuing work, Agra et al. (2013) proposed two new formulations for the same problem and developed a new column generation algorithm based on two techniques that can significantly reduce the extreme points of an uncertainty set. However, their exact algorithm is only able to solve instances of up to 40 nodes. Jaillet et al. (2016) proposed a novel mathematical framework based on distributionally robust optimization to address a specific class of routing problems with uncertainty. To illustrate the efficacy of their framework, they solved the VRP with soft time windows given uncertain travel times. Heuristic methods are a more recent introduction to solving robust versions of the VRPTW.

Toklu et al. (2014) proposed an ant colony optimization heuristic to solve a robust version of the VRPTW under travel time uncertainty. Braaten et al. (2017) developed an ALNS heuristic to solve the same problem discussed in Agra et al. (2013).

Our review of the relevant literature only reveals a limited number of papers that deal with robust versions of the VRPTW. Most only consider one kind of uncertainty, either travel time or customer demand; few consider multiple kinds of uncertainty or explore the joint impact of multiple uncertain parameters on route planning. Moreover, the existing exact algorithms can only tackle robust versions of the VRPTW with small-to-medium-sized instances. Efficient algorithms that are able to solve real-world scenarios with large-sized instances are still needed. Thus, in this paper, we consider a robust version of the VRPTW with both demand and travel time uncertainty and develop a two-stage algorithm that can solve the problem with large-sized instances.

# 3 Problem description and model formulation

Subsection 3.1 briefly describes the VRPTW and presents a deterministic mathematical formulation for the problem. In Subsection 3.2, we consider a robust version of the VRPTW under demand and travel time uncertainty, and extend the deterministic model into a robust formulation based on novel route-dependent uncertainty sets.

## 3.1 The deterministic VRPTW

The VRPTW is defined on a complete digraph $G = (N, A)$, where the set of nodes is represented by $N = \{0, 1, \cdots, n, n+1\}$, and the set of arcs is represented by $A = \{(i, j)|i, j \in N, i \neq j\}$. $N_c = \{1, \cdots, n\}$ denotes the set of customers and $K = \{1, \cdots, m\}$ denotes the set of homogeneous vehicles with a capacity of $Q$. The vehicle depot is represented by two nodes 0 and $n+1$. Every vehicle starts from the depot node 0, visits a subset of customer nodes, and ends its route at depot node $n+1$. Each customer node $i \in N_c$ has a non-negative demand $q_i$, a service time $s_i$, and a time window $[a_i, b_i]$. $a_i$ and $b_i$ denote the earliest and latest allowable arrival time at customer node $i$, respectively. If a vehicle arrives at customer node $i$ before $a_i$, it must wait until $a_i$, and the time window is missed if it arrives at node $i$ after $b_i$. A time window is also associated with the depot nodes 0 and $n + 1$, i.e., $[a_0, b_0] = [a_{n+1}, b_{n+1}]$. The travel time between node $i$ and node $j$ is denoted as $t_{ij}$ and, similarly, the travel distance between these nodes is denoted as $d_{ij}$, where $(i, j) \in A$. Each customer can only be served by exactly one vehicle and the vehicle fleet is big enough to serve all customers. The primary objective of the problem is to minimize the number of vehicle routes; the secondary objective is to minimize the total travel distance.

The mathematical formulation for the VRPTW contains two types of variables. The first are flow variables $x_{ijk}$ for all $(i,j) \in A$ and $k \in K$. Let $x_{ijk}$ be 1 if arc $(i,j)$ is used by vehicle $k$, and 0 otherwise. The second type of variables are time variables $y_i$ for all $i \in N$. $y_i$ specifies the arrival time of a vehicle at node $i$. The (deterministic) VRPTW is formulated as follows:

$$
\begin{aligned}
(VRPTW)\ \min \quad & \left( \sum_{k\in K}\sum_{j\in N_c} x_{0jk}, \sum_{k\in K}\sum_{(i,j)\in A} x_{ijk} d_{ij} \right) && && (1)\\
\text{s.t.} \quad & \sum_{k\in K}\sum_{j\in N} x_{ijk} = 1 && \forall i \in N_c && (2)\\
& \sum_{j\in N_c} x_{0jk} = 1 && \forall k \in K && (3)\\
& \sum_{i\in N_c} x_{i,n+1,k} = 1 && \forall k \in K && (4)\\
& \sum_{j\in N} x_{ijk} - \sum_{j\in N} x_{jik} = 0 && \forall k \in K, i \in N && (5)\\
& \sum_{i\in N} q_i \sum_{j\in N} x_{ijk} \leq Q && \forall k \in K && (6)\\
& y_i - y_j + (b_i + t_{ij} + s_i - a_j) x_{ijk} && && \\
& \quad \leq b_i - a_j \qquad \forall k \in K, (i,j) \in A && && (7)\\
& a_i \leq y_i \leq b_i && \forall k \in K, i \in N && (8)\\
& x_{ijk} \in \{0,1\} && \forall k \in K, (i,j) \in A && (9)
\end{aligned}
$$

Objective (1) lexicographically minimizes the number of vehicles (routes) and the total travel distance. Constraints (2) guarantee that each customer can only be visited by one vehicle. Constraints (3)-(5) are the flow conservation constraints, which ensure that each vehicle's route starts from depot 0 and ends at depot $n+1$. Constraints (6) are the vehicle capacity constraints. Constraints (7) calculate the arrival time of a vehicle at node $i\,(i \in N)$. Constraints (8) ensure that each time window is respected. Constraints (9) deal with the nature of the variables.

## 3.2 A robust VRPTW under demand and travel time uncertainty

As discussed in Section 1, real-world VRPTW applications are often subject to a variety of uncertainties. Therefore, a deterministic VRPTW model, which ignores the uncertainty in data, may not be an appropriate choice; a robust optimization model, which does consider uncertainty, may be more suitable. We use the robust optimization theory discussed in Ben-Tal et al. (2009) and consider a robust version of the VRPTW with both demand and travel time uncertainty. However, to effectively represent the uncertain parameters, robust optimization requires a careful and practical definition of the uncertainty sets. In real-world VRPTW applications, each vehicle route services a different subset of customers. Thus, each vehicle suffers different levels of uncertainty. For example, if a vehicle has a longer pre-planned route and/or more customers on its route, it is likely to experience higher uncertainty. Further, not every customer on a route will have the same level of demand uncertainty, and not every route segment will have the same level of travel time uncertainty. In reality, a route may only contain a few customers with high demand uncertainty and a few crowed route segments with high travel time uncertainty. Lee et al. (2012) made similar observations and, hence, restricted the number of customers or segments with high uncertainty on each route to control the overall robustness of the routing strategy.

Based on the above observations and discussions, we define two types of uncertainty sets for each vehicle used $k$: the customer demand uncertainty set $U_q^k$ and the travel time uncertainty set $U_t^k$. Each is defined as a budget uncertainty polytope, as discussed in Bertsimas and Sim (2003, 2004). The following polytopes formally define the overall demand uncertainty set $\mathcal{U}_q$ and travel time uncertainty set $\mathcal{U}_t$ for the considered robust version of the VRPTW.

$$\mathcal{U}_q = \times_{k \in K} U_q^k \tag{10}$$

with

$$U_q^k = \Big\{ \widetilde{q} \in R^{|N^k|} | \widetilde{q}_i = \overline{q}_i + \alpha_i \hat{q}_i, \sum_{i \in N^k} |\alpha_i| \leq \Gamma_q^k, \\ |\alpha_i| \leq 1, \Gamma_q^k = \left\lceil \theta_q |N^k| \right\rceil, \forall i \in N^k \Big\} \tag{11}$$

$$\mathcal{U}_t = \times_{k \in K} U_t^k \tag{12}$$

with

$$U_t^k = \Big\{ \widetilde{t} \in R^{|A^k|} | \widetilde{t}_{ij} = \overline{t}_{ij} + \beta_{ij} \hat{t}_{ij}, \sum_{(i,j) \in A^k} |\beta_{ij}| \leq \Gamma_t^k, \\ |\beta_{ij}| \leq 1, \Gamma_t^k = \left\lceil \theta_t |A^k| \right\rceil, \forall (i,j) \in A^k \Big\} \tag{13}$$

Equation (10) reflects that the overall demand uncertainty set $\mathcal{U}_q$ is the Cartesian product of the demand uncertainty set $U_q^k$ for each vehicle. In equation (11), $N^k$ denotes the set of customers on the route of vehicle $k$; $\overline{q}_i$ represents the nominal value of uncertain demand $\widetilde{q}_i$; and $\hat{q}_i$ denotes the maximum deviation from the nominal value for each node $i \in N^k$. $\alpha_i$ is the auxiliary variable, and $\Gamma_q^k$ is the uncertainty budget that controls the level of demand uncertainty. Given the level of demand uncertainty experienced by each vehicle relates to the number of customers on its route, $\Gamma_q^k$ is defined as equaling $\lceil \theta_q |N^k| \rceil$. $\lceil \theta_q |N^k| \rceil$ is the least integer that is greater than or equal to $\theta_q |N^k|$. $\theta_q$ is the demand uncertainty budget coefficient, and has a value of between 0 and 1. If $\theta_q = 0$, $\Gamma_q^k = 0$ and $\widetilde{q}_i = \overline{q}_i$, there is no demand uncertainty. If $\theta_q = 1$, $\Gamma_q^k = |N^k|$, each customer demand $\widetilde{q}_i$ can take any value in the interval $[\overline{q}_i - \hat{q}_i, \overline{q}_i + \hat{q}_i]$. Broadly speaking, $\Gamma_q^k$ imposes an upper bound on the number of customers with high demand uncertainty and is determined by $\theta_q$ and $|N^k|$. Equation (12) shows that the overall travel uncertainty set $\mathcal{U}_t$ is the Cartesian product of the travel time uncertainty set $U_t^k$ for each vehicle. The parameters in equation (13) have similar meanings to those in equation (11). $A^k$ denotes the set of arcs on the route of vehicle $k$. $\overline{t}_{ij}$ denotes the nominal value of uncertain travel time $\widetilde{t}_{ij}$, and $\hat{t}_{ij}$ represents the maximum deviation from the nominal value for each arc $(i, j) \in A^k$. $\beta_{ij}$ is the auxiliary variable. $\theta_t$ is the travel time uncertainty budget coefficient, and has a value of between 0 and 1. The travel time uncertainty budget $\Gamma_t^k$ controls the degree of travel time uncertainty and is defined as equaling $\lceil \theta_t |A^k| \rceil$. $|A^k|$ denotes the number of arcs on the route of vehicle $k$.

Based on the above defined route-dependent uncertainty sets, we extend the deterministic model outlined in equations (1)-(9) into a robust model by adapting the resource inequalities formulation from Agra et al. (2013). Agra et al. (2013) proposed the resource inequalities formulation for a robust version of the VRPTW with travel time uncertainty based on adjustable robust optimization discussed in Ben-Tal et al. (2009). In adjustable robust optimization, some decision variables are allowed to adapt themselves as uncertain parameters vary in uncertainty sets. As such, we change each decision variable $y_i$ in the deterministic model to $y_i(q, t)$ for all $i \in N$, where $y_i(q, t)$ is a function of $q \in \mathcal{U}_q$ and $t \in \mathcal{U}_t$. According to Agra et al. (2013), only the demand vector $q \in ext(\mathcal{U}_q)$ and travel time vector $t \in ext(\mathcal{U}_t)$ need to be considered in the robust formulation. $ext(\mathcal{U}_q)$ and $ext(\mathcal{U}_t)$ are sets that contain all the extreme points of sets $\mathcal{U}_q$ and $\mathcal{U}_t$, respectively. Thus, the considered robust version of the VRPTW given the route-dependent uncertainty sets can be formulated as follows:

$$(RVRPTW) \min \left( \sum_{k \in K} \sum_{j \in N_c} x_{0jk}, \sum_{k \in K} \sum_{(i,j) \in A} x_{ijk} d_{ij} \right) \tag{14}$$

$$
\begin{aligned}
\text{s.t.}\quad & (2), (3), (4), (5) \\
& y_i(q,t) - y_j(q,t) + (b_i + t_{ij} + s_i - a_j)x_{ijk} \le b_i - a_j \\
& \qquad\qquad\qquad\qquad \forall k \in K, (i,j) \in A, q \in ext(\mathcal{U}_q), t \in ext(\mathcal{U}_t) && (15) \\
& a_i \le y_i(q,t) \le b_i \qquad \forall k \in K, i \in N, q \in ext(\mathcal{U}_q), t \in ext(\mathcal{U}_t) && (16) \\
& x_{ijk} \in \{0,1\} \qquad \forall k \in K, (i,j) \in A && (17)
\end{aligned}
$$

Constraints (7) and (8) are rewritten for every $q \in ext(\mathcal{U}_q)$ and $t \in ext(\mathcal{U}_t)$ as shown in constraints (15) and (16). One key observation from the above formulation is that the number of decision variables $y_i(q,t)$ and the number of constraints (15) and (16) increase with the size of the sets $N$, $ext(\mathcal{U}_q)$, and $ext(\mathcal{U}_t)$. Notably, finding all the extreme points in the sets $\mathcal{U}_q$ and $\mathcal{U}_t$ is a non-trivial task. However, the column-and-row generation algorithm discussed in Agra et al. (2013) can be used to solve the above formulation with small-sized instances.

# 4 A two-stage algorithm for the robust VRPTW

The above adapted robust formulation always finds it difficult to handle large-sized VRPTW instances using standard solvers. Therefore, to generate high-quality solutions with large-sized instances, we devise a simple two-stage algorithm based on a modified AVNS heuristic. Subsection 4.1 provides an overview of the two-stage algorithm, followed by a detailed description of the modified AVNS heuristic in Subsection 4.2.

## 4.1 Overview of the two-stage algorithm

Given there are two lexicographical objectives in the considered robust version of the VRPTW, the proposed two-stage algorithm separates the optimization process into two stages. The first stage minimizes the total number of routes and, therefore, the number of vehicles used. The second stage minimizes the total travel distance in the overall routing strategy. Both stages rely on a modified AVNS heuristic for optimization. An overview of the two-stage algorithm is shown in Figure 1.

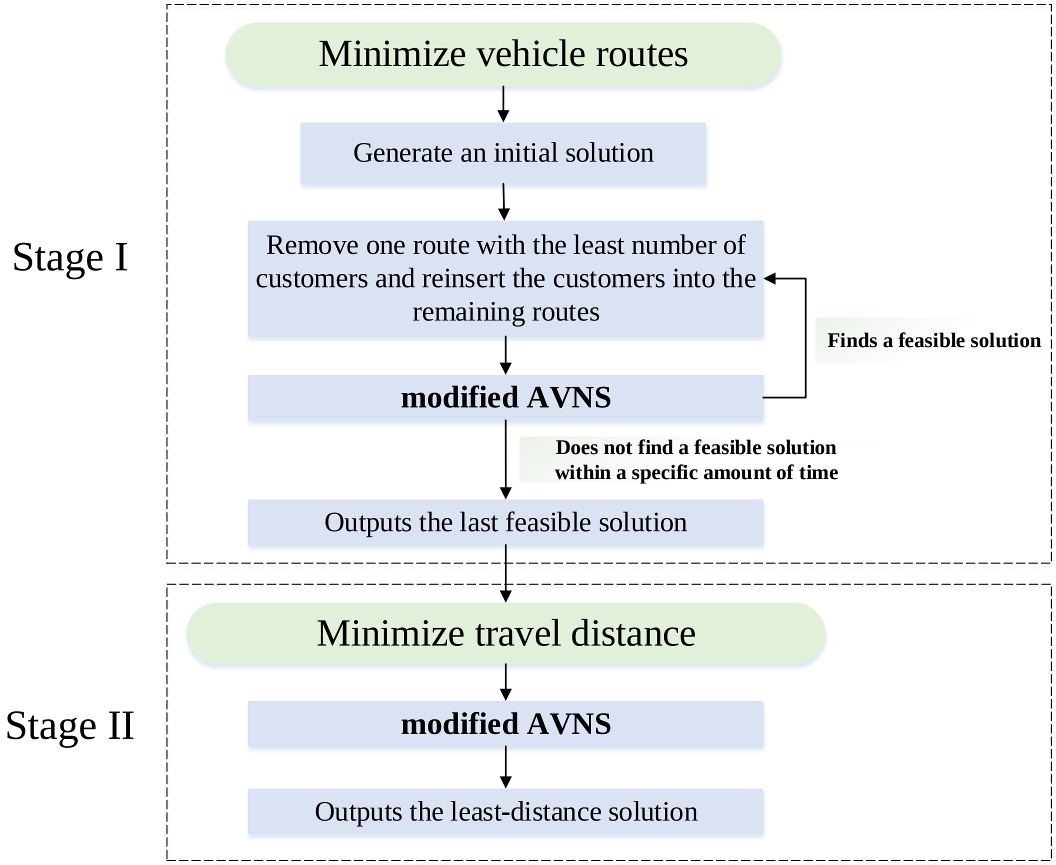

Stage I
Minimize vehicle routes
Generate an initial solution
Remove one route with the least number of customers and reinsert the customers into the remaining routes
Finds a feasible solution
modified AVNS
Does not find a feasible solution within a specific amount of time
Outputs the last feasible solution
Stage II
Minimize travel distance
modified AVNS
Outputs the least-distance solution

Figure 1: The two-stage algorithm for the robust VRPTW

As shown in Stage I of Figure 1, an initial feasible solution is generated using a sequential best insertion heuristic and the maximal possible values of the uncertain parameters from the predefined route-dependent uncertainty sets. In the next step, the route with the least number of customers is removed from the initial solution, and the customers on the removed route are randomly reinserted into the remaining routes. A modified AVNS heuristic then optimizes the resulting solution. Note that the heuristic allows for infeasible solutions during this search process. If a feasible solution is found, the solution becomes the new candidate for the minimum number of vehicle routes and the heuristic is immediately stopped. Next, the vehicle route with the least number of customers is removed from the new candidate, the resulting solution is optimized by the heuristic, and the process reiterates. If no feasible solutions are found within a specified time by the modified AVNS heuristic, the last feasible solution is output as the result for Stage I. The output solution from Stage I is used as the initial solution for Stage II. In Stage II, the modified heuristic simply minimizes the overall total travel distance for all vehicle routes in the final solution from the first stage.

## 4.2 A modified AVNS heuristic

The AVNS heuristic was first proposed by Stenger et al. (2013). It incorporates an adaptive mechanism that biases the random shaking step in the basic VNS algorithm (Mladenović and Hansen, 1997). The AVNS heuristic has two major advantages. First, several selected routes, and the sequence of nodes in those routes, can be modified during the adaptive shaking step in one iteration, which produces strong diversification possibilities to escape the local minimum. Second, the algorithm can adapt based on recent search performance, so running times with large-sized instances are reasonable. As discussed in Subsection 4.1, we use a modified AVNS heuristic in both stages of the two-stage algorithm. The modified AVNS heuristic considers the characteristics of the studied robust version of the VRPTW. The structure of the modified AVNS heuristic is shown in Algorithm 1 in pseudo-code.

**Algorithm 1** The structure of the modified AVNS heuristic

```
Define adaptive shaking neighborhood structures $\mathcal{N}_{\mathcal{I}}^{shake}$ with $\mathcal{I} = 1, \cdots, \mathcal{I}_{max}$ and the
  local search neighborhood structures $\mathcal{N}_{\mathcal{J}}^{search}$ with $\mathcal{J} = 1, \cdots, \mathcal{J}_{max}$
Generate an initial solution $R^{initial}$ and improve $R^{initial}$ by local search
Set current solution $R^{current} \leftarrow R^{initial}$ and $\mathcal{I} \leftarrow 1$
repeat
    {Adaptive shaking}
    Determine route and node selection methods based on the adaptive mechanism
    Generate a shaking solution $R^{shake} \in \mathcal{N}_{\mathcal{I}}^{shake}(R^{current})$
    {Local search}
    Perform local search on $R^{shake}$ with neighborhoods $\mathcal{N}_{\mathcal{J}}^{search}, \mathcal{J} = 1, \cdots, \mathcal{J}_{max}$ to find
      local optimal solution $R^{search}$
    {Acceptance decision}
    if ($R^{search}$ improves $R^{current}$) or ($R^{search}$ accepted) then
        $R^{current} \leftarrow R^{search}$ and $\mathcal{I} \leftarrow 1$
    else
        if $\mathcal{I} < \mathcal{I}_{max}$ then
            $\mathcal{I} \leftarrow \mathcal{I} + 1$
        else
            $\mathcal{I} \leftarrow 1$
        end if
    end if
    Update the value of parameters in the adaptive selection mechanism
until a given time limit $T$ or a given number $Iter$ of iterations without improvement
```

The main components of the modified AVNS heuristic consist of an adaptive shaking step and a local search step that are repeated until a given time limit $T$ is met or a given number of main iterations $Iter$ without improvement has been reached. Initially, two sets of neighborhood structures $\{\mathcal{N}^{shake}_{\mathcal{I}} | \mathcal{I} = 1, \cdots, \mathcal{I}_{max}\}$ and $\{\mathcal{N}^{search}_{\mathcal{J}} | \mathcal{J} = 1, \cdots, \mathcal{J}_{max}\}$ are defined. The initial solution $R^{initial}$ is generated, then improved through a local search. In the adaptive shaking step, a shaking solution $R^{shake}$ is generated in the $\mathcal{I}$-th neighborhood $\mathcal{N}^{shake}_{\mathcal{I}}$ given the current solution $R^{current}$. In the local search step, a local optimum solution $R^{search}$ is determined by a local search method with neighborhoods $\mathcal{N}^{search}_{\mathcal{J}}, \mathcal{J} = 1, \cdots, \mathcal{J}_{max}$. Specifically, a randomized variable neighborhood descent (RVND) heuristic is used in the local search step. If $R^{search}$ is accepted, it replaces the current solution $R^{current}$ and $\mathcal{I}$ is reset to one. Otherwise, $R^{search}$ is discarded and $\mathcal{I}$ is increased by one. Since infeasible solutions are allowed in the search process, the next subsection explains the feasibility check and how the value of the objective function is calculated for a given solution, followed by a detailed description of each of the major steps of the modified AVNS heuristic in subsequent subsections.

### 4.2.1 Feasibility check and objective calculation

Infeasible solutions are allowed in the search process because, in tightly constrained problems, the local search process often quickly converges to local optimal solutions (e.g., the considered robust version of the VRPTW), but penalties are added to the objective function for violated constraints. However, allowing for infeasible solutions creates the challenge of checking route feasibility and calculating the objective function value of a given solution. As mentioned in Subsection 3.2, uncertain travel times and customer demands take their values from the route-dependent uncertainty sets. Thus, the largest possible vehicle load and the latest vehicle arrival time at each customer node on a route can be used to check a route's feasibility. The intuition behind this notion is that if the largest possible vehicle load on a given route is greater than the vehicle capacity or the latest possible vehicle arrival time for any customer node falls behind the schedule time window, the route is infeasible.

To calculate the largest possible vehicle load and the latest possible vehicle arrival time, consider a given solution (routing strategy) $R$ that consists of $m$ routes and $R = \{r_1, \cdots, r_m\}$. Let $r_k$ denote a route served by vehicle $k \in K$, and $r_k \in R$. The total number of nodes on route $r_k$ is $|r_k|$ and the $i$th node on route $r_k$ is denoted as $p^k_i$. Thus, route $r_k$ can be presented as an ordered sequence of nodes, and $r_k = \{p^k_1, p^k_2, \cdots, p^k_{|r_k|-1}, p^k_{|r_k|}\}$. Note that $p^k_1$ is node 0 and $p^k_{|r_k|}$ is node $n+1$, which represent the starting and ending depots, respectively. $\{p^k_2, \cdots, p^k_{|r_k|-1}\}$ is included in the customer set $N_c$. As discussed in Subsection 3.2, $\Gamma^k_q$ and $\Gamma^k_t$ control the number of nodes and segments on route $r_k$ that have high levels of demand and travel time uncertainty. Thus, the latest possible vehicle arrival time at the $i$th node on route $r_k$ is denoted as $\Lambda^k(i, \Gamma^k_t)$, and the vehicle's largest possible load is denoted as $\Theta^k(\Gamma^k_q)$.

$\Lambda^k(i,\Gamma_t^k)$ is calculated using a function originally proposed by Agra et al. (2013), formulated as follows:

$$\Lambda^k(i,\Gamma_t^k) = \begin{cases} 0 & \text{if} \quad i=1 \\ max\left(a_{p_i^k}, \Lambda^k(i-1,\Gamma_t^k) + s_{p_{i-1}^k} + \bar{t}_{p_{i-1}^k,p_i^k}\right) & \text{if} \quad 2 \le i \le |r_k| \text{ and } \Gamma_t^k = 0 \\ max\Big(a_{p_i^k},\ \Lambda^k(i-1,\Gamma_t^k-1) + s_{p_{i-1}^k} + \bar{t}_{p_{i-1}^k,p_i^k} & \\ \qquad +\hat{t}_{p_{i-1}^k,p_i^k},\ \Lambda^k(i-1,\Gamma_t^k) + s_{p_{i-1}^k} + \bar{t}_{p_{i-1}^k,p_i^k}\Big) & \text{if} \quad i \le |r_k| \text{ and } 1 \le \Gamma_t^k \le i-1 \\ \Lambda^k(i,\Gamma_t^k-1) & \text{if} \quad 2 \le i \le \Gamma_t^k \end{cases} \tag{18}$$

Equation (18) shows that the latest possible vehicle arrival time at each customer node on route $r_k$ is a recursive function of the sequence of the nodes on the route and the parameter $\Gamma_t^k$ defined in the uncertainty set $U_t^k$. In addition to $\Lambda^k(i,\Gamma_t^k)$, we can also calculate the largest possible vehicle load $\Theta^k(\Gamma_q^k)$ on route $r_k$ by arranging the sequence of nodes in route $r_k$ in decreasing order of customer demand and generating a new sequence $\{o_1^k, o_2^k, o_3^k, \cdots, o_{|r_k|-1}^k o_{|r_k|}^k\}$. Note that $o_i^k$ represents the node with the $i$th largest customer demand on route $r_k$. Based on this sequence, the largest possible vehicle load $\Theta^k(\Gamma_q^k)$ on route $r_k$ can be calculated by equation (19).

$$\Theta^k(\Gamma_q^k) = \sum_{i=1}^{|r_k|} \bar{q}_{o_i^k} + \sum_{j=1}^{\Gamma_q^k} \hat{q}_{o_j^k} \tag{19}$$

The objective function value of a given solution $R$ can be calculated as follows:

$$f(R) = Cost(R) + \delta_t \sum_{k=1}^{m} \sum_{i=1}^{|r_k|} max(0, \Lambda^k(i,\Gamma_t^k) - b_{p_i^k}) + \delta_d \sum_{k=1}^{m} max(0, (\Theta^k(\Gamma_q^k) - Q)) \tag{20}$$

where $Cost(R)$ represents the total travel distance in solution $R$, $\delta_t$ denotes the penalty factor for a time window violation, and $\delta_d$ denotes the penalty factor for a vehicle capacity violation in equation (20).

### 4.2.2 Initial solution

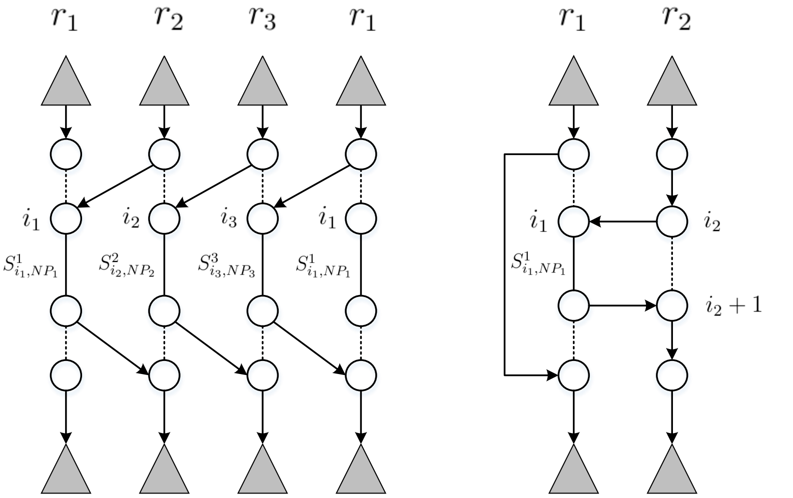
As outlined in Subsection 4.1, the modified AVNS heuristic is used in both stages of the two-stage algorithm. The initial solution for the modified AVNS heuristic is constructed using a sequential best insertion heuristic in the first stage of the algorithm. In the second stage of the algorithm, the final output from the first stage is used as the initial solution for the modified AVNS heuristic.

### 4.2.3 Adaptive shaking

In the shaking step of the modified AVNS heuristic, several selection methods are employed to initially determine the routes and node sequences of the current solution $R^{current}$ to be involved in the shaking. Then, the shaking solution $R^{shake}$ is generated based on the predefined shaking neighborhood operators. An adaptive mechanism is also incorporated into this step to improve the convergence rate of the heuristic toward good-quality solutions.

**Shaking neighborhoods**

The choice of shaking neighborhood structures is critical since the shaking step helps the AVNS explore a larger search space. In the modified AVNS, the shaking neighborhoods are generated based on two operators: a cyclic exchange and a sequence reinsert. Note that the sequence reinsert operator can be seen as a special case of the cyclic exchange operator. The cyclic exchange operator, originally introduced by Thompson and Psaraftis (1993), moves nodes among routes in a cyclic way. There are two important parameters for the cyclic exchange operator: the number of routes involved $NR$ and the maximum number of nodes to be exchanged $NP$. The cyclic exchange operator works as follows. For each route $r_k$, the operator moves the node sequence $S^k_{i,NP_k}$ starting with node $i$ at a length of $NP_k$ to route $r_{k+1}$ at the former position of sequence $S^{k+1}_{j,NP_{k+1}}$ (Ibaraki et al., 2005).



(a) Cyclic exchange operator (b) Sequence reinsert operator

Figure 2: Example of neighborhood operators

An example of a cyclic exchange with three routes is shown in Figure 2(a). Note that, if the total number of current routes is less than the number of routes to be moved, $NR$ is reduced accordingly. Similarly, $NP$ is adjusted if it is greater than the number of customer nodes on a given route. The sequence reinsert operator relocates a node sequence from one route to another. An example of the sequence reinsert operator with two routes is shown in Figure 2(b). Table 1 shows the neighborhood operators used in the shaking process. The modified AVNS considers neighborhood operators that exchange up to four routes simultaneously and move up to 10 nodes among them. Sequence lengths of five or fewer nodes are randomly chosen within the interval $[0, min(NP, |r_k|)]$. Sequence lengths of more than five nodes are assumed to be fixed.

**Table 1.** The shaking neighborhood structures

| N.O. | Type | $NR$ | $NP$ | N.O. | Type | $NR$ | $NP$ |
|---|---|---|---|---|---|---|---|
| 1 | sequence reinsert | 2 | 1 | 14 | cyclic exchange | 3 | 1 |
| 2 | sequence reinsert | 2 | 2 | 15 | cyclic exchange | 3 | 2 |
| 3 | sequence reinsert | 2 | 3 | 16 | cyclic exchange | 3 | 3 |
| 4 | sequence reinsert | 2 | 4 | 17 | cyclic exchange | 3 | 4 |
| 5 | sequence reinsert | 2 | 5 | 18 | cyclic exchange | 3 | 5 |
| 6 | cyclic exchange | 2 | 1 | 19 | cyclic exchange | 3 | 8 |
| 7 | cyclic exchange | 2 | 2 | 20 | cyclic exchange | 3 | 9 |
| 8 | cyclic exchange | 2 | 3 | 21 | cyclic exchange | 3 | 10 |
| 9 | cyclic exchange | 2 | 4 | 22 | cyclic exchange | 4 | 1 |
| 10 | cyclic exchange | 2 | 5 | 23 | cyclic exchange | 4 | 2 |
| 11 | cyclic exchange | 2 | 8 | 24 | cyclic exchange | 4 | 3 |
| 12 | cyclic exchange | 2 | 9 | 25 | cyclic exchange | 4 | 4 |
| 13 | cyclic exchange | 2 | 10 | 26 | cyclic exchange | 4 | 5 |

### Selection methods

To determine the routes and node sequences to be involved in the shaking, we perform a route selection process and a node sequence selection process. We develop some route and node sequence selection methods and apply several successful methods originating from Stenger et al. (2013) to make the adaptive shaking step more effective. The developed selection methods are based on a potential type of uncertainty associated with a vehicle route, e.g., the largest possible vehicle load or the latest vehicle arrival time at any node.

***Route selection***. Given the current shaking neighborhood $\mathcal{N}_{\mathcal{I}}^{shake}$, the route selection process needs to determine $NR$ routes that will be involved in the shaking. The first of the $NR$ routes is selected using one of the five route selection methods below.

1. *Random.* Each route has the same probability of being selected.

2. *Route length.* The probability of selecting a route is proportional to the route length. The idea is to redistribute nodes on long routes into shorter routes in the hope of reducing

the total traveling distance.

3. *Largest possible vehicle load on a route.* The probability of selecting a route is proportional to the largest possible vehicle load on a route. This is intended to reduce the number of routes at risk of exceeding vehicle capacity because the larger the vehicle load, the more likely it is that the route will violate the vehicle capacity constraint.

4. *Total customer slack time on a route.* The probability of selecting a route is inversely proportional to the total customer slack time on a route. The customer slack time is the difference between the upper bound of each customer's time window and the latest possible vehicle arrival time at its location. The goal is to reduce time-sensitive routes because the smaller the total customer slack time, the more likely it is that a vehicle will miss customer time windows on its route.

5. *Total number of time windows missed on a route.* The probability of selecting a route is proportional to the total number of time windows missed on a route. Since the shaking step may generate infeasible solutions, this is intended to convert infeasible routes into feasible routes. If all routes are feasible, one is chosen at random.

After selecting the first route, the remaining routes are chosen based on the following rule: the next route is randomly chosen from the three spatially closest routes to the previously selected route. The distance between any two routes is assumed to be the distance between the gravity points of these two routes.

***Node sequence selection.*** Once the routes involved in the shaking have been determined, the node sequence to be exchanged for each selected route is determined using one of the following four methods.

1. *Random.* Each node sequence is randomly chosen.

2. *Distance to the next route.* The probability of selecting a node sequence is inversely proportional to the distance between the gravity points of the node sequence and the route they will be inserted into.

3. *Customer slack time.* The probability of selecting a node sequence is inversely proportional to the corresponding total customer slack time of the sequence. This is intended to remove time-sensitive nodes from the route, which may increase the route's robustness.

4. *Missed time windows.* The probability of selecting a node sequence is proportional to the total number of missed time windows in that sequence. The intention is to remove nodes with violated time windows and reinsert them into other routes, which may result in an overall feasible solution. If the route is feasible, the node sequence is chosen at random.

**Adaptive mechanism**

At each shaking step, we use the roulette wheel selection procedure described in Pisinger and Ropke (2007) to determine the route and node selection methods. In the beginning, each route or node sequence selection method is assigned the same probability. Then, the probability of each method is dynamically updated after $\gamma$ main iterations based on a scoring system. The scoring system has the following rules. If a new overall best solution is achieved after applying a selection method, then a score of nine is added to the method. If the current solution is improved, a score of three is added. If the solution is worse than the current one, but accepted according to the acceptance criterion, a score of one is added to the method. Suppose the weights of $s$ selection methods are $w_i\,(i = 1, \cdots, s)$ and the current score of method $i$ is $\pi_i$. If the number of times the method was selected since the last weights update is $\phi_i$, the new weight $w_i$ can be computed by $w_i(1-\rho) + \rho(\pi_i/\phi_i)$ every $\gamma$ main AVNS iterations, where $\rho \in [0, 1]$. Note that $\pi_i$ and $\phi_i$ are reset to zero once the corresponding parameters are updated.

### 4.2.4 Local search

The solution generated within the shaking step $R^{shake}$ is then improved by a RVND heuristic. The RVND heuristic employs a set of neighborhood operators in random order and implements the best improvement strategy to prevent poor-quality local optimal solutions. Six neighborhood operators are used in the RVND: intra-route swap, intra-route reinsert, intra-route 2-opt, inter-route swap, inter-route reinsert, and inter-route 2-opt. A graphic illustration of the six neighborhood operators appears in Figure 3.

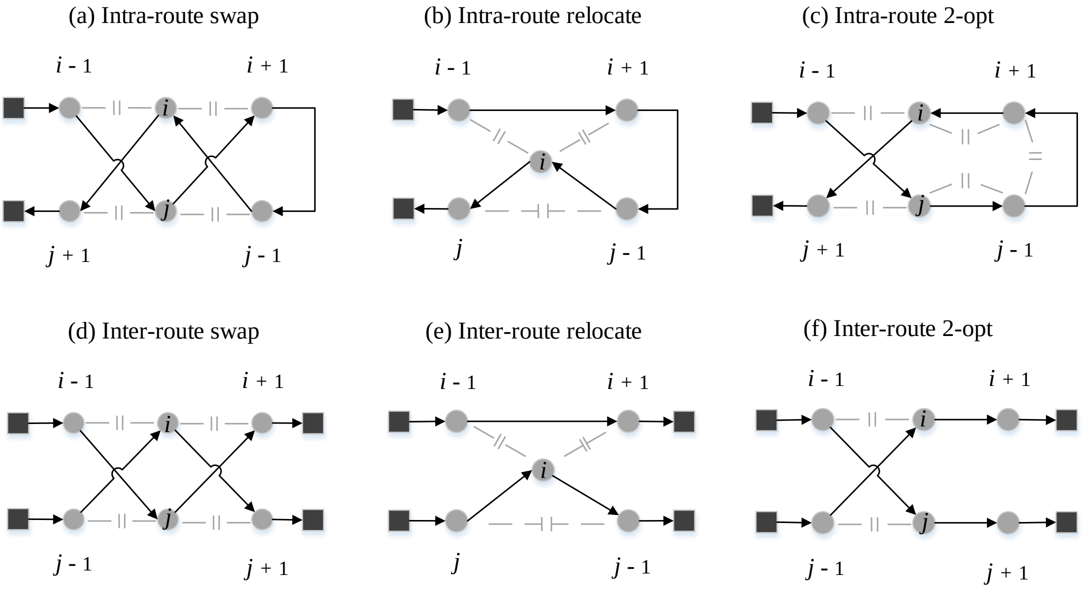


Figure 3: Six neighborhood operators

As shown in Figure 3, the intra-route swap operator exchanges a node $i$ with another node $j$ located in the same route, while the inter-route swap operator exchanges a node $i$ with another node $j$ located in two different routes. The intra-route reinsert operator removes one node $i$ from one route and inserts it into the middle of node $j-1$ and $j$ of the same route, while the inter-route reinsert operator removes one node $i$ from one route and inserts it into the middle of node $j-1$ and $j$ of another route. Note that the inter-route reinsert operator is able to remove the only node in a route. The resulting empty route is then deleted because it is no longer necessary. The intra-route 2-opt operator disconnects $(i-1, i)$ and $(j, j+1)$ in one route, then reconnects $(i-1, j)$ and $(i, j+1)$. The inter-route 2-opt operator removes the connection between $(i-1, i)$ and $(j-1, j)$ where $i, j$ are in two different routes; and $(i-1, j)$ and $(j-1, i)$ are reconnected, which exchanges the second sections of the two affected routes.

#### 4.2.5 Acceptance decision

The solution generated from the local search $R^{search}$ is compared to the current solution $R^{current}$. If $R^{search}$ is accepted, it becomes the current solution $R^{current}$ and $\mathcal{I}$ is reset to one. The acceptance criterion used in the heuristic was proposed by Hemmelmayr et al. (2009), and is inspired by a simulated annealing mechanism. Specifically, if $f(R^{search}) < f(R^{current})$, $R^{search}$ is always accepted. If $f(R^{search}) \geq f(R^{current})$, $R^{search}$ is accepted with the probability $e^{\frac{-(f(R^{search})-f(R^{current}))}{\eta}}$. $\eta$ is the temperature parameter which is decreased from its initial value $\eta^0$ by factor $\eta^-$ after every AVNS iteration. We also reset $\eta$ to $\eta^0$ after $\epsilon$ non-improving iterations to increase diversification of the solutions.

# 5 Computational experiments

In this section, we describe the design of the test instances for the considered robust version of the VRPTW, followed by a discussion on the parameter settings for the proposed two-stage algorithm in Subsection 5.1. Subsection 5.2 presents the results of each experiment along with a detailed analysis.

## 5.1 Experiment description and parameter settings

The test instances were derived from the well-known Solomon's instances (Solomon, 1987). These instances were designed to test the VRPTW with deterministic customer demands and travel times. The instances are grouped into six datasets, called R1, R2, C1, C2, RC1, and RC2. We refer to the names of each data set and the instances they contain interchangeably. The R1 and R2 instances include random customer locations, the C1 and C2 instances contain clustered customer locations, and the RC1 and RC2 instances contain a mix of random and clustered customer locations. However, the time windows in the R1, C1 and RC1 instances tend to be narrow whereas, in the R2, C2, and RC2 instances, they tend to be wide. We made no changes to the customer locations and time windows from the original Solomon's instances. However, the customer demands and travel times in the considered robust version of the VRPTW are uncertain and they can take any possible value from the route-dependent uncertainty sets. Thus, descriptions of the parameter values for the uncertainty sets $U_q^k$ and $U_t^k$ follow. Each nominal value $\overline{q}_i$ $(i \in N)$ and $\overline{t}_{ij}$ $((i,j) \in A)$ was assumed to equal to the corresponding customer demand and travel time in each Solomon's instance, respectively. In addition, we assumed that the maximal demand deviation $\hat{q}_i$ was $0.2\overline{q}_i$, and the maximal travel time deviation $\hat{t}_{ij}$ was $0.2\overline{t}_{ij}$. As shown in equations (11), the uncertainty budget coefficient $\theta_q$ determines the uncertainty budget $\Gamma_q^k$, and $\theta_t$ determines $\Gamma_t^k$ as per equation (13). We assumed that $\theta_q = \theta_t = 0.3$ for all R1, C1 and RC1 instances since the time windows in these instances are narrow. We set $\theta_q = \theta_t = 0.2$ for all R2, C2, and RC2 instances because these instances contain wide time windows.

The proposed two-stage algorithm was coded using Matlab 2015a, and the experiments were run on a laptop with a 2.4 GHz dual processor and 8G RAM. We conducted preliminary tests to find the ideal parameter settings for all adapted instances. In the two-stage algorithm, the modified AVNS heuristic was used to do the optimization tasks in both stages. Therefore, we set the stopping criteria $T$ and $Iter$ for the modified AVNS heuristic to 1200 seconds and 200 iterations in the first stage of the algorithm, and to 2400 seconds and 400 iterations in the second stage of the algorithm. All other parameter values for the modified AVNS heuristic were the same in both stages of the algorithm, and were set as follows: the penalty factors were $\delta_t = 1000$ and $\delta_d = 500$; $\gamma$ was set to 20 and $\rho$ was set to 0.3 in the adaptive shaking step; and $\eta^0$ was set to 20, $\eta^-$ was set to 0.1, and $\epsilon$ was set to 50 for the solution acceptance decision. All instances were tested ten times, and the best solution for each instance was selected as the result for inclusion in Subsection 5.2.

Evaluating the robustness of the generated solutions was an important issue in these experiments. Therefore, we used a Monte-Carlo simulation to assess the robustness of the final solutions. In these simulation tests, we assumed that each uncertain travel time parameter $\tilde{t}_{ij}$ $((i,j) \in A)$ followed a normal distribution with a mean equal to $\overline{t}_{ij}$ and a standard deviation of $0.2\overline{t}_{ij}$. Each uncertain demand parameter $\tilde{q}_i$ $(i \in N)$ was also assumed to have a normal distribution with a mean equal to $\overline{t}_{ij}$ and a standard deviation of $0.2\overline{t}_{ij}$. We generated 1000 scenarios which have random travel time matrices and demand vectors. The robustness of a solution was determined by calculating how many scenarios: succeeded in serving all customers; missed one customer time window at most; and missed two customer time windows at most among the 1000 generated. The number of missed customer time windows for a given solution can be counted by examining customer time windows and vehicle capacities using the current scenario's travel time matrix and demand vector. It is worth noting that, although we evaluated the robustness of the solutions based on the assumption of uncertain parameters with normal distributions, it is not necessary to know the exact distributions to produce a robust solution with the proposed robust formulation and the two-stage algorithm. The developed methods can tackle the VRPTW with uncertain parameters of any symmetric distribution as long as the parameter values are drawn from the proposed route-dependent uncertainty sets.

## 5.2 Computational Results

The following subsections present the results of a series of experiments we conducted to evaluate how a range of factors affect the solutions of the considered robust version of the VRPTW. By analyzing the effect of these factors, it is possible to investigate whether data variations in a factor or a type of uncertainty are likely to generate riskier solutions. The factors examined include demand and travel time uncertainty, customer location patterns, time window length and vehicle capacity, and some key parameters of the route-dependent uncertainty sets. A performance analysis of the proposed two-stage algorithm solving several adapted small-sized instances has also been included.

### 5.2.1 The effect of demand and travel time uncertainty

To see the detailed effects of demand and travel time uncertainty on vehicle routes, we compared the solutions of the considered robust version of the VRPTW with the solutions of the deterministic VRPTW using the R1 instances. The results of this comparison are shown in Table 2. For each test instance in Table 2, "Det" row shows the details of the deterministic solution and "Rob" row shows the details of the robust solution. The columns "N.V.", "OBJ", and "Increase" respectively represent the number of vehicles used, the total travel distance, and the percentage increase in the total travel distance of the robust solution vs. the deterministic solution in each instance. Each solution's robustness, as measured by the Monte-Carlo simulation test, is reported under the columns "Total", "Demand", and "Travel time". The columns under "Total" show the results of the simulation tests that consider both demand and travel time uncertainty. The columns under "Demand" show the results of the simulation tests that only consider demand uncertainty, while the columns under "Travel time" show the results of the simulation tests that only consider travel time uncertainty. Columns "$V_0$", "$V_1$", and "$V_2$" respectively mean the probability of serving every customer, the probability of not serving one customer at most, and the probability of not serving two customers at most within a solution. Note that the deterministic solutions for all large-sized test instances were obtained by the proposed two-stage algorithm. However, we made some small modifications to the feasibility check and objective calculation procedure discussed in Subsection 4.2.1 because uncertainty is not considered in the deterministic VRPTW.

**Table 2.** Comparison of two solution strategies on the R1 instances

| Instance | Type | N.V. | OBJ | Increase | Total | | | Demand | | | Travel time | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | $V_0$ | $V_1$ | $V_2$ | $V_0$ | $V_1$ | $V_2$ | $V_0$ | $V_1$ | $V_2$ |
| R101 | Det | 19 | 1650.80 | - | 0.000 | 0.014 | 0.054 | 1.000 | 1.000 | 1.000 | 0.000 | 0.014 | 0.054 |
| | Rob | 23 | 1763.69 | 6.84% | 0.303 | 0.649 | 0.932 | 1.000 | 1.000 | 1.000 | 0.303 | 0.649 | 0.932 |
| R102 | Det | 17 | 1486.86 | - | 0.002 | 0.016 | 0.062 | 1.000 | 1.000 | 1.000 | 0.002 | 0.016 | 0.062 |
| | Rob | 20 | 1567.24 | 5.41% | 0.353 | 0.672 | 0.868 | 1.000 | 1.000 | 1.000 | 0.353 | 0.672 | 0.868 |
| R103 | Det | 13 | 1292.67 | - | 0.000 | 0.000 | 0.000 | 0.506 | 0.718 | 0.968 | 0.000 | 0.000 | 0.001 |
| | Rob | 15 | 1294.21 | 0.12% | 0.407 | 0.765 | 0.921 | 0.967 | 0.999 | 0.999 | 0.421 | 0.776 | 0.927 |
| R104 | Det | 9 | 1008.42 | - | 0.000 | 0.001 | 0.003 | 0.437 | 0.678 | 0.930 | 0.000 | 0.002 | 0.006 |
| | Rob | 11 | 1051.55 | 4.28% | 0.664 | 0.904 | 0.966 | 0.983 | 0.992 | 1.000 | 0.677 | 0.910 | 0.969 |
| R105 | Det | 14 | 1377.11 | - | 0.005 | 0.030 | 0.090 | 1.000 | 1.000 | 1.000 | 0.005 | 0.030 | 0.090 |
| | Rob | 15 | 1434.74 | 4.18% | 0.506 | 0.822 | 0.947 | 1.000 | 1.000 | 1.000 | 0.506 | 0.822 | 0.947 |
| R106 | Det | 12 | 1275.73 | - | 0.015 | 0.073 | 0.207 | 1.000 | 1.000 | 1.000 | 0.015 | 0.073 | 0.207 |
| | Rob | 13 | 1303.95 | 2.21% | 0.503 | 0.852 | 0.964 | 1.000 | 1.000 | 1.000 | 0.503 | 0.852 | 0.964 |
| R107 | Det | 10 | 1118.06 | - | 0.002 | 0.008 | 0.025 | 0.437 | 0.675 | 0.911 | 0.004 | 0.015 | 0.055 |
| | Rob | 11 | 1143.74 | 2.30% | 0.689 | 0.896 | 0.969 | 0.976 | 0.996 | 1.000 | 0.707 | 0.904 | 0.970 |
| R108 | Det | 9 | 963.99 | - | 0.006 | 0.025 | 0.081 | 0.462 | 0.755 | 0.893 | 0.013 | 0.044 | 0.130 |
| | Rob | 10 | 976.62 | 1.31% | 0.707 | 0.900 | 0.981 | 0.933 | 0.998 | 1.000 | 0.755 | 0.914 | 0.985 |
| R109 | Det | 11 | 1197.42 | - | 0.010 | 0.035 | 0.074 | 0.633 | 0.954 | 1.000 | 0.019 | 0.048 | 0.092 |
| | Rob | 12 | 1218.79 | 1.78% | 0.794 | 0.958 | 0.994 | 0.999 | 1.000 | 1.000 | 0.794 | 0.959 | 0.994 |
| R110 | Det | 10 | 1126.63 | - | 0.011 | 0.038 | 0.095 | 0.723 | 0.910 | 0.989 | 0.013 | 0.044 | 0.107 |
| | Rob | 11 | 1154.34 | 2.46% | 0.675 | 0.891 | 0.954 | 0.995 | 1.000 | 1.000 | 0.678 | 0.892 | 0.955 |
| R111 | Det | 10 | 1103.00 | - | 0.000 | 0.007 | 0.027 | 0.941 | 0.997 | 0.999 | 0.000 | 0.007 | 0.029 |
| | Rob | 11 | 1142.14 | 3.55% | 0.662 | 0.875 | 0.958 | 0.976 | 1.000 | 1.000 | 0.678 | 0.881 | 0.960 |
| R112 | Det | 9 | 982.14 | - | 0.002 | 0.003 | 0.010 | 0.532 | 0.886 | 0.973 | 0.002 | 0.004 | 0.015 |
| | Rob | 10 | 1009.16 | 2.75% | 0.806 | 0.959 | 0.990 | 0.933 | 0.998 | 0.999 | 0.868 | 0.965 | 0.991 |

In analyzing the detailed results for the R1 instances, one clear observation is that the probability of servicing every customer with the deterministic solutions is very low. This shows that deterministic solutions are very fragile when travel times and customer demands are uncertain. The R1 instances contain randomly distributed customers with narrow time windows, short scheduling horizons, and small-capacity vehicles, which is likely the reason for this fragility. However, we observe that, in most instances, the robustness of the routes improved significantly with the robust solutions. For example, in instance R112, the probability of servicing all customers within the prescribed time window increased to 80.6% compared to

0.2% with the deterministic solution. In instance R101, this same probability only increased to 30.3% with the robust solution, but the probability of not serving one customer at most increased from 1.4% to 64.9%. We consider this to be a satisfactory result given the most of the customers with narrow time windows in the R101 instance. One interesting finding is that travel time uncertainty seems to be the main cause of route infeasibility for most of the R1 instances, whereas customer demand uncertainty does not appear to have a strong impact. For example, in the deterministic solutions for the R101, R102, R105, and R106 instances, the probability of serving every customer was 100% when only demand uncertainty was considered. This may be because some of the R1 instances contain many narrow time windows with short scheduling horizons. Hence, fewer customers are serviced on each route and demand uncertainty can be offset by vehicle capacity, yet travel time uncertainty can still easily result in violations to these narrow customer time windows.

### 5.2.2 The effect of customer location patterns

As discussed in Subsection 5.1, instances in different Solomon's datasets contain different customer location patterns. To show the effect of customer location patterns, we present the details of deterministic and robust solutions for all instances of the R1, RC1, and C1 datasets in Tables 3, 4, and 5, respectively. In these tables, the columns under "Deterministic" show the details of the each deterministic solution. The columns under "Robust" show the details of each robust solution. The columns under "Increase" show the increase in the number of vehicles used and the percentage increase in the total travel distance of the robust solution vs. the deterministic solution in each instance. Note that only the results of simulation tests that consider both demand and travel time uncertainty have been included.

**Table 3.** Results for the R1 instances

| Instance | Deterministic | | | | | Robust | | | | | Increase | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ |
| R101 | 19 | 1650.80 | 0.000 | 0.014 | 0.054 | 23 | 1763.69 | 0.303 | 0.649 | 0.932 | 4 | 6.84% |
| R102 | 17 | 1486.86 | 0.002 | 0.016 | 0.062 | 20 | 1567.24 | 0.353 | 0.672 | 0.868 | 3 | 5.41% |
| R103 | 13 | 1292.67 | 0.000 | 0.000 | 0.000 | 15 | 1294.21 | 0.407 | 0.765 | 0.921 | 2 | 0.12% |
| R104 | 9 | 1008.42 | 0.000 | 0.001 | 0.003 | 11 | 1051.55 | 0.664 | 0.904 | 0.966 | 2 | 4.28% |
| R105 | 14 | 1377.11 | 0.005 | 0.030 | 0.090 | 15 | 1434.74 | 0.506 | 0.822 | 0.947 | 1 | 4.18% |
| R106 | 12 | 1275.73 | 0.015 | 0.073 | 0.207 | 13 | 1303.95 | 0.503 | 0.852 | 0.964 | 1 | 2.21% |
| R107 | 10 | 1118.06 | 0.002 | 0.008 | 0.025 | 11 | 1143.74 | 0.689 | 0.896 | 0.969 | 1 | 2.30% |
| R108 | 9 | 963.99 | 0.006 | 0.025 | 0.081 | 10 | 976.62 | 0.707 | 0.900 | 0.981 | 1 | 1.31% |
| R109 | 11 | 1197.42 | 0.010 | 0.035 | 0.074 | 12 | 1218.79 | 0.794 | 0.958 | 0.994 | 1 | 1.78% |
| R110 | 10 | 1126.63 | 0.011 | 0.038 | 0.095 | 11 | 1154.34 | 0.675 | 0.891 | 0.954 | 1 | 2.46% |
| R111 | 10 | 1103.00 | 0.000 | 0.007 | 0.027 | 11 | 1142.14 | 0.662 | 0.875 | 0.958 | 1 | 3.55% |
| R112 | 9 | 982.14 | 0.002 | 0.003 | 0.010 | 10 | 1009.16 | 0.806 | 0.959 | 0.990 | 1 | 2.75% |
| Average | 11.9 | 1215.24 | 0.004 | 0.021 | 0.061 | 13.5 | 1255.02 | 0.589 | 0.845 | 0.954 | 1.6 | 3.27% |

**Table 4.** Results for the RC1 instances

| Instance | Deterministic | | | | | Robust | | | | | Increase | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ |
| RC101 | 14 | 1696.95 | 0.000 | 0.000 | 0.002 | 16 | 1791.28 | 0.425 | 0.725 | 0.880 | 2 | 5.56% |
| RC102 | 12 | 1554.75 | 0.000 | 0.000 | 0.003 | 14 | 1657.08 | 0.394 | 0.669 | 0.826 | 2 | 6.58% |
| RC103 | 11 | 1261.67 | 0.000 | 0.000 | 0.000 | 12 | 1399.36 | 0.567 | 0.807 | 0.924 | 1 | 10.91% |
| RC104 | 10 | 1164.15 | 0.000 | 0.007 | 0.025 | 11 | 1236.34 | 0.521 | 0.776 | 0.918 | 1 | 6.20% |
| RC105 | 14 | 1548.41 | 0.001 | 0.006 | 0.016 | 15 | 1622.50 | 0.389 | 0.700 | 0.839 | 1 | 4.79% |
| RC106 | 11 | 1424.73 | 0.000 | 0.002 | 0.005 | 13 | 1460.49 | 0.465 | 0.785 | 0.915 | 2 | 2.51% |
| RC107 | 11 | 1232.26 | 0.003 | 0.021 | 0.058 | 12 | 1314.22 | 0.568 | 0.854 | 0.960 | 1 | 6.65% |
| RC108 | 10 | 1139.82 | 0.002 | 0.006 | 0.022 | 11 | 1258.93 | 0.505 | 0.823 | 0.943 | 1 | 10.45% |
| Average | 11.6 | 1377.84 | 0.001 | 0.005 | 0.016 | 13.0 | 1467.52 | 0.479 | 0.767 | 0.901 | 1.4 | 6.51% |

**Table 5.** Results for the C1 instances

| Instance | Deterministic | | | | | Robust | | | | | Increase | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ |
| C101 | 10 | 828.94 | 0.030 | 0.107 | 0.263 | 11 | 1002.89 | 0.777 | 0.949 | 0.990 | 1 | 20.99% |
| C102 | 10 | 828.94 | 0.025 | 0.086 | 0.246 | 11 | 980.92 | 0.705 | 0.939 | 0.986 | 1 | 18.33% |
| C103 | 10 | 828.06 | 0.037 | 0.164 | 0.379 | 11 | 975.56 | 0.641 | 0.913 | 0.987 | 1 | 17.81% |
| C104 | 10 | 824.78 | 0.018 | 0.110 | 0.300 | 11 | 960.50 | 0.688 | 0.919 | 0.987 | 1 | 16.45% |
| C105 | 10 | 828.94 | 0.021 | 0.107 | 0.259 | 11 | 996.20 | 0.708 | 0.906 | 0.984 | 1 | 20.18% |
| C106 | 10 | 828.94 | 0.028 | 0.096 | 0.242 | 11 | 998.58 | 0.737 | 0.958 | 0.993 | 1 | 20.46% |
| C107 | 10 | 828.94 | 0.021 | 0.089 | 0.023 | 11 | 993.28 | 0.676 | 0.919 | 0.985 | 1 | 19.82% |
| C108 | 10 | 828.94 | 0.023 | 0.102 | 0.249 | 11 | 980.19 | 0.756 | 0.954 | 0.996 | 1 | 18.25% |
| C109 | 10 | 828.94 | 0.019 | 0.082 | 0.236 | 11 | 976.13 | 0.789 | 0.964 | 0.993 | 1 | 17.76% |
| Average | 10.0 | 828.38 | 0.025 | 0.105 | 0.244 | 11.0 | 984.92 | 0.720 | 0.936 | 0.989 | 1.0 | 18.90% |

A detailed analysis of the effect of customer location patterns produces similar conclusions to the above. The deterministic solutions for the instances in all three datasets are very fragile, especially the solutions for the R1 and RC1 instances. For example, the average probability of servicing every customer was only 0.4% with the deterministic solutions for the R1 instances and 0.1% with the deterministic solutions for the RC1 instances. However, in the robust solutions, the average number of vehicles used for the R1 instances increased by 1.6, and the average total travel distance increased by 3.27% compared to the deterministic solutions. Additionally, the average probability of servicing every customer increased from 0.4% to 58.9%, and the average probability of not serving one customer at most increased from 2.1% to 84.5%, which shows that the robustness of the solutions has been improved significantly. The robust solutions for the RC1 instances show similar results. However, the average total travel distance of the robust solutions for the C1 instances increased by nearly 19% compared to the deterministic solutions because of the one additional vehicle included in the robust solution must travel through different customer clusters. This additional vehicle significantly increases the total travel distance in the robust solution for each C1 instance. So, although the robust solution greatly reduces the risk of a missed customer time window, decision-makers may still need to assess the trade-off between total travel distance and route robustness in situations with clustered location patterns.

### 5.2.3 The effect of time window length and vehicle capacity

Contrary to the instances in the R1, C1 and RC1 datasets, the R2, RC2, and C2 instances tend to contain wide customer time windows with a long scheduling horizon and large-capacity vehicles. These characteristics allow many customers to be served along the same route. To show the effect of wide customer time windows and large-capacity vehicles on the planned routes, we present the details of the deterministic and robust solutions for the R2, RC2 and C2 instances in Tables 6, 7, and 8, respectively.

**Table 6.** Results for the R2 instances

| Instance | Deterministic | | | | | Robust | | | | | Increase | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ |
| R201 | 4 | 1254.80 | 0.138 | 0.469 | 0.791 | 4 | 1286.34 | 0.886 | 0.993 | 1.000 | 0 | 2.51% |
| R202 | 3 | 1200.40 | 0.061 | 0.200 | 0.390 | 4 | 1120.06 | 0.884 | 0.990 | 1.000 | 1 | -6.69% |
| R203 | 3 | 948.14 | 0.262 | 0.711 | 0.937 | 3 | 969.15 | 0.895 | 0.996 | 0.999 | 0 | 2.22% |
| R204 | 2 | 833.65 | 0.251 | 0.677 | 0.900 | 2 | 877.51 | 0.925 | 0.996 | 0.999 | 0 | 5.26% |
| R205 | 3 | 1005.94 | 0.471 | 0.716 | 0.838 | 3 | 1066.75 | 0.976 | 1.000 | 1.000 | 0 | 6.04% |
| R206 | 3 | 914.10 | 0.367 | 0.810 | 0.968 | 3 | 937.79 | 0.966 | 0.996 | 1.000 | 0 | 2.59% |
| R207 | 2 | 900.63 | 0.152 | 0.433 | 0.692 | 3 | 836.95 | 1.000 | 1.000 | 1.000 | 1 | -7.07% |
| R208 | 2 | 729.73 | 0.543 | 0.717 | 0.865 | 2 | 737.24 | 1.000 | 1.000 | 1.000 | 0 | 1.03% |
| R209 | 3 | 913.14 | 0.231 | 0.545 | 0.822 | 3 | 942.76 | 0.979 | 1.000 | 1.000 | 0 | 3.24% |
| R210 | 3 | 947.88 | 0.341 | 0.703 | 0.886 | 3 | 967.99 | 0.996 | 1.000 | 1.000 | 0 | 2.12% |
| R211 | 2 | 885.71 | 0.206 | 0.386 | 0.607 | 3 | 782.80 | 0.976 | 1.000 | 1.000 | 1 | -11.62% |
| Average | 2.7 | 957.65 | 0.275 | 0.579 | 0.791 | 3.0 | 956.85 | 0.953 | 0.997 | 1.000 | 0.3 | -0.08% |

**Table 7.** Results for the RC2 instances

| Instance | Deterministic | | | | | Robust | | | | | Increase | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ |
| RC201 | 4 | 1413.52 | 0.256 | 0.551 | 0.784 | 4 | 1454.15 | 0.897 | 0.991 | 0.999 | 0 | 2.87% |
| RC202 | 3 | 1368.14 | 0.061 | 0.178 | 0.313 | 4 | 1176.55 | 0.956 | 1.000 | 1.000 | 1 | -14.00% |
| RC203 | 3 | 1064.14 | 0.137 | 0.330 | 0.546 | 3 | 1129.51 | 0.932 | 0.998 | 1.000 | 0 | 6.14% |
| RC204 | 3 | 800.28 | 0.197 | 0.509 | 0.754 | 3 | 830.14 | 0.977 | 1.000 | 1.000 | 0 | 3.73% |
| RC205 | 4 | 1297.65 | 0.086 | 0.335 | 0.692 | 4 | 1346.95 | 0.970 | 1.000 | 1.000 | 0 | 3.80% |
| RC206 | 3 | 1146.32 | 0.166 | 0.474 | 0.718 | 3 | 1243.13 | 0.950 | 0.990 | 0.996 | 0 | 8.45% |
| RC207 | 3 | 1061.14 | 0.183 | 0.423 | 0.606 | 3 | 1123.11 | 0.903 | 0.990 | 1.000 | 0 | 5.84% |
| RC208 | 3 | 832.36 | 0.269 | 0.522 | 0.799 | 3 | 865.66 | 0.990 | 0.999 | 0.999 | 0 | 4.00% |
| Average | 3.3 | 1122.94 | 0.169 | 0.415 | 0.652 | 3.4 | 1146.15 | 0.947 | 0.996 | 0.999 | 0.1 | 2.07% |

**Table 8.** Results for the C2 instances

| Instance | Deterministic | | | | | Robust | | | | | Increase | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ |
| C201 | 3 | 591.56 | 0.505 | 0.665 | 0.998 | 3 | 621.51 | 0.996 | 1.000 | 1.000 | 0 | 5.06% |
| C202 | 3 | 591.56 | 0.505 | 0.663 | 0.999 | 3 | 607.64 | 0.992 | 0.999 | 1.000 | 0 | 2.72% |
| C203 | 3 | 591.17 | 0.496 | 0.665 | 1.000 | 3 | 600.21 | 0.993 | 0.999 | 1.000 | 0 | 1.53% |
| C204 | 3 | 590.60 | 0.530 | 0.676 | 1.000 | 3 | 599.29 | 0.987 | 0.997 | 1.000 | 0 | 1.47% |
| C205 | 3 | 588.88 | 0.674 | 0.792 | 1.000 | 3 | 603.48 | 0.998 | 0.999 | 1.000 | 0 | 2.48% |
| C206 | 3 | 588.49 | 0.364 | 0.728 | 0.905 | 3 | 603.48 | 0.998 | 1.000 | 1.000 | 0 | 2.55% |
| C207 | 3 | 588.29 | 0.635 | 0.761 | 0.969 | 3 | 605.96 | 0.998 | 1.000 | 1.000 | 0 | 3.00% |
| C208 | 3 | 588.32 | 0.626 | 0.767 | 0.998 | 3 | 602.93 | 0.997 | 0.999 | 1.000 | 0 | 2.48% |
| Average | 3.0 | 589.86 | 0.542 | 0.715 | 0.984 | 3.0 | 605.56 | 0.995 | 0.999 | 1.000 | 0.0 | 2.66% |

Based on the results, it is clear that the deterministic solutions for the R2, RC2, and C2 instances are more robust than the deterministic solutions for the R1, RC1 and C1 instances. For example, the average probability of servicing every customer with the deterministic solutions for the C2 instances was 54.2%, which indicates that these deterministic solutions are not very fragile. In the robust solutions for the R2, RC2, and C2 instances, the total travel distance only increased a little, and the number of vehicles used stayed the same in most of the instances. However, the route robustness with these solutions improved significantly. For example, the average probability of serving every customers was 95.3%, 94.7%, and 99.5% for the R2, RC2 and C2 instances, respectively. These results indicate that the solutions are very robust given both travel time and demand uncertainty. We can therefore conclude that a high level of route robustness can be reached for instances with wide time windows and large-capacity vehicles using the robust solutions at almost no additional cost (i.e., little increase in the total travel distance and the number of vehicles used). However, failing to serve all customers is more prevalent in deterministic solutions given instances with narrow time windows, and ensuring a high level of route robustness is much more expensive, as shown in Tables 3, 4, and 5.

From Tables 6 and 7, one interesting finding is that the robust solutions use more vehicles but travel less total distance than the deterministic solutions, and yet still improve the route robustness significantly in some of the R2 and RC2 instances. For example, in instance R202, the robust solution used one more vehicle than the deterministic solution, but the total travel distance was 6.69% less and the probability of serving all customers increased to 88.4%. Instances R207, R211, and RC202 show similar results. Thus, robust solutions may be very attractive in some instances because they can decrease the total travel distance and increase the route robustness with the addition of only a small number of vehicles. It is also worth noting that adding more vehicles does not necessarily decrease the total travel distance, but this can be effective strategy for improving the robustness of the planned routes.

### 5.2.4 The effect of uncertainty set parameters

A solution's robustness is affected by the parameters of the route-dependent uncertainty sets outlined in equations (10)-(13). Hence, our next set of experiments were designed to analyze the effects of the uncertainty budget coefficients $\theta_q$ and $\theta_t$ and the maximal deviation of uncertain demand $\hat{q}_i$ $(i \in N_c)$ and travel time $\hat{t}_{ij}$ $((i,j) \in A)$. Instances RC101 and RC201 were selected for analysis, and the assumptions made during these experiments follow. The ratio $\nu_q$ between the maximal demand deviation $\hat{q}_i$ and its nominal value $\overline{q}_i$ stayed the same for all customer nodes $i \in N_c$; the ratio $\nu_t$ between the maximal travel time deviation $\hat{t}_{ij}$ and its nominal value $\overline{t}_{ij}$ stayed the same for all arcs $(i,j) \in A$. These ratios are referred to as the uncertainty ranges. Note that we analyzed the uncertainty ranges and the uncertainty budget coefficients separately. When analyzing the effect of the uncertainty budgets, the uncertainty ranges were assumed to be fixed, and were set to $\nu_q = \nu_t = 0.2$. When analyzing the effect of the uncertainty ranges, the uncertainty budget coefficients were assumed to be fixed, and were set to $\theta_q = \theta_t = 0.3$ for the RC101 instance and to $\theta_q = \theta_t = 0.2$ for the RC201 instance. The robust solutions for the RC101 and RC201 instances given different uncertainty ranges are shown in Table 9. The robust solutions given different uncertainty budget coefficients are shown in Table 10.

**Table 9.** Results for instances RC101 and RC201 with different uncertainty range values

| Uncertainty ranges | RC101 | | | | | RC201 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ |
| 0 | 14 | 1696.95 | 0.000 | 0.000 | 0.002 | 4 | 1413.52 | 0.075 | 0.298 | 0.570 |
| 0.1 | 15 | 1683.06 | 0.039 | 0.163 | 0.350 | 4 | 1446.23 | 0.847 | 0.988 | 0.999 |
| 0.2 | 16 | 1791.28 | 0.425 | 0.725 | 0.880 | 4 | 1454.15 | 0.897 | 0.991 | 0.999 |
| 0.3 | 18 | 1855.76 | 0.761 | 0.930 | 0.972 | 4 | 1495.56 | 0.886 | 0.994 | 1.000 |
| 0.4 | 19 | 1906.10 | 0.938 | 0.988 | 0.993 | 4 | 1603.16 | 1.000 | 1.000 | 1.000 |
| 0.5 | 20 | 2139.63 | 0.983 | 0.996 | 1.000 | 5 | 1422.77 | 1.000 | 1.000 | 1.000 |

Let us first focus on the effect of different uncertainty ranges with the robust solution for the RC101 instance, shown in Table 9. As the value of the uncertainty ranges increased from 0 to 0.5, the number of vehicles used increased from 14 to 20 and the total travel distance increased from 1696.95 to 2139.63 with the robust solution. Instance RC101 contains many narrow time windows, small-capacity vehicles, and a short scheduling horizon. Thus, more vehicles need to be scheduled to meet the prescribed time windows as the value of the uncertainty ranges increases and this results in a greater total travel distance. However, the effect of varying uncertainty ranges with the RC201 instance is quite different. As the value of the uncertainty ranges increased from 0 to 0.4, the number of vehicles used stayed the same but the total travel distance increased from 1413.52 to 1603.16 with the robust solution. Instance RC201 contains wide customer time windows, large-capacity vehicles, and a long scheduling horizon. Hence, the sequence in which each customer is visited along a route can simply be reordered to achieve a high level of route robustness without increasing the number of vehicles used when the value of the uncertainty ranges is relatively small. However, as the value of the uncertainty ranges increased from 0.4 to 0.5, the number of vehicles used increased by one and the total travel distance decreased from 1603.16 to 1422.77. The reason is that one more vehicle is needed to deal with the greater levels of uncertainty and the vehicle routes are rearranged which significantly decreases the total travel distance.

**Table 10.** Results for instances RC101 and RC201 with different uncertainty budget coefficient values

| Uncertainty budget | RC101 | | | | | RC201 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| coefficients | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ | N.V. | OBJ | $V_0$ | $V_1$ | $V_2$ |
| 0 | 14 | 1696.95 | 0.000 | 0.000 | 0.002 | 4 | 1413.52 | 0.075 | 0.298 | 0.570 |
| 0.2 | 16 | 1763.69 | 0.252 | 0.537 | 0.806 | 4 | 1454.15 | 0.897 | 0.991 | 0.999 |
| 0.4 | 16 | 1800.56 | 0.402 | 0.741 | 0.908 | 4 | 1477.13 | 0.915 | 0.993 | 0.999 |
| 0.6 | 16 | 1814.30 | 0.491 | 0.797 | 0.900 | 4 | 1477.13 | 0.915 | 0.993 | 0.999 |
| 0.8 | 17 | 1770.82 | 0.494 | 0.812 | 0.907 | 4 | 1477.13 | 0.915 | 0.993 | 0.999 |
| 1 | 17 | 1792.92 | 0.562 | 0.869 | 0.962 | 4 | 1477.13 | 0.915 | 0.993 | 0.999 |

Next, we turn to the effect of the uncertainty budget coefficients with the robust solution for the RC101 instance, shown in Table 10. As the value of uncertainty budget coefficients increased from 0 to 0.2, the number of vehicles used increased by two and the total travel distance increased from 1696.95 to 1763.69. Since instance RC101 contains many narrow time windows, more vehicles are needed to serve customers on time when the value of the coefficients increases by a small amount. However, as the value of the uncertainty budget coefficients increased from 0.2 to 0.6, the number of vehicles used stayed the same and the total travel distance slightly increased. The reason is that the sequence of customer visits within each route can be rearranged to handle the greater levels of uncertainty without adding more vehicles. As the value of the coefficients increased from 0.6 to 0.8, the number of vehicles used increased from 16 to 17 and the total travel distance decreased from 1814.30 to 1770.82. When the coefficient value reached 1, the number of vehicles used stayed with 17 and the total travel distance increased by a small amount with the robust solution. Compared to the RC101 instance, the uncertainty budget coefficients do not have as strong an effect on the robust solution of the RC201 instance. As shown in Table 10, the number of vehicles used stayed the same and the total travel distance increased from 1413.52 to 1477.13 as the value of the uncertainty budget coefficients increased from 0 to 0.4, after which the robust solution stayed the same in all respects. Although, notably, the resulting solution with a 90% probability that every customer will be served had a high level of route robustness.

Based on the above analysis, we can conclude that adjusting the uncertainty ranges and/or the uncertainty budget coefficients can affect the solutions generated for the considered robust version of the VRPTW. The results also highlight two insights for decision-makers: (1) decision-makers should select routing strategies with a high level of robustness for practical applications characterized by wide customer time windows and large-capacity vehicles; and (2) decision-makers may have to balance the total travel distance and the number of vehicles used (i.e., the cost) with the robustness of the planned routes when selecting an ideal routing strategy for practical applications characterized by narrow customer time windows and small-capacity vehicles.

### 5.2.5 Algorithm performance

To test the performance of our proposed two-stage algorithm, we adapted 24 of Solomon's instances by selecting the first ten customers in each instance. Given the vehicle capacity of each Solomon's instance is too large to be meaningful, we reset the vehicle capacity to 75 for the R1 and R2 instances, to 100 for the C1 and C2 instances, and to 150 for the RC1 and RC2 instances. We also reset the uncertainty budget coefficients to $\theta_q = \theta_t = 0.6$. The stopping criteria $T$ and $Iter$ for the modified AVNS heuristic were set to 50 seconds and 20 iterations in both stages of the two-stage algorithm due to the small size of the instances. We retained the values described in Subsection 5.1 for all other parameters of the modified AVNS heuristic. We tested the proposed two-stage algorithm ten times with each instance and compared the resulting solutions to the optimal solutions generated by solving the robust formulation outlined in Subsection 3.2. The comparison results are shown in Table 11. Note that the column-and-row generation algorithm discussed in Agra et al. (2013) was adapted to solve the robust formulation. The algorithm was coded in GAMS software with the CPLEX 12.3 solver. To further assess the quality of the proposed algorithm, we solved a robust version of the VRP with deadlines under demand and travel time uncertainty discussed by Lee et al. (2012). Lee et al. (2012) developed an exact algorithm to solve the problem with the adapted small-sized Solomon's R1 instances. However, the robust version of the VRP with deadlines considers only one objective and different uncertainty sets. Thus, we adapted the modified AVNS heuristic described in Subsection 4.2 to solve the problem. Each instance was also tested ten times. The comparison between the exact algorithm in Lee et al. (2012) and the modified AVNS heuristic is shown in Table 12.

**Table 11.** Comparison results for the adapted Solomon's instances with 10 customers

| Instance | AC&RG algorithm | | | Two-stage algorithm | | | |
|---|---|---|---|---|---|---|---|
| | N.V. | Optimal | Time (s) | N.V. | Best | Avg | Time avg (s) |
| R101 | 4 | 287.34 | 0.94 | 4 | 287.34 | 287.34 | 8.72 |
| R102 | 4 | 262.19 | 393.83 | 4 | 262.19 | 262.19 | 6.35 |
| R111 | 2 | 237.40 | 2.61 | 2 | 237.40 | 237.40 | 7.46 |
| R112 | 2 | 198.21 | 11.17 | 2 | 198.21 | 198.21 | 7.13 |
| R201 | 2 | 259.58 | 0.69 | 2 | 259.58 | 259.58 | 6.42 |
| R202 | 2 | 198.21 | 10.58 | 2 | 198.21 | 198.21 | 6.67 |
| R210 | 2 | 198.21 | 2.36 | 2 | 198.21 | 198.21 | 7.57 |
| R211 | 2 | 198.21 | 12.34 | 2 | 198.21 | 198.21 | 7.34 |
| C101 | 2 | 90.19 | 1.09 | 2 | 90.19 | 90.19 | 8.26 |
| C102 | 2 | 90.19 | 422.04 | 2 | 90.19 | 90.19 | 8.14 |
| C108 | 2 | 89.87 | 0.95 | 2 | 89.87 | 89.87 | 7.21 |
| C109 | 2 | 89.87 | 25.93 | 2 | 89.87 | 89.87 | 8.36 |
| C201 | 2 | 176.49 | 3.06 | 2 | 176.49 | 176.49 | 8.13 |
| C202 | 2 | 162.36 | 37.71 | 2 | 162.36 | 162.36 | 9.12 |
| C207 | 2 | 176.49 | 3.64 | 2 | 176.49 | 176.49 | 7.54 |
| C208 | 2 | 168.84 | 4.18 | 2 | 168.84 | 168.84 | 7.21 |
| RC101 | 3 | 239.31 | 6.78 | 3 | 239.31 | 239.31 | 6.53 |
| RC102 | 2 | 203.91 | 220.52 | 2 | 203.91 | 203.91 | 8.39 |
| RC107 | 2 | 202.30 | 212.55 | 2 | 202.30 | 202.30 | 7.29 |
| RC108 | 2 | 202.68 | 252.49 | 2 | 202.68 | 202.68 | 7.37 |
| RC201 | 2 | 212.33 | 4.79 | 2 | 212.33 | 212.33 | 7.14 |
| RC202 | 2 | 203.91 | 499.22 | 2 | 203.91 | 203.91 | 7.65 |
| RC207 | 2 | 204.80 | 152.23 | 2 | 204.80 | 204.80 | 8.23 |
| RC208 | 2 | 202.30 | 583.42 | 2 | 202.30 | 202.30 | 6.82 |

In Table 11, the columns under ”AC&RG algorithm” show the details of the solutions obtained by adapting the column-and-row generation algorithm discussed in Agra et al. (2013). The columns ”N.V.”, ”Optimal”, and ”Time” represent the number of vehicles used, the optimal objective value, and the computation time of the algorithm for each instance, respectively. Similarly, the columns under ”Two-stage algorithm” show the details of the solutions obtained using the proposed two-stage algorithm. The columns ”Best”, ”Avg”, and ”Time Avg” represent the best objective value, the average objective value, and the average computational time of ten runs, respectively. Table 11 shows that the proposed two-stage algorithm was able to find the optimal solution for the considered robust VRPTW with each small-sized instance. The two-stage algorithm was very stable in terms of computation time, taking less time to generate solutions than the adapted column-and-row generation algorithm for more than half the test instances. However, the adapted column-and-row generation algorithm was faster when solving several instances including R101, R111, R201, R210, C101, C108, C201, and RC201. The reason is related to the instance structures. For example, the R101 and C101 instances contain many narrow time windows and small-capacity vehicles. This type of instance structure is particularly suited to the CPLEX solver, and hence it is able to find an optimal solution in every iteration of the algorithm very quickly. For instances with many wide time windows and large-capacity vehicles such as instances R201, C201 and RC201, the adapted column-and-row generation algorithm only requires a few iterations to generate the optimal robust solution.

**Table 12.** Comparison results for the instances in Lee et al. (2012)

| Instance | Lee et al. (2012) | | Modified AVNS | | |
|---|---|---|---|---|---|
| | Optimal | Time (s) | Best | Avg | Time avg (s) |
| R101 | 464.4 | 3.62 | 464.4 | 464.4 | 9.14 |
| R102 | 444.7 | 201.09 | 444.7 | 444.7 | 8.36 |
| R103 | 417.3 | 1440.65 | 417.3 | 417.3 | 7.89 |
| R104 | 407.9 | 3517.91 | 407.9 | 407.9 | 10.47 |
| R105 | 446.6 | 41.14 | 446.6 | 446.6 | 7.23 |
| R106 | 426.3 | 465.82 | 426.3 | 427.9 | 6.57 |
| R107 | 408.3 | 3600 | 408.3 | 408.3 | 8.05 |
| R108 | 399.3 | 836.41 | 399.3 | 399.5 | 7.32 |
| R109 | 429.4 | 1095.92 | 429.4 | 430.9 | 11.95 |
| R110 | 409.0 | 471.67 | 409.0 | 409.0 | 9.26 |
| R111 | 417.3 | 3600 | 417.3 | 417.3 | 12.02 |
| R112 | 399.3 | 2573.76 | 399.3 | 399.5 | 11.57 |

In Table 12, the columns under "Lee et al. (2012)" show the details of the solutions obtained by the exact algorithm proposed in Lee et al. (2012). The columns "Optimal" and "Time" represent the optimal objective value and the computation time of the exact algorithm for each instance, respectively. The columns under "Modified AVNS" summarize the details of the solutions obtained by the modified AVNS heuristic. The columns "Best", "Avg", and "Time Avg" represent the best objective value, the average objective value, and the average computational time of ten runs, respectively. Table 12 clearly shows that the best solution produced by the heuristic is the same as the optimal solution generated by the exact algorithm for each instance. Moreover, with the exception of the first instance, the modified AVNS heuristic was much faster than the exact algorithm.

# 6 Conclusions

In this paper, we considered a robust version of the VRPTW with demand and travel time uncertainty. To address the problem, we assumed the uncertain parameters took values from the route-dependent uncertainty sets and adapted a robust mathematical formulation. Due to the complexity of the problem, the adapted robust model was only able to tackle small-sized instances using standard solvers. To solve large-sized instances, we proposed a two-stage algorithm based on a modified AVNS heuristic. Extensive computational experiments were conducted using modified versions of Solomon's benchmark instances. The results show that the two-stage algorithm was able to produce optimal solutions for small-sized instances and the planned routes generated for large-sized instances were significantly more robust at the cost of a small increase in the total travel distance and the number of vehicles used. We performed a comprehensive analysis of the results and several managerial insights are revealed as follows:

(1) Generating vehicle routes without considering demand and travel time uncertainty yields very fragile routing strategies that often result in missed time windows.

(2) Incorporating more vehicles into a routing schedule and reordering the sequence of customer visits within each vehicle route can sometimes increase the robustness of a given routing strategy.

(3) Highly robust routing strategies can be generated for settings characterized by wide customer time windows and large-capacity vehicles at little additional cost.

(4) Failing to serve a customer within the prescribed time window in situations characterized by narrow time windows and small-capacity vehicles is more prevalent when the routing strategy is generated with a deterministic model. And reaching a high level of robustness is much more expensive - more vehicles and much longer travel distances are required.

# References

Adulyasak, Y. and Jaillet, P. (2015). Models and algorithms for stochastic and robust vehicle routing with deadlines. *Transportation Science*, 50(2):608–626.

Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L. M., Poss, M., and Requejo, C. (2012). Layered formulation for the robust vehicle routing problem with time windows. In *International Symposium on Combinatorial Optimization*, pages 249–260. Springer.

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

Ando, N. and Taniguchi, E. (2006). Travel time reliability in vehicle routing and scheduling with time windows. *Networks and spatial economics*, 6(3):293–311.

Baldacci, R., Mingozzi, A., and Roberti, R. (2011). New route relaxation and pricing strategies for the vehicle routing problem. *Operations research*, 59(5):1269–1283.

Baldacci, R., Mingozzi, A., and Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1):1–6.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.

Ben-Tal, A. and Nemirovski, A. (2002). Robust optimization–methodology and applications. *Mathematical Programming*, 92(3):453–480.

Bertsimas, D., Brown, D. B., and Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM review*, 53(3):464–501.

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

Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations research*, 52(1):35–53.

Braaten, S., Gjønnes, O., Hvattum, L. M., and Tirado, G. (2017). Heuristics for the robust vehicle routing problem with time windows. *Expert Systems with Applications*, 77:136–147.

Bräysy, O. and Gendreau, M. (2005a). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation science*, 39(1):104–118.

Bräysy, O. and Gendreau, M. (2005b). Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation science*, 39(1):119–139.

Caceres, H., Batta, R., and He, Q. (2017). School bus routing with stochastic demand and duration constraints. *Transportation Science*.

Chang, M.-S. (2005). A vehicle routing problem with time windows and stochastic demands. *Journal of the Chinese institute of engineers*, 28(5):783–794.

Cordeau, J.-F., Laporte, G., and Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, 52(8):928–936.

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1):80–91.

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

Dondo, R. and Cerdá, J. (2007). A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research*, 176(3):1478–1507.

Ehmke, J. F., Campbell, A. M., and Urban, T. L. (2015). Ensuring service levels in routing problems with time windows and stochastic travel times. *European Journal of Operational Research*, 240(2):539–550.

Gendreau, M., Jabali, O., and Rei, W. (2016). 50th anniversary invited articlefuture research directions in stochastic vehicle routing. *Transportation Science*, 50(4):1163–1173.

Gounaris, C. E., Repoussis, P. P., Tarantilis, C. D., Wiesemann, W., and Floudas, C. A. (2014). An adaptive memory programming framework for the robust capacitated vehicle routing problem. *Transportation Science*, 50(4):1239–1260.

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

Han, J., Lee, C., and Park, S. (2013). A robust scenario approach for the vehicle routing problem with uncertain travel times. *Transportation science*, 48(3):373–390.

Hemmelmayr, V. C., Doerner, K. F., and Hartl, R. F. (2009). A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, 195(3):791–802.

Ibaraki, T., Imahori, S., Kubo, M., Masuda, T., Uno, T., and Yagiura, M. (2005). Effective local search algorithms for routing and scheduling problems with general time-window constraints. *Transportation science*, 39(2):206–232.

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

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

Kumar, S. N. and Panneerselvam, R. (2012). A survey on the vehicle routing problem and its variants. *Intelligent Information Management*, 4(3):66.

Lee, C., Lee, K., and Park, S. (2012). Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society*, 63(9):1294–1306.

Lei, H., Laporte, G., and Guo, B. (2011). The capacitated vehicle routing problem with stochastic demands and time windows. *Computers & Operations Research*, 38(12):1775–1783.

Li, X., Tian, P., and Leung, S. C. (2010). Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *International Journal of Production Economics*, 125(1):137–145.

Mester, D. and Bräysy, O. (2005). Active guided evolution strategies for large-scale vehicle routing problems with time windows. *Computers & Operations Research*, 32(6):1593–1614.

Miranda, D. M. and Conceição, S. V. (2016). The vehicle routing problem with hard time windows and stochastic travel and service time. *Expert Systems with Applications*, 64:104–116.

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & operations research*, 24(11):1097–1100.

Nanry, W. P. and Barnes, J. W. (2000). Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B: Methodological*, 34(2):107–121.

Ong, H., Ang, B., Goh, T., and Deng, C. (1997). A vehicle routing and scheduling problem with time windows and stochastic demand constraints. *Asia-Pacific Journal of Operational Research*, 14(1):1.

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

Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & operations research*, 34(8):2403–2435.

Ritzinger, U., Puchinger, J., and Hartl, R. F. (2016). A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54(1):215–231.

Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4):455–472.

Russell, R. and Urban, T. (2008). Vehicle routing with soft time windows and erlang travel times. *Journal of the Operational Research Society*, 59(9):1220–1228.

Solano-Charris, E., Prins, C., and Santos, A. C. (2015). Local search based metaheuristics for the robust vehicle routing problem with discrete scenarios. *Applied Soft Computing*, 32:518–531.

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

Stenger, A., Vigo, D., Enz, S., and Schwind, M. (2013). An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transportation Science*, 47(1):64–80.

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

Taş, D., Dellaert, N., Van Woensel, T., and De Kok, T. (2013). Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research*, 40(1):214–224.

Taş, D., Dellaert, N., van Woensel, T., and de Kok, T. (2014). The time-dependent vehicle routing problem with soft time windows and stochastic travel times. *Transportation Research Part C: Emerging Technologies*, 48:66–83.

Thompson, P. M. and Psaraftis, H. N. (1993). Cyclic transfer algorithm for multivehicle routing and scheduling problems. *Operations research*, 41(5):935–946.

Toklu, N. E., Gambardella, L. M., and Montemanni, R. (2014). A multiple ant colony system for a vehicle routing problem with time windows and uncertain travel times. *Journal of Traffic and Logistics Engineering*, 2(1).

Toth, P. and Vigo, D. (2014). *Vehicle routing: problems, methods, and applications*. SIAM.

Zhang, J., Lam, W. H., and Chen, B. Y. (2016). On-time delivery probabilistic models for the vehicle routing problem with stochastic demands and time windows. *European Journal of Operational Research*, 249(1):144–154.