Layered Formulation for the Robust Vehicle Routing Problem with Time Windows

Agostinho Agra[1], Marielle Christiansen[2], Rosa Figueiredo[1], Lars Magnus Hvattum[2], Michael Poss[3], and Cristina Requejo[1]

[1] CIDMA, Department of Mathematics, University of Aveiro, 3810-193 Aveiro, Portugal
{aagra,rosa.figueiredo,crequejo}@ua.pt
[2] Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway
{marielle.christiansen,lars.m.hvattum}@iot.ntnu.no
[3] Computer Science Department, Free University of Brussels, B-1050 Brussels, Belgium
mposs@ulb.ac.be

Abstract. This paper studies the vehicle routing problem with time windows where travel times are uncertain and belong to a predetermined polytope. The objective of the problem is to find a set of routes that services all nodes of the graph and that are feasible for all values of the travel times in the uncertainty polytope. The problem is motivated by maritime transportation where delays are frequent and must be taken into account. We present an extended formulation for the vehicle routing problem with time windows that allows us to apply the classical (static) robust programming approach to the problem. The formulation is based on a layered representation of the graph, which enables to track the position of each arc in its route. We test our formulation on a test bed composed of maritime transportation instances.

Keywords: vehicle routing problem, robust programming, time windows, maritime transportation, layered formulation

1 Introduction

In this paper, we study the vehicle routing problem with time windows in the uncertain context. Given a graph with a special node called depot and a set of vehicles, the vehicle routing problem aims at prescribing routes for the vehicles starting at and returning to the depot in such a way that each remaining node of the graph is visited by exactly one vehicle. The problem has numerous applications in transportation, distribution and logistics, see [11]. In this work, we are more particularly interested by an application that arises in maritime transportation [8].

Among the many versions of the vehicle routing problem that have been studied in the literature, we consider the problem where time windows are given for each node of the network, yielding the vehicle routing problem with time windows ($VRPTW$). Hence, each node must be serviced during specific time intervals (or time windows) and traveling along the arcs consumes time. Most authors consider in addition that each vehicle has also a capacity that can not be exceeded along its route. In this work however, we consider the problem version without the capacity constraint, often called $m{-}TSPTW$ in the literature. This assumption is motivated by our application in maritime transportation where each ship carries only one cargo at the time, from the loading port to the unloading port. Since it is straightforward to extend our model and solution method to the problem with the capacity constraint, we keep the notation $VRPTW$ in what follows.

Exact solution methods for the $VRPTW$ have been studied extensively and many integer programming formulations have been proposed for the problem, see the reviews [9,13]. Among them, Bard [2] studies the resource inequalities formulation for the problem. The formulation from [2] contains two sets of variables: arc variables state which arcs belong to the solution and node variables indicate at what time vehicles arrive at each node. Kallehauge et al. [14] extend to the $VRPTW$ the path inequalities formulation proposed in [1] for the asymmetric traveling salesman with time windows. In [14], routes or paths that cannot satisfy the time windows are cut-off by path inequalities. Formulations based on path variables are also very popular for all versions of vehicle routing problems, including the $VRPTW$, see [9,15], among others. These formulations contain a little number of constraints but a very large number of variables so that efficient branch-and-price algorithms are required.

In this work, we study the $VRPTW$ in the uncertain context where travel times are not known with precision and belong to an uncertainty polytope $\mathcal{T}$. Hence, our approach falls into the framework of robust programming. Robust programming stems from the original work of [18] and has witnessed a continuous attention it the last ten years. We refer the interested reader to the survey [3] and the book [5].

The robust vehicle routing problem with time windows and uncertain travel times ($\mathcal{T}$-$VRPTW$) has been mentioned already in [20]. However, their modeling assumption leads to all travel times taking their maximum values, which is an over-conservative model. In fact, [20] mainly focus on the robust capacitated vehicle routing problem, see also [17]. The literature on the stochastic version of the $VRPTW$ is also scant. Up to our knowledge, [7] is the only previous work that considers stochastic travel times. Hence, the present work is the first general approach to the robust vehicle routing problem with time windows and uncertain travel times. Travel times belong here to a demand uncertainty polytope, which makes the problem harder to solve, yet tractable, than its deterministic counterpart. The retribution of the addition in complexity is that our model is more flexible than the one from [20] and leads to less conservative robust solutions.

Our objective in this paper is to make use of the classical framework of (static) robust programming. In that framework, a vector is feasible for the problem if and only if it is feasible for all values of the travel time in the uncertainty polytope

$\mathcal{T}$. Hence, the formulation based on resource inequalities cannot be used in that context because it makes no sense to choose arrival times that are independent of the travel times (see Example 1 in the next section). In fact, as explained below, none of the existing formulations for the $VRPTW$ can be used with the classical approach for static robust programming.

The classical approach for static robust programming under polyhedral uncertainty relies on dualizing the constraints that contain uncertain parameters [?]. Hence, the approach requires that the uncertainty parameters appear explicitly in the constraints of the problem. This is not the case of the formulations based, respectively, on path inequalities and path variables. In each of these formulations, the uncertain parameter appear implicitly in the paths that define the inequalities and the variables, respectively. Moreover, other formulations proposed in the literature present one (or both) of the two problems: some of their variables are somehow related to the values taken by travel times (non-static robust programming) or the travel times do not appear explicitly in the constraints of the formulations.

For this reason, we propose an extended formulation for the $VRPTW$ that is suitable for robust programming. This formulation is based on two ideas: i) considering a layered representation of the routes followed by the vehicles (see [12], among others), and ii) rewriting the time windows in an extended form [7]. We apply the dualization technique to this formulation to provide a formulation for the $\mathcal{T}$-$VRPTW$. We assess our formulation numerically on instances modeling a problem that arises in maritime transportation.

This paper is structured as follows. Next section presents our extended formulation for the $VRPTW$, denoted by (LF). The dualization technique is recalled in Section 3.1 and applied to formulation (LF) in Section 3.2. Our numerical experiments are described in Section 4 while some concluding remarks are given in Section 5.

2 Extended Formulation for the *VRPTW*

We are given a directed graph $G = (N, A)$, a set of vehicles K, a cost function $c : A \times K \to \mathbb{R}_+$, and a time function $t : A \times K \to \mathbb{R}_+$ for traveling along the arcs of G. The graph contains special depot nodes o and d connected to all other nodes of G, and we denote by N^* the set of nodes that are not depots, $N^* := N \backslash \{o, d\}$. We are given time windows $[a_i, b_i]$ with $a_i, b_i \in \mathbb{R}$, for each $i \in N^*$. The $VRPTW$ consists of defining routes for the vehicles in K such that the union of all routes passes exactly once by each $i \in N^*$. When $|K| = 1$, the problem contains a unique vehicle and reduces to the Asymmetric Traveling Salesman with Time Windows, see [1].

We first recall the classical resource inequality formulation for the problem and show through an example why it cannot be extended to the (static) robust context. The formulation uses a set of binary flow variables x_{ij}^k which indicates whether vehicle k travels from node $i \in N$ to node $j \in N$, and a set of continuous variables y_i^k indicating the arrival time of vehicle k at node $i \in N$. Then, the satisfaction of the time windows is expressed by the following set of constraints

$$x_{ij}^k(y_i^k + t_{ij}^k - y_j^k) \leq 0, \qquad (i,j) \in A,\ k \in K, \tag{1}$$

$$a_i \leq y_i \leq b_i, \qquad i \in N^*, \tag{2}$$

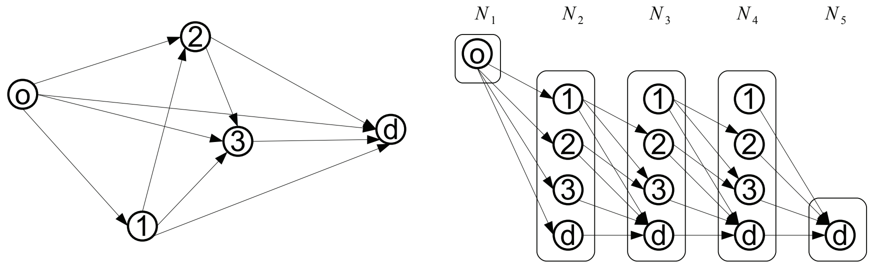
where (1) is linearized using classical "big-M" techniques. Extending the formulation to the robust context would require that inequalities (1) and (2) be satisfied for all values of t in the uncertainty polytope $\mathcal{T}$. We show in Example 1 that this does not work.

Example 1. Consider $N^* = \{1,2,3,4\}$ and time windows $[1,2],[3,4]$ and $[5,6]$ for, respectively, nodes 1,2 and 3. Suppose that the uncertainty polytope $\mathcal{T}$ is defined as follows: $\mathcal{T} := \{(t_{12}, t_{23}) = (1-\lambda)(3,2) + \lambda(2,3),\ 0 \leq \lambda \leq 1\}$. Now, consider a path $p := o \to 1 \to 2 \to 3 \to d$. It is easy to see that the time windows are feasible along p for all $t \in \mathcal{T}$. Consider now the binary vector $\overline{x}$ such that $\overline{x}_{o1} = \overline{x}_{12} = \overline{x}_{23} = \overline{x}_{3d} = 1$. Constraints (1) become $y_2 \geq y_1 + t_{12}$ and $y_3 \geq y_2 + t_{23}$. Because each of these constraints must be satisfied for all $t \in \mathcal{T}$, they become $y_2 \geq y_1 + 3$ and $y_3 \geq y_2 + 3$. Because the smallest feasible value for y_1 is 1, the smallest feasible value for y_2 is 4 and it is impossible to find a value for y_3 that satisfies (2) for node 3.

The aim of this section is to provide a formulation for the $VRPTW$ that is easily adaptable to (static) robust programming. Hence, the formulation satisfies two properties: all variables are related to the routes taken by the vehicles (to avoid situations as in Example 1), and travel times appear explicitly in the constraints. The formulation is based on the rewriting of the time windows constraints (1) and (2) as performed in [7]. Consider path $p = i_0 \to \ldots \to i_n$ for vehicle k and a binary vector $\overline{x} \in \{0,1\}^{|A|}$ that describes p, that is, $\overline{x}_{ij} = 1$ for each $(i,j) \in p$ and $\overline{x}_{ij} = 0$ otherwise. The authors of [7] show that (1) and (2) are satisfied along p if and only if the constraints

$$a_{i_{l_1}} + \sum_{l=l_1,\ldots,l_2-1} t_{i_l i_{l+1}}^k \leq b_{i_{l_2}}, \qquad 0 \leq l_1 < l_2 \leq n \tag{3}$$

are satisfied. To take advantage of constraints (3), we construct a layered graph which keeps track explicitly of the position of each arc along its path. Layered graphs have been used for many network design problems, starting from Gouveia [12], and have already been applied to the VRP (see [10], among others). However, to the best of our knowledge, layered graphs in the sense proposed by [12] have not yet been applied to the $VRPTW$. The main idea of the formulation below is to model the flow problem associated to each vehicle with a directed graph composed of $L = |N|$ layers as illustrated in Figure 1. Namely, from the original graph $G = (N, A)$, we create a directed layered graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ for each vehicle, where $\mathcal{N} := N_1 \cup \ldots \cup N_L$ with $N_1 := \{o\}$, $N_L := \{d\}$ and $N_l := N \backslash \{o\}$, $l = 2, \ldots, L-1$. Let i_l be the copy of $i \in N$ in the *l-th* layer of graph $\mathcal{G}$. Then, the arc sets are defined by $\mathcal{A} := \{(i,j,l) \mid (i,j) \in A, i_l \in N_l, j_{l+1} \in N_{l+1}, l \in \{1, \ldots, L-1\}\} \cup \{(d,d,l), l \in \{2, \ldots, L-1\}\}$, see Figure 1. Hence, (i,j,l) denotes



(a) (b)

Fig. 1. Basic Network for commodity k (a) and its Layered Representation (b).

an arc between $i_l \in N_l$ and $j_{l+1} \in N_{l+1}$. Note that each path between o and d in the layered graph $\mathcal{G}$ is composed of exactly $L-1$ arcs, that corresponds to a path of less than or equal to $L-1$ arcs in G.

Then, we introduce the set of additional binary flow-position variables z_{ij}^{kl} for each $k \in K$, $(i,j,l) \in \mathcal{A}$ defined as follows: $z_{ij}^{kl} = 1$ whenever vehicle $k \in K$ services $i \in N$ exactly in position l of its path from the artificial origin node o to the artificial destination node d and just before servicing node $j \in N$. The extended arc-flow model for the deterministic case, denoted by (LF), follows.

$$\min \quad \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k \tag{4}$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{j \in N:(i,j) \in A} x_{ij}^k = 1, \qquad i \in N^*, \tag{5}$$

$$\sum_{j:(j,i,l-1) \in \mathcal{A}} z_{ji}^{kl-1} - \sum_{j:(i,j,l) \in \mathcal{A}} z_{ij}^{kl} = \begin{cases} -1 \; if & (i = o) \\ 1 \quad if & (i = d \; and \; l = L) \\ 0 \quad else & \end{cases}, \qquad 1 \le l \le L,\ i \in N_l,\ k \in K, \tag{6}$$

$$\sum_{l:(i,j,l) \in \mathcal{A}} z_{ij}^{kl} = x_{ij}^k, \qquad (i,j) \in A,\ k \in K, \tag{7}$$

$$\sum_{(i,j) \in A:(i,j,l_1) \in \mathcal{A}} a_i z_{ij}^{kl_1} + \sum_{l = l_1, \ldots, l_2 - 1} \sum_{(i,j) \in A:(i,j,l) \in \mathcal{A}} t_{ij}^k z_{ij}^{kl} \le \sum_{(i,j) \in A:(i,j,l_2) \in \mathcal{A}} b_j z_{ij}^{kl_2}, \qquad 1 \le l_1 < l_2 < L,\ k \in K, \tag{8}$$

$$x_{ij}^k \in \{0,1\}, \qquad (i,j) \in A,\ k \in K, \tag{9}$$

$$z_{ij}^{kl} \in \{0,1\}, \qquad (i,j,l) \in \mathcal{A},\ k \in K. \tag{10}$$

The objective function (4) minimizes the cost of operating the set of vehicles. Constraints (5) ensure that all $i \in N^*$ are serviced exactly once. Equations (6) are flow balance constraints in the directed layered graph. Variables x and z

are linked by constraints (7). Finally, constraints (8) adapt constraints (3) when path p is not yet decided and depends on variables z.

3 Robust Formulation

3.1 Dualization Approach

In this work, we consider that the travel times are uncertain and belong to a polytope $\mathcal{T}$. This makes the problem a robust program, a class of optimization problems that has witnessed a tremendous attention in the recent years. Conducting an exhaustive literature review of robust programming is beyond the scope of this paper and we redirect the interested reader to [3] and [5]. We recall below the well-known dualization technique for linear robust programs under polyhedral uncertainty, introduced by [4]. Consider the following linear program in $\{0,1\}-$variables

$$
(P) \qquad \begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Bx \le b, & (11) \\ & Tx \le d, & (12) \\ & x \in \{0,1\}^n, \end{aligned}
$$

with $c \in \mathbb{R}^n, b \in \mathbb{R}^r, d \in \mathbb{R}^s, B \in \mathbb{R}^{rn}$, and $T \in \mathbb{R}^{sn}$. Suppose that the problem is subject to uncertainty in the sense that matrix T belongs to a polytope $\mathcal{T} \subset \mathbb{R}^{sn}$. The robust counterpart of (P) is

$$
(\mathcal{T}\text{-}P) \qquad \begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Bx \le b, \\ & Tx \le d, \qquad T \in \mathcal{T}, & (13) \\ & x \in \{0,1\}^n, \end{aligned}
$$

where the s linear constraints in (12) must now be satisfied for each value of $T \in \mathcal{T}$. Hence, the finite set of constraints (12) has been replaced by the infinite set of constraints (13).

The method explained next works in two steps. The first step amounts to realize [3, Section 1.2.1] that x satisfies constrains (13) if and only if it satisfies the following constraints

$$
T_i x \le d_i, \qquad T_i \in \mathcal{T}_i, i = 1, \dots, s, \tag{14}
$$

where $\mathcal{T}_i \subset \mathbb{R}^n$ is the projection of $\mathcal{T}$ into the space corresponding to the coefficients of the i-th row of (13), for each $i = 1, \dots, s$. Said differently, the whole set of constraints $Tx \le d$ is satisfied for each value of the uncertain matrix $T \in \mathcal{T}$ if and only if each constraint $T_i x \le d_i$ is satisfied for each value of the uncertain vector $T_i \in \mathcal{T}_i$.

For the second step, we need to describe more precisely the uncertainty polytope and its projections, which we suppose non-empty. Let the projections of $\mathcal{T}$

be defined as $\mathcal{T}_i = \{T_i \in \mathbb{R}^n : A_i T_i \leq a_i, T_i \geq 0\}$ where matrices A_i, $i = 1, \ldots, s$ and vectors a_i, $i = 1, \ldots, s$ have appropriate dimensions and, of course, depend on the definition of the uncertainty polytope $\mathcal{T}$. Then, the (infinite) constraint set associated to each row i of (14) can be rewritten as $\max_{T_i \in \mathcal{T}_i} T_i x \leq d_i$. The optimization problem of the left-hand side is equivalent to

$$\begin{aligned} \max\ & T_i x \\ \text{s.t.}\ & A_i T_i \leq a_i \\ & T_i \geq 0. \end{aligned} \tag{15}$$

Because (15) is always bounded and feasible, linear programming duality ensures us that its optimal solution is equal to the optimal solution of its dual:

$$\begin{aligned} \min\ & a_i u_i \\ \text{s.t.}\ & (A_i)^T u_i \geq x \\ & u_i \geq 0. \end{aligned}$$

Thus, each constraint i of inequality set (14) is equivalent to

$$\begin{aligned} a_i u_i &\leq d_i \\ (A_i)^T u_i &\geq x \\ u_i &\geq 0, \end{aligned} \tag{16}$$

that is, the infinite number of constraints (13) is replaced by a finite number of constraints and variables. Moreover, the numbers of new constraints and variables are equal to the dimensions of A_i. The dualization technique described above has been applied to numerous robust linear programs subject to polyhedral uncertainty in the literature, see [6], among others.

3.2 Formulation for $\mathcal{T}$-$VRPTW$

In this section, we apply the methodology recalled in Section 3.1 to the $VRPTW$. We consider the budget uncertainty polytope studied by Bertsimas and Sim [6]: we suppose that each component t^k_{ij} of t lies between its mean value $\bar{t}^k_{ij}$ and its peak value $\bar{t}^k_{ij} + \hat{t}^k_{ij}$ and that, for each $k \in K$, at most Γ of them can reach their peak values simultaneously. Formally, this is written as $\mathcal{T}_\Gamma = \times_{k \in K} \mathcal{T}^k_\Gamma$ where all vectors in $\mathcal{T}^k_\Gamma$ are of the form $t^k := \bar{t}^k + \delta^k \hat{t}^k$ and δ^k satisfies the following constraints:

$$\sum_{(i,j) \in A} \delta^k_{ij} \leq \Gamma, \tag{17}$$

$$0 \leq \delta^k_{ij} \leq 1, \quad (i,j) \in A. \tag{18}$$

The robust version of the problem is obtained by replacing (8) with

$$\sum_{(i,j) \in A : (i,j,l_1) \in \mathcal{A}} a_i z^{kl_1}_{ij} + \sum_{l = l_1, \ldots, l_2 - 1} \sum_{(i,j) \in A} t^k_{ij} z^{kl}_{ij} \leq \sum_{(i,j) \in A : (i,j,l_2) \in \mathcal{A}} b_j z^{kl_2}_{ij}, \quad 1 \leq l_1 < l_2 < L,\ k \in K,\ t^k \in \mathcal{T}^k_\Gamma. \tag{19}$$

All variables in (LF) are first-stage variables since they describe the paths taken by the vehicles. Moreover, travel times only appear in (19). Since (19) must be satisfied for all $t^k \in \mathcal{T}_\Gamma^k$, it is convenient to rewrite these constraints as

$$\sum_{(i,j)\in A:(i,j,l_1)\in\mathcal{A}} a_i z_{ij}^{kl_1} + \max_{t^k\in\mathcal{T}_\Gamma^k} \sum_{(i,j)\in A} t_{ij}^k \sum_{l=l_1,\ldots,l_2-1} z_{ij}^{kl} \le \sum_{(i,j)\in A:(i,j,l_2)\in\mathcal{A}} b_j z_{ij}^{kl_2}, \quad 1\le l_1 < l_2 < L,\ k\in K. \tag{20}$$

Let us introduce a dual variable $v^{kl_1l_2}$ and $u_{ij}^{kl_1l_2}$ for each constraint in (17) and (18), respectively, associated to each constraint of (20). Dualizing the maximization problem in (20) as in (15)–(16) yield the following set of constraints:

$$\sum_{(i,j)\in A:(i,j,l_1)\in\mathcal{A}} a_i z_{ij}^{kl_1} + \sum_{(i,j)\in A} \bar{t}_{ij}^k \sum_{l=l_1,\ldots,l_2-1} z_{ij}^{kl} + \Gamma v^{kl_1l_2} + \sum_{(i,j)\in A} u_{ij}^{kl_1l_2} \le \sum_{(i,j)\in A:(i,j,l_2)\in\mathcal{A}} b_j z_{ij}^{kl_2}, \quad 1\le l_1 < l_2 < L,\ k\in K, \tag{21}$$

$$v^{kl_1l_2} + u_{ij}^{kl_1l_2} \ge \hat{t}_{ij}^k \sum_{l=l_1,\ldots,l_2-1} z_{ij}^{kl}, \quad (i,j)\in A,\ 1\le l_1<l_2<L,\ k\in K, \tag{22}$$

$$v^{kl_1l_2} \ge 0, \quad 1\le l_1<l_2<L,\ k\in K \tag{23}$$

$$u_{ij}^{kl_1l_2} \ge 0, \quad 1\le l_1<l_2<L,\ k\in K,\ (i,j)\in A, \tag{24}$$

so that a robust version of (LF) can be formulated as follows:

$$(\mathcal{T}\text{-}LF) \quad \begin{array}{lll} \min & \sum_{k\in K}\sum_{(i,j)\in A} c_{ij}x_{ij}^k & \\ \text{s.t.} & (5),(6),(7),(21)-(24), & \\ & x_{ij}^k \in \{0,1\}, & (i,j)\in A,\ k\in K, \\ & z_{ij}^{kl} \in \{0,1\}, & (i,j,l)\in\mathcal{A},\ k\in K. \end{array}$$

4 Computational Experiments

4.1 Application to the ship routing and scheduling problem

In this section, we apply our model to the Ship Routing and Scheduling problem with Time Windows and uncertain travel times. The deterministic version of this problem is described in [8] where an integer linear programming formulation is presented. Maritime transportation is the major component in international trade and a key part of many economic sectors. Freight transport volumes in maritime systems have been growing for many years and continues to show an upward trend. A great variety of optimization problems is involved in the improvement of maritime transport systems, which makes maritime transportation a challenging research area. Indeed, there has been an increasing research interest in maritime transportation problems over the last years. We refer to [8] for a discussion of practical and theoretical aspects of maritime transportation operations including the description of prescriptive mathematical models and solution approaches.

Time is a crucial factor in maritime transportation and deliveries must occur during predetermined intervals. This makes the maritime transportation problem a special case of the vehicle routing problem with time windows. However, while it can be acceptable to use estimations for travel and service time for some instances of $VRPTW$, this is not the case in maritime transportation. Delays are important and schedules must account for them.

We consider a heterogeneous fleet of ships with specific ship characteristics including different cost structures and load capacities. We assume that a ship is loaded to its capacity in a loading port and the cargo is transported directly to its unloading port. Only one cargo is transported at a time and the cargo size is less than or equal to the capacity of the ship. The fleet has sufficient capacity to serve all committed cargoes during the planning horizon. The corresponding loading and unloading ports are known. Time windows are imposed for loading cargoes. Herein, the service time of a cargo is the time from the arrival at its loading port until the time of departure from its unloading port. Ships are charged port and channel tolls when visiting ports and passing channels, and these costs depend on the size of the ship. The remaining variable sailing costs consist mainly of fuel and oil costs, and depend usually on the ship size.

4.2 Instance Description

The instances used in this paper have been created with a random instance generator made as realistic as possible. The instance generator is based on a real distance matrix that contains 56 ports from around the world, with actual sailing distances between each pair of ports.

Two non-overlapping subsets of ports are selected as pickup ports and delivery ports respectively, to represent the structure of a company operating within deep sea industrial shipping. Cargoes requests are generated between two ports based on a simple inventory model for the delivery port. Time windows are associated with each cargo based on when the request would be generated and an acceptable time before the delivery should be made.

The instance generator also specifies the possible delay in sailing time for each arc in the network. This delay is calculated based on the time normally required to perform the transportation represented by the arc. Since the planning horizon is long, there is a significant risk of a ship being delayed at some point during its route, but the probability of experiencing longer travel times for all legs would be small. Hence it makes sense to make routes that can handle some delays, with Γ equal to some small number.

In the computational testing we generate five instances for each combination of values for the number of cargoes and number of ships.

4.3 Reducing the number of layers

Some vehicle routing problems [10] consider that there exists a constraint on the number of nodes that any route can follow. These problems can be modeled through layered formulations such as the one used in this paper with the difference that the number of layers is a small integer L (part of the problem input), instead of being the total number of nodes of the graph.

The $VRPTW$ studied in this paper does not present this additional constraint: any vehicle can visit an arbitrary number of nodes, as long as time windows are satisfied. Hence, we use a layered formulation in this paper in a different purpose: our aim herein is to present a formulation that is suitable for the robust programming dualization approach. The drawback of this approach is that the number of layers used in the formulation is equal to the number of nodes in the graph, which yields very large number of variables and constraints.

We cannot simply reduce the number of layers because this may cut-off the optimal solution to the problem. However, due to the presence of time windows, the vehicles are usually not able to visit all nodes of the graph. With this idea in mind, we apply a pre-processing to the problem which, for each vehicle, computes the longest path that satisfies the time windows. The pre-processing step was solved by an integer programming formulation based on the MTZ-inequalities [16].

4.4 Numerical Results

We present in this section computational results for formulations (LF) and ($\mathcal{T}$-LF) on instances with 10 and 20 cargoes and a number of ships varying between 1 and 5. The formulations have been coded using the modeling language Xpress Mosel 3.2.3 and solved by Xpress Optimizer 22.01.09 [19]. The experiments were run on computer equipped with a processor Intel Core i5 at 2.53 GHz and 4 GB of RAM memory.

We present in Table 1 the average longest path computed for the different values of $|N|$ and $|K|$. We see that the reduction increases with the number of ships. This was expected because the instances are generated in such a way that all ships are necessary. Hence, more ships lead to time windows harder to satisfy and smaller feasible paths. For problems with one ship ($|K| = 1$), the pre-processing has no effect since the ship must visit all nodes of the graph.

$\|N\|\backslash\|K\|$	2	3	4	5
10	6.6	6.8	6.1	4.96
20	14.8	12.7	10.7	9.96

Table 1. Average longest path.

Tables 2 and 3 present the average solution times for instances with 10 and 20 cargoes, respectively. Column *Reduction* provides the average solution times necessary to compute the longest paths for each vehicle, while columns below *With reduction* and *Without reduction* provide the average solution times necessary to solve the problem to optimality, respectively with and without reducing the number of layers using the pre-processing. A time limit of 3600 seconds has been set. Then, unsolved instances within this limit are written in parentheses and the average solution times are computed without them. We see from Tables 2 and 3 that the pre-processing allows to reduce significantly the solution times, especially for instances with 20 nodes. In particular, the 7 instances that could not be solved within the time limit can be solved after the reduction.

	reduction	With reduction			Without reduction		
$\|K\|\backslash\Gamma$		0	1	2	0	1	2
1	–	–	–	–	0.0746	0.0798	0.0846
2	0.168	0.319	0.162	0.169	0.299	0.25	0.192
3	0.615	0.19	0.847	0.769	0.388	0.976	0.907
4	1.2	0.182	0.426	0.557	0.38	0.8	0.914
5	1.34	0.104	0.312	0.326	0.466	0.658	0.681

Table 2. Results for instances with 10 cargoes.

	reduction	With reduction					Without reduction				
$\|K\|\backslash\Gamma$		0	1	2	4	6	0	1	2	4	6
1	–	–	–	–	–	–	7.16	5.13	8.56	7.58	9.95
2	6.09	18.8	54.4	140	134	192	22.3	174	269	528	536
3	8.59	44.4	332	908	1990	592	73.2	265(**1**)	307(**1**)	609(**1**)	741
4	81.9	13.2	89.1	160	237	253	101	760	828(**1**)	1161(**1**)	857(**1**)
5	334	13.6	52.8	67	188	153	57.9	218	323	193(**1**)	1487

Table 3. Results for instances with 20 cargoes.

5 Conclusion

In this paper, we present the first robust formulation for the vehicle routing problem with time windows and uncertain travel times. To this aim, we introduce a new layered formulation for the vehicle routing problem with time windows that enables us to apply the classical static robust programming approach. We test this approach on instances that describe a maritime transportation problem. We see that this methodology can solve instances with up to 20 nodes. In order to solve larger instances, future work will address different approaches for the problem that rely on the use of alternative robust programming techniques, such as adjustable robust programming.

References

1. Ascheuer, N., Fischetti, M., Grötschel, M.: A polyhedral study of the asymmetric traveling salesman problem with time windows. Networks 36(2), 69–79 (2000)
2. Bard, J.F., Kontoravdis, G., Yu, G.: A branch-and-cut procedure for the vehicle routing problem with time windows. Transportation Science 36, 250–269 (May 2002)
3. Ben-Tal, A., Ghaoui, L.E., Nemirovski, A.: Robust optimization. Princeton University Press (2009)
4. Ben-Tal, A., Nemirovski, A.: Robust solutions of uncertain linear programs. Operations Research Letters 25, 1–13 (1999)
5. Bertsimas, D., Brown, D., Caramanis, C.: Theory and applications of robust optimization. SIAM Review 53, 464–501 (2011)
6. Bertsimas, D., Sim, M.: The price of robustness. Operations Research 52, 35–53 (2004)
7. Chardy, M., Klopfenstein, O.: Handling uncertainties in vehicle routing problems through data preprocessing. Transportation Research Part E: Logistics and Transportation Review 48(3), 667 – 683 (2012)
8. Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D.: Handbooks in Operations Research and Management Science, vol. 14, chap. Maritime Transportation, pp. 150–168. Elsevier Science Publishers, North-Holland (2007)
9. Desaulniers, G., Desrosiers, J., Spoorendonk, S.: The Vehicle Routing Problem with Time Windows: State-of-the-Art Exact Solution Methods. John Wiley & Sons, Inc. (2010)
10. Godinho, M.T., Gouveia, L., Magnanti, T.L.: Combined route capacity and route length models for unit demand vehicle routing problems. Discrete Optimization 5(2), 350 – 372 (2008)
11. Golden, B., Raghavan, S., Wasil, E.A.: The vehicle routing problem : latest advances and new challenges. Operations research/Computer science interfaces series, 43, Springer (2008)
12. Gouveia, L.: Using variable redefinition for computing lower bounds for minimum spanning and steiner trees with hop constraints. INFORMS J. on Computing 10, 180–188 (1998)
13. Kallehauge, B.: Formulations and exact algorithms for the vehicle routing problem with time windows. Computers & OR 35(7), 2307–2330 (2008)
14. Kallehauge, B., Boland, N., Madsen, O.B.G.: Path inequalities for the vehicle routing problem with time windows. Networks 49(4), 273–293 (2007)
15. Kallehauge, B., Larsen, J., Madsen, O.B.G., Solomon, M.M.: Column Generation, chap. Vehicle routing problem with time window. Springer (2005)
16. Miller, C.E., Tucker, A.W., Zemlin, R.A.: Integer programming formulation of traveling salesman problems. J. ACM 7, 326–329 (1960)
17. Ordónez, F.: Robust vehicle routing. TUTORIALS in Oprations Research pp. 153–178 (2010)
18. Soyster, A.L.: Convex programming with set-inclusive constraints and applications to inexact linear programming. Operations Research 21, 1154–1157 (1973)
19. Suite, F.X.O.: Xpress-Optimizer reference manual. Tech. Rep. Release 22.01 (2011)
20. Sungur, I., nez, F.O., Dessouky, M.: A robust optimization approach for the capacitated vehicle routing. IIE Transactions 40(5), 509–523 (2008)