HAL
open science

Robust constrained shortest path problems under budgeted uncertainty

Artur Alves Pessoa, Luigi Di Puglia Pugliese, Francesca Guerriero, Michael Poss

To cite this version:

Artur Alves Pessoa, Luigi Di Puglia Pugliese, Francesca Guerriero, Michael Poss. Robust constrained shortest path problems under budgeted uncertainty. Networks, 2015, 66 (2), pp.98-111. 10.1002/net.21615 . hal-01229790

HAL Id: hal-01229790

https://hal.science/hal-01229790v1

Submitted on 24 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Robust constrained shortest path problems under budgeted uncertainty

Artur Alves Pessoa*[1], Luigi Di Puglia Pugliese†[2], Francesca Guerriero‡[2] and Michael Poss§[3]

[1]Production Engineering Department, Fluminense Federal University, Rua Passo da Pátria 156, 24210-240 Niteroí, RJ, Brazil
[2]Department of Mechanical, Energy and Management Engineering, University of Calabria, Rende, Italy
[3]UMR CNRS 7253 Heudiasyc, Université de Technologie de Compiègne, Centre de Recherches de Royallieu, 60200 Compiègne, France

Abstract

We study the robust constrained shortest path problem under resource uncertainty. After proving that the problem is $\mathcal{NP}$-hard in the strong sense for arbitrary uncertainty sets, we focus on budgeted uncertainty sets introduced by Bertsimas and Sim (2003) and their extension to variable uncertainty by Poss (2013). We apply classical techniques to show that the problem with capacity constraint can be solved in pseudo-polynomial time. However, we prove that the problem with time windows is $\mathcal{NP}$-hard in the strong sense when Γ is not fixed, using a reduction from the independent set problem. We introduce then new robust labels that yield dynamic programming algorithms for the problems with time windows and capacity constraints. The running times of these algorithms are pseudo-polynomial when Γ is fixed, exponential otherwise. We present numerical results for the problem with time windows which show the effectiveness of the label-setting algorithm based on the new robust labels. Our numerical results also highlight the reduction in price of robustness obtained when using variable budgeted uncertainty instead of classical budgeted uncertainty.

Keywords: Constrained shortest path, Robust optimization, Budgeted uncertainty, Dynamic programming, NP-hard.

1 Introduction

Let $G = (N, A)$ be a directed graph with two special nodes o and d, $\kappa : A \to \mathbb{R}_+$ be a cost function and $r : A \to \mathbb{Z}_+$ be a weight function that represents the resource consumption along each arc of G. The resource constrained shortest path problem looks for the shortest path (according to cost κ) from o to d that satisfies the resource constraint(s), which can be, for instance, capacity constraints or time windows. Namely, a capacity constraint considers an upper bound $W \in \mathbb{Z}_+$ on the resource available on a path, and imposes that any feasible path p satisfies

$$w(p) \leq W, \tag{1}$$

where $w(p) = \sum_{a \in p} r_a$. The shortest path problem with capacity constraint is denoted by CSP. Differently from the capacity constraint, time windows must be satisfied at each node pertaining to

*artur@producao.uff.br
†luigi.dipugliapugliese@unical.it
‡francesca.guerriero@unical.it
§michael.poss@hds.utc.fr, *Corresponding author*

feasible paths and they are defined by two vectors $\underline{b}$ and $\overline{b}$ in $\mathbb{Z}_+^{|N|}$ such that $\underline{b}_i \leq \overline{b}_i$ for each $i \in N$. Consider a path $p = (i_0, i_1, \ldots, i_l)$ containing $l = |p|$ arcs. For each $j \in \{1, \ldots, l\}$, we define the arrival time at node i_j as $t_j(p) = \max(\underline{b}_{i_j}, t_{j-1}(p) + r_{i_{j-1}i_j})$, and the time window constraints are written as

$$t_j(p) \leq \overline{b}_{i_j}, \quad j = 1, \ldots, l. \tag{2}$$

The shortest path problem with time windows is denoted by $TWSP$. One can define similarly problems with multiple capacity constraints and/or time window constraints. To keep simple notations, we disregard these generalizations in the following and consider a unique resource constraint. One can readily extend the algorithms presented in this paper to problems that contain multiple resource constraints. The literature on constrained shortest path problems is very broad. Providing an exhaustive account of the papers published on the topic would be beyond the scope of this work, and we rather redirect the interested reader to the surveys [17, 11] and the references therein.

Constrained shortest paths problem are $\mathcal{NP}$-hard [14], however, in the weak sense since pseudo-polynomial algorithms exist. Indeed, early papers propose dynamic programming algorithms for these problems (e.g. [19]), based on states defined by the current node, the resource consumption, and the cost consumption. An important dynamic programming algorithm that is often regarded as one of the most efficient (e.g. [18]) is the label-setting algorithm from [10]. Denoting $B = \max_{i \in N}(\overline{b}_i - \underline{b}_i + 1)$ and using appropriate data structure, [10] show that the worst-case running time of the label-setting algorithm for solving $TWSP$ is $O(|A|B)$. Similarly, the worst-case running time of the algorithm applied to CSP is $O(|A|W)$. When costs and/or resource consumptions are allowed to take negative values, the optimal solutions may contain cycles. Forbidding such cycles leads to the problem called *Elementary Resource Constrained Shortest Path Problem* ($ERCSP$), $\mathcal{NP}$-hard in the strong sense [12]. In this work, we focus on problems with positive costs and resource consumptions and avoid the burden of forbidding cycles. Nevertheless, the algorithms proposes herein could be adapted to solve the robust versions of the $ERCSP$.

In this paper, we consider uncertain versions of the above problems and consider that r is not known with precision. Instead, we suppose that the weights are uncertain parameters that can take any value in a given uncertainty set $\mathcal{U} \subset \mathbb{Z}_+^{|A|}$, which we assume to be finite. Given a fixed $r \in \mathcal{U}$, we redefine the values of $w(p)$ and $t_j(p)$ as $w(p, r)$ and $t_j(p, r)$, respectively, to mark the dependency on uncertain parameter r. In the uncertain context, we impose that constraints (1) and (2) be satisfied for all $r \in \mathcal{U}$, that is

$$w(p, r) \leq W, \quad r \in \mathcal{U}, \tag{3}$$

and

$$t_j(p, r) \leq \overline{b}_{i_j}, \quad j = 1, \ldots, l, \ r \in \mathcal{U}. \tag{4}$$

We denote the associated optimization problems by $\mathcal{U}$-CSP and $\mathcal{U}$-$TWSP$, respectively.

Remark 1. *The recent literature on robust optimization [5] very often considers* convex *uncertainty sets, such as polytopes, which may seem different from our assumption that $\mathcal{U}$ is finite. Nevertheless, one readily sees that, because $w(p, r)$ and $t(p, r)$ are convex functions of r, it is equivalent to consider $\mathcal{U}$ or its convex hull* $\mathrm{conv}(\mathcal{U})$. *Namely, constraints* (3) *and* (4) *are satisfied for all $r \in \mathcal{U}$ if and only if they are satisfied for all $r \in \mathrm{conv}(\mathcal{U})$.*

By enforcing resource constraints for all $r \in \mathcal{U}$, our approach falls into the framework of robust optimization, a celebrated approach to handle optimization problems in the presence of uncertainty [5, 20]. In [6, 7], Bertsimas and Sim study budgeted uncertainty sets, denoted by $\mathcal{U}^\Gamma$ herein, that control with positive number Γ the number of uncertain parameters that can deviate simultaneously from their means to their peak values. The nice properties enjoyed by set $\mathcal{U}^\Gamma$ have led to a very large use of budgeted uncertainty for discrete optimization problems under uncertainty. Two important such properties are related to the computational complexity and probabilistic guarantee offered by the robust model.

Complexity The robust problems very often belong to the same complexity class as their deterministic counterpart. This is true, for instance, for robust combinatorial optimization problems

under cost uncertainty. In fact, prior to this paper, no one had came up with a robust combinatorial optimization problem under budgeted uncertainty that does not belong to the complexity class of its deterministic counterpart.

Probabilistic guarantee If the uncertain parameters of an optimization problem are random variables, the constraints of the problem become probabilistic constraints. Under some conditions, any solution to the associated robust optimization problem under uncertainty set $\mathcal{U}^\Gamma$ satisfies the probabilistic constraints with a guarantee that depends on Γ (see also [7]).

Recent works have seek to improve these properties. For instance, [4] and [15] study how to generalize the complexity results of [6] to combinatorial optimization problems with uncertainty in the constraints and integer variables, respectively, while [21, 22] discuss how the probabilistic guarantee can be improved by allowing Γ to depend on the decision variables, yielding variable budgeted uncertainty, denoted $\mathcal{U}^\gamma$ in what follows.

While the robust shortest path problem has been investigated in several works (see [3, 6, 9], among others), we are not aware of previous works on robust approaches for the constrained shortest path problem. The closely related vehicle routing problem under uncertainty has, on the contrary, been the subject of several studies. The problem with a robust capacity constraint has been first investigated by Sungur et al. in [23] who study conditions under which robust versions of the capacitated vehicle routing problem (CVRP) can be solved through methods similar to the ones used for the deterministic version of the CVRP. More recently, Gounaris et al. present in [16] study more refined robust approaches for the CVRP, including the use of affine decision rules and the extension of rounded capacity inequalities to the robust case. The interest in the robust VRP with time windows is more recent. In [1], the authors present an extended formulation that can be dualized using classical tools of robust optimization [7]. The authors pursue their work in [2] where they present more subtle formulations, based on implicit formulations and adjustable robust optimization, that are tackled through decomposition algorithms.

Contributions and structure of the paper To our knowledge, this paper is the first to study the computational complexity for problems $\mathcal{U}$-CSP and $\mathcal{U}$-$TWSP$. In particular, the main contribution of the paper is two-fold:

1. We prove that $\mathcal{U}^\Gamma$-$TWSP$ is $\mathcal{NP}$-hard in strong sense, contrasting with the pseudo-polynomial algorithms that can be devised for $\mathcal{U}^\Gamma$-CSP.

2. We propose a label-setting algorithm for $\mathcal{U}^\Gamma$-$TWSP$ that is exponential in Γ. Numerical results show the effectiveness of the algorithm.

We show in Section 4 that the problems are $\mathcal{NP}$-hard in the strong sense for uncertainty sets of unbounded cardinalities. In view of this difficulty, we turn then to budgeted uncertainty sets, known to enjoy easier robust counterparts than arbitrary uncertainty sets. We recall the definition of budgeted uncertainty and its recent extension to variable uncertainty in Section 2.2, yielding problems $\mathcal{U}^\Gamma$-CSP and $\mathcal{U}^\Gamma$-$TWSP$ (budgeted uncertainty), and $\mathcal{U}^\gamma$-CSP and $\mathcal{U}^\gamma$-$TWSP$ (variable budgeted uncertainty).

Section 3 is devoted to $\mathcal{U}^\Gamma$-CSP and $\mathcal{U}^\gamma$-CSP. In particular, we define an algorithm based on the solution of $|A| + 1$ deterministic CSP by employing similar techniques to those developed in [6, 22, 4, 15]. We show then in Section 4 that $\mathcal{U}^\Gamma$-$TWSP$ is $\mathcal{NP}$-hard in the strong sense even when G is a directed and acyclic graph, the mean weight vector $\overline{r} = 0$, and $\underline{b} = 0$, which highlights a fundamental difference between $\mathcal{U}^\Gamma$-CSP and $\mathcal{U}^\Gamma$-$TWSP$. Our proof is based on a reduction from the independent set problem. This result exhibits for the first time a combinatorial optimization problem that admits a pseudo-polynomial algorithm whose robust version (with budgeted uncertainty) is $\mathcal{NP}$-hard in the strong sense.

We present in Section 5 a label-setting algorithm based on that proposed in [10]. We pay a particular attention to the generalization of dominance rules. On one hand, the resulting algorithm is pseudo-polynomial when Γ is fixed. On the other hand, it has an exponential worst-case running-time when Γ is part of the input data. The proposed label-setting algorithms are evaluated by considering

instances inspired by the scientific literature. We report in Section 6 a detailed accounting of the numerical results. The goals of our numerical experiments for the $\mathcal{U}^\Gamma$-$TWSP$ are two-fold. First, they assess the effectiveness of the label-setting algorithm based on the defined robust labels for $\mathcal{U}^\Gamma$-$TWSP$ and $\mathcal{U}^\gamma$-$TWSP$. Second, they show that a reduction in the price of robustness can be obtained when using variable budgeted uncertainty instead of the classical budgeted uncertainty. The paper ends with Section 7, where we give conclusions and final remarks.

2 Uncertainty set

2.1 $\mathcal{NP}$-hardness for general uncertainty sets

When $\mathcal{U}$ is reduced to a singleton, $\mathcal{U}$-CSP and $\mathcal{U}$-$TWSP$ reduce to the deterministic counterparts. Otherwise, the problems can be seen as deterministic problems that contain multiples capacity or time windows constraints, one for each value of r in $\mathcal{U}$. When the cardinality of $\mathcal{U}$ is bounded by a small constant, $\mathcal{U}$-CSP and $\mathcal{U}$-$TWSP$ could therefore be solved in pseudo-polynomial time using, for instance, the classical label-setting algorithm. However, the running-time of such algorithms is exponential in the cardinality of $\mathcal{U}$, making them impractical for solving problems with uncertainty sets of large cardinalities. One can wonder whether different efficient approaches could exist to solve these problems under arbitrary uncertainty sets. The next result answers by the negative, showing that $\mathcal{U}$-CSP and $\mathcal{U}$-$TWSP$ are $\mathcal{NP}$-hard in the strong sense when the cardinality of $\mathcal{U}$ is unbounded. The proof consists in reducing the robust knapsack problem ($\mathcal{U}$-KP) to $\mathcal{U}$-CSP. A similar proof can be devised to reduce $\mathcal{U}$-KP to $\mathcal{U}$-$TWSP$.

ROBUST KNAPSACK PROBLEM ($\mathcal{U}$-KP)

Input: Set $\{1, \ldots, m\}$ of items, set $\mathcal{V} \subset \mathbb{Z}_+^m$ of weights w, profit vector $c \in \mathbb{R}_+^m$, capacity D.

Task: Find a subset of items $I \subseteq \{1, \ldots, m\}$ of maximum profit such that $\sum_{i \in I} w_i \leq D$ for all $w \in \mathcal{V}$.

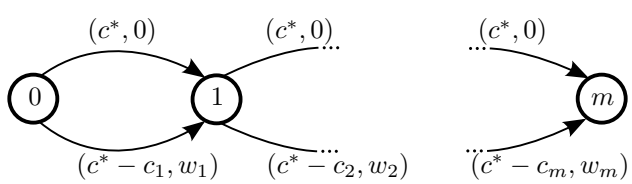

Figure 1: Reduction from the knapsack problem to $\mathcal{U}$-CSP.

Theorem 1. *$\mathcal{U}$-CSP is $\mathcal{NP}$-hard in the strong sense for an uncertainty set $\mathcal{U}$ of unbounded cardinality.*

Proof. Problem $\mathcal{U}$-KP is $\mathcal{NP}$-hard in the strong sense, see Theorem 4.5 from [24]. Given an instance to $\mathcal{U}$-KP, the corresponding instance of $\mathcal{U}$-CSP is built as follows. First, we compute $c^* = \max_{1 \leq i \leq m} c_i$. Then, we consider the directed graph with $m+1$ nodes and $2m$ arcs described in Figure 1. For each pair of nodes $\{i-1, i\}$, let $\overrightarrow{(i-1, i)}$ and $\overrightarrow{(i-1, i)}$ denote the top and bottom arc, respectively. The instance of the $\mathcal{U}$-CSP looks for a path from 0 to m in the graph from Figure 1 with $W = D$, $\kappa_{\overrightarrow{i-1i}} = c^*$ and $r_{\overrightarrow{i-1i}} = 0$ for each $i \in \{1, \ldots, m\}$, $\kappa_{\overrightarrow{i-1i}} = c^* - c_i$ and $r_{\overrightarrow{i-1i}} = w_i$ for each $i \in \{1, \ldots, m\}$. Accordingly, supposing that bottom arcs have smaller indexes than top arcs, $\mathcal{U} = \mathcal{V} \times (0, \ldots, 0)$. Then, for any positive integer K, we see that the optimal solution cost to $\mathcal{U}$-KP is equal to K if and only if the optimal solution cost to $\mathcal{U}$-CSP is equal to $mc^* - K$. $\square$

The above result suggests that problems $\mathcal{U}$-CSP and $\mathcal{U}$-$TWSP$ can be quite hard to solve exactly for arbitrary uncertainty sets. This is not very surprising as it is well known that many robust

versions of combinatorial optimization problems become $\mathcal{NP}$-hard in the presence of uncertainty [20]. In the following, we thus focus on special uncertainty sets, $\mathcal{U}^\Gamma$ and $\mathcal{U}^\gamma$, known to often lead to robust optimization problems with computational complexities similar to the complexities of their deterministic counterparts. These sets are defined formally in the next subsection.

2.2 Budgeted uncertainty and probabilistic guarantee

To avoid the difficulty often present in robust combinatorial optimization (such as Theorem 1), Bertsimas and Sim introduced in [6] a particular type of uncertainty sets, called budgeted uncertainty, whose conservatism can be adjusted through a budget parameter $\Gamma \in \mathbb{R}_+$.

Budgeted uncertainty set $\mathcal{U}^\Gamma$ considers nominal value $\overline{r}_a$ and peak value $\overline{r}_a + \hat{r}_a$ for each $a \in A$. Whenever Γ is integer, each vector $r \in \mathcal{U}^\Gamma$ has up to Γ of its components equal to their peak values while the other components are equal to their nominal values. If Γ is fractional, a vector in $\mathcal{U}^\Gamma$ is formed of up to $\lfloor \Gamma \rfloor$ peak values plus a component a' equal to $\overline{r}_{a'} + (\Gamma - \lfloor \Gamma \rfloor)\hat{r}_{a'}$, the rest of the components being at their nominal values. The set is defined formally as follows: $\mathcal{U}^\Gamma \equiv$

$$\left\{ r \in \mathbb{R}^{|A|} \middle| \exists S, \hat{S} \subseteq A, a' \in A \backslash (S \cup \hat{S}), |\hat{S}| \leq \Gamma, \begin{array}{l} S \cap \hat{S} = \emptyset, \\ S \cup \hat{S} \cup \{a'\} = A \end{array} : r_a = \left[\begin{array}{ll} \overline{r}_a & a \in S \\ \overline{r}_a + \hat{r}_a & a \in \hat{S} \\ \overline{r}_a + (\Gamma - \lfloor \Gamma \rfloor)\hat{r}_a & a = a' \end{array} \right] \right\}. \tag{5}$$

Notice that although we define $\mathcal{U}^\Gamma$ as a finite set, it is often described as a polytope in the literature (see Remark 1). In what follows, we study the complexity of $\mathcal{U}$-CSP and $\mathcal{U}$-$TWSP$ under budgeted uncertainty, yielding problems $\mathcal{U}^\Gamma$-CSP and $\mathcal{U}^\Gamma$-$TWSP$. We include in our study the consideration of variable budgeted uncertainty that improves over the probabilistic guarantee offered by $\mathcal{U}^\Gamma$.

Let $\tilde{r}_a = \overline{r}_a + \eta_a \hat{r}_a$ be the random variable associated with parameter r_a and suppose that η_a, $a \in A$, are arbitrary random variables independently and symmetrically distributed in $[-1, 1]$. Bertsimas and Sim [7] construct different functions $\delta : \mathbb{R}^2 \to \mathbb{R}$ such that any vector x that satisfies the robust constraint $\sum_{a \in A} r_a x_a \leq h$, for all $r \in \mathcal{U}^\Gamma$, also satisfies

$$P\left(\sum_{a \in A} \tilde{r}_a x_a > h \right) \leq \delta(\Gamma, |A|). \tag{6}$$

Hence, given a probability level $\epsilon \in (0, 1)$, choosing Γ such that $\delta(\Gamma, |A|) \leq \epsilon$ ensures that probabilistic constraint $P\left(\sum_{a \in A} r_a x_a > h\right) \leq \epsilon$ be satisfied. This is a very strong result since robust constraints are significantly easier to handle than individual probabilistic constraints. We refer to Appendix A for an example of function δ.

Remark 2. *Bound* (6) *is not directly applicable to time windows inequalities* (2) *because the latter are non-linear. Nevertheless, we can linearize these inequalities by replacing each occurrence of* $\max(X, Y)$ *hidden in notation* $t_j(p)$ *by one inequality with* X *and one inequality with* Y*. Doing so recursively yields a set of linear inequalities equivalent to non-linear inequalities* (2)*. The same transformation can be applied to the probabilistic constraints that involve random vector* $\tilde{r}$*, which yields a set of linear probabilistic constraints, each of which approximated by a robust constraint through bound* (6)*.*

Recently, [21] has noticed that the bound provided by (6) can be improved when x is binary by considering a more general model of uncertainty. We sketch below how this can be done and redirect the interested reader to [21] for details. The author of [21] shows that (6) can be changed to

$$P\left(\sum_{a \in A} \tilde{r}_a x_a > h \right) \leq \delta(\Gamma, |x|), \tag{7}$$

where $|x| = \sum_{a \in A} x_a$ is the cardinality of x. Inequality (7) tells us that the value of Γ can be changed according to the cardinality of x while ensuring the same level of probability protection. This observation led [21] to introduce a new model of uncertainty, called variable budgeted uncertainty, where the fixed value of Γ is replaced by the non-decreasing function $\gamma : \{0, 1, \ldots, |A|\} \to \mathbb{R}_+$. Coming

back to our shortest path problem, variable budgeted uncertainty adjusts the level of conservatism according to the cardinality of the path considered. For a path $p \subseteq A$ from o to d, the point-to-set mapping $\mathcal{U}^\gamma : \mathbb{Z} \rightrightarrows \mathbb{R}^{|A|}$ is defined as $\mathcal{U}^\gamma(|p|) \equiv$

$$\left\{ r \in \mathbb{R}^{|A|} \middle| \exists S, \hat{S} \subseteq A, a' \in A \backslash (S \cup \hat{S}), |\hat{S}| \leq \gamma(|p|), \begin{array}{l} S \cap \hat{S} = \emptyset, \\ S \cup \hat{S} \cup \{a'\} = A \end{array} : r_a = \left[\begin{array}{ll} \bar{r}_a & a \in S \\ \bar{r}_a + \hat{r}_a & a \in \hat{S} \\ \bar{r}_a + (\gamma(|p|) - \lfloor \gamma(|p|) \rfloor)\hat{r}_a & a = a' \end{array} \right] \right\}. \tag{8}$$

When $\gamma(k) = \Gamma$ for each $k \in \{0, 1, \ldots, |A|\}$, $\mathcal{U}^\gamma(x)$ reduces to $\mathcal{U}^\Gamma$. However, it has been shown in [21, 22] that $\mathcal{U}^\gamma$ is in general less conservative than $\mathcal{U}^\Gamma$. A measure of conservatism of a robust model is its price of robustness, that is, the increase in solution cost when imposing a probabilistic protection with a specified guarantee $\epsilon > 0$. The reduction in the price of robustness offered by model $\mathcal{U}^\gamma$ is particularly significant when the cardinality of the optimal solution is small in comparison to the total number of variables of the problem. This is the case for the shortest path problem since an optimal solution typically contains $O(N)$ variables while the number of arcs can be as large as $O(N^2)$.

Remark 3. *In view of the above discussion, one way to choose Γ is to follow probabilistic bound $\delta(\Gamma, |A|)$ used in constraint (6). Unfortunately, this is likely to yield values of Γ that are greater than the cardinality of the optimal solution. We have argued that function γ avoids this problem. However, the resulting optimization problem may be more complex than the one using a constant Γ. An alternative is to refine the choice of Γ by computing an upper bound UB on the cardinality of the longest path from o to d that satisfies the resource constraint. Then, Γ would be defined in accordance with $\delta(\Gamma, UB)$. Notice that computing an exact upper bound may be as difficult as solving the robust constrained shortest path, so that we should restrict ourselves to bounds UB on the longest path that can be computed easily.*

3 Complexity of $\mathcal{U}^\Gamma$-CSP and $\mathcal{U}^\gamma$-CSP

We consider in this section $\mathcal{U}^\gamma$-CSP and show how it can be solved in pseudo-polynomial time. The results presented in what follows are obtained by applying tools similar to those described in [22, Section 3]. However, their importance should not be undervalued since they show that $\mathcal{U}^\gamma$-CSP is pseudo-polynomial, and therefore, belongs to the same complexity class as its deterministic counterpart. In contrast, we show in the next section that $\mathcal{U}^\Gamma$-$TWSP$ is $\mathcal{NP}$-hard in the strong sense.

One of the major result of [6] shows that combinatorial optimization problems with cost uncertainty and uncertainty polytope $\mathcal{U}^\Gamma$ can be addressed by solving $n+1$ deterministic problems (where n is the number of binary variables). The authors of [4, 15] have recently shown how the aforementioned result can be extended to a robust *constraint* under uncertainty polytope $\mathcal{U}^\Gamma$. Combining ideas from [22] and [4, 15], we see easily that combinatorial problems with one robust constraint under uncertainty point-to-set mapping $\mathcal{U}^\gamma$ (assuming γ affine) can be solved by solving $n+1$ deterministic problems.

Theorem 2. *Let $\mathcal{Y} \subseteq \{0,1\}^{|I|}$ be the feasibility set of a combinatorial optimization problem and γ be the affine function defined by reals γ^0 and γ^1 through $\gamma(y) = \gamma^0 + \gamma^1 \sum_{i \in I} y_i$. Moreover, let $\kappa \in \mathbb{R}^{|I|}$ be a cost vector, $b \in \mathbb{R}$, and r be an uncertain vector taking values in $\mathcal{U}^\gamma(y)$. Without loss of generality, suppose that indices are ordered such that $\hat{r}_1 \geq \hat{r}_2 \geq \ldots \geq \hat{r}_{|I|}$ and let $\hat{r}_{|I|+1}$ be 0. Then, the optimal solution to*

$$\left\{ \min \sum_{i \in I} \kappa_i y_i \ : \ \sum_{i \in I} r_i y_i \leq b \quad \forall r \in \mathcal{U}^\gamma(y), \ y \in \mathcal{Y} \right\}$$

is equal to the optimal solution of $\min_{l \in \{1, \ldots, |I|+1\}} Z^l$ where

$$Z^l = \left\{ \min \sum_{i \in I} \kappa_i y_i \ : \ \sum_{i \in I} (\bar{r}_i + \gamma^1 \hat{r}_l) y_i + \sum_{j=1}^{l} (\hat{r}_j - \hat{r}_l) y_j + \leq b - \gamma^0 \hat{r}_l, \ y \in \mathcal{Y} \right\},$$

for $l = 1, \ldots, |I| + 1$.

Proof. The proof follows closely the line of the proof of [4, Lemma 2], replacing Γ by $\gamma(y)$. $\square$

There are two differences between Theorem 2 and [4, Lemma 2]. First, and most importantly, the fact that γ is an affine function rather than a constant leads the presence of terms $\gamma^1 \hat{r}_l y_i$ in the constraints of Z^l. Second, we cannot reduce the number of problems to be solved as done in [4], because γ is in general not integer-valued.

We formulate next $\mathcal{U}^\gamma$-CSP as an integer linear program with one linear robust constraint. Let $x \in \{0,1\}^{|A|}$ be a vector of optimization variables stating which arcs belong to the solution, and let $\mathcal{X} \subset \{0,1\}^{|A|}$ contain all vectors x that correspond to paths from o to d. With these variables, $\mathcal{U}^\gamma$-CSP can be cast as

$$\begin{aligned} \min \quad & \sum_{a \in A} \kappa_a x_a \\ \text{s.t.} \quad & \sum_{a \in A} r_a x_a \leq W, \quad r \in \mathcal{U}^\gamma(x) \\ & x \in \mathcal{X}. \end{aligned} \tag{9}$$

Applying Theorem 2 to problem (9) yields the results below.

Corollary 1. *If γ is affine, $\mathcal{U}^\gamma$-CSP can be handled by solving $|A|+1$ problems CSP with modified weights.*

The above result implies that $\mathcal{U}^\gamma$-CSP can be solved in pseudo-polynomial time. It is interesting to notice that Corollary 1 can be applied both to $\mathcal{U}^\Gamma$-CSP and $\mathcal{U}^\gamma$-CSP and that the resulting computational complexity is the same. This is a surprisingly positive result given that model $\mathcal{U}^\gamma$ relies upon more complex mathematical tools and often leads to difficulties absent with the use of $\mathcal{U}^\Gamma$ (see [21] for details). When the appropriated data structures are used, CSP can be solved by the label-setting algorithm in $O(|A|W)$ if there are no zero weights [10], yielding the following corollary.

Corollary 2. *If γ is affine, $\mathcal{U}^\gamma$-CSP can be solved in $O(|A|^2 W)$.*

Notice that whenever γ is not affine, we can still solve $\mathcal{U}^\gamma$-CSP in pseudo-polynomial time, albeit at a higher computational cost. Let $CSP_k^{\leq}$ and $\mathcal{U}^\Gamma$-$CSP_k^{\leq}$ be defined as CSP and $\mathcal{U}^\Gamma$-CSP, respectively, with the additional restriction that feasible paths cannot contain more than k arcs, for $0 \leq k \leq |N|$, and $\Gamma = \gamma(k)$. Clearly, the label-setting algorithm applied to $CSP_k^{\leq}$ with the appropriate data structure has a running-time of $O(|A|W|N|)$. Moreover, Theorem 2 can be applied to $\mathcal{U}^\Gamma$-$CSP_k^{\leq}$ so that the problem can be solved in $O(|A|^2 W|N|)$.

Theorem 3. *$\mathcal{U}^\gamma$-CSP can be solved in $O(|A|^2 W|N|^2)$ for any non-decreasing function γ.*

Proof. Let $\text{opt}(P)$ be the optimal solution cost of optimization problem P. The proof results from equality

$$\text{opt}(\mathcal{U}^\gamma\text{-}CSP) = \min_{k=1,\ldots,|N|} \text{opt}(\mathcal{U}^\Gamma\text{-}CSP_k^{\leq}), \tag{10}$$

which has been proved in [22, (19)] for combinatorial optimization problems under cost uncertainty. $\square$

Unfortunately, the approach used in this section cannot be used to provide pseudo-polynomial algorithms for $\mathcal{U}^\gamma$-$TWSP$. The reason is that time windows restrictions are expressed through $|N|$ non-linear constraints, instead of a single linear constraint as it is the case for $\mathcal{U}^\gamma$-CSP. The issue has been discussed in [1] where the authors propose an extended linear programming formulation for $\mathcal{U}^\Gamma$-$TWSP$ using additional binary variables and a number of robust constraints that is quadratic in $|N|$. Since the formulation from [1] contains many robust constraints, applying Theorem 2 to that formulation would enable us to solve $\mathcal{U}^\Gamma$-$TWSP$ by solving more than $O\left(|A|^{|N|^2}\right)$ problems $TWSP$, instead of the $O(|A|)$ involved in Corollary 1. This approach would hardly be of any practical use.

4 Complexity of $\mathcal{U}^\Gamma$-$TWSP$ and $\mathcal{U}^\gamma$-$TWSP$

We show in this section that the decision version of a simplification of $\mathcal{U}^\Gamma$-$TWSP$ is $\mathcal{NP}$-complete in the strong sense. Namely, we consider the robust path with deadlines, obtained from the decision version of $\mathcal{U}^\Gamma$-$TWSP$ by looking for a path with zero nominal travel time that satisfies the upper time windows (hence we suppose $\overline{r} = 0$ and $\underline{b} = 0$); we also suppose that the graph is acyclic.

ROBUST PATH WITH DEADLINES ($\mathcal{U}^\Gamma$-PD)

Input: A directed and acyclic graph $D = (N, A)$ with corresponding $\hat{r}_a$ for each $a \in A$ and $\overline{b}_i$ for each $i \in N$, and a positive integer Γ.

Question: There exists a path $p = (i_0, i_1, \ldots, i_l)$ in D starting at $o \in N$ ($i_0 = o$) and ending at $d \in N$ ($i_l = d$) such that

$$\max\left\{\sum_{a\in S} \hat{r}_a \,\middle|\, S \subseteq (i_0, i_1, \ldots, i_h), |S| \leq \Gamma\right\} \leq \overline{b}_{i_h}, \text{ for each } h = 1, \ldots, l?$$

Our proof is based on reducing the decision version of the independent set problem to $\mathcal{U}^\Gamma$-PD. It is well-known that the former problem is $\mathcal{NP}$-complete in the strong sense (e.g. [14]).

INDEPENDENT SET (IS)

Input: An undirected graph $G = (V, E)$ and a positive integer K.

Question: There exists $W \subseteq V$ such that $|W| \geq K$ and $\{i, j\} \nsubseteq W$ for each $\{i, j\} \in E$?

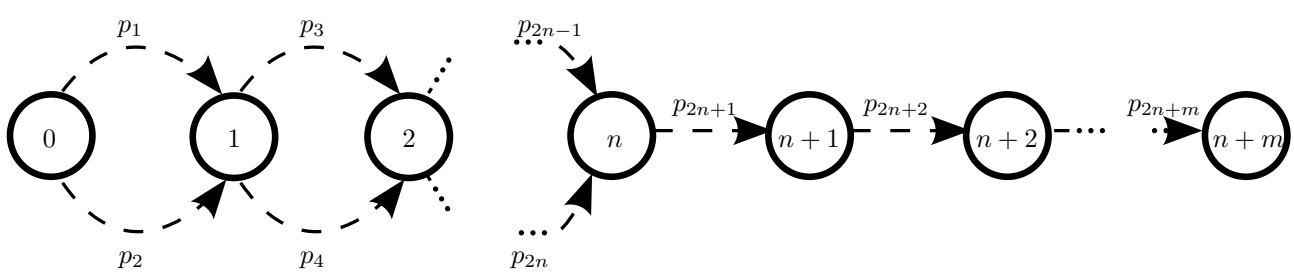

Figure 2: Reduction from the independent set problem to $\mathcal{U}^\Gamma$-PD.

Theorem 4. *$\mathcal{U}^\Gamma$-PD is $\mathcal{NP}$-complete in the strong sense.*

Proof. First, we show that $\mathcal{U}^\Gamma$-PD belongs to $\mathcal{NP}$. For that, the feasibility of a path $p = (i_0, i_1, \ldots, i_l)$ must be checked in polynomial time. This is true because, for each $h = 1, \ldots, l$, the time window constraint on i_h needs to be checked only for a single S containing $\min(h, \Gamma)$ arcs, with largest values of $\hat{r}_a$. Next, we show that IS can be reduced to $\mathcal{U}^\Gamma$-PD.

Given an instance to IS with $|V| = n$ nodes and $|E| = m$ edges, we show next how to build an instance for $\mathcal{U}^\Gamma$-PD. Graph D is described on Figure 2 where p_i represents a directed path that contains $m+1$ arcs for each $i \in \{1, \ldots, 2n\}$ and n arcs for each $i \in \{2n+1, \ldots, 2n+m\}$. Moreover, $o = 0$, $d = n + m$ and $\Gamma = n(m+1)$. In what follows we denote the k-th arc of p_i by $a(i, k)$ and the elements of E by $\{e_1, \ldots, e_m\}$.

One sees immediately that any path p from o to d must contain all paths p_{2n+i} for $i \in \{1, \ldots, m\}$. Furthermore, for each $i \in \{1, \ldots, n\}$, path p contains either p_{2i} or p_{2i-1}. Hence, there are 2^n different paths in D from o to d, which is as many as the number of different subsets of V. The correspondence between subsets of V and paths in D works as follows. Let $W \subseteq V$ be a subset of V and let p_W

be the path in D associated to W. Then, for each $i \in \{1, \ldots, n\}$, path p_W contains p_{2i} if $i \in W$ and path p_W contains p_{2i-1} otherwise. We can describe concisely p_W by introducing function $\widetilde{W}$ defined as follows: $\widetilde{W}(i) = 2i$ if $i \in W$ and $\widetilde{W}(i) = 2i - 1$ otherwise, for each $i \in \{1, \ldots, n\}$. Hence, $p_W = p_{\widetilde{W}(1)} \cup p_{\widetilde{W}(2)} \cup \ldots \cup p_{\widetilde{W}(n)} \cup p_{2n+1} \cup \ldots \cup p_{2n+m}$.

We explain next how to choose parameters $\hat{r}$ and $\bar{b}$ such that the deadline constraints imposed to p_W are equivalent to the constraints of IS imposed to W. Namely, we choose these parameters such that constraint

$$\max \left\{ \sum_{a \in S} \hat{r}_a \middle| S \subseteq p_{\widetilde{W}(1)} \cup p_{\widetilde{W}(2)} \cup \ldots \cup p_{\widetilde{W}(n)}, |S| = n(m+1) \right\} \leq \bar{b}_n \tag{11}$$

written for p_W is equivalent to $|W| \geq K$ and constraints

$$\max \left\{ \sum_{a \in S} \hat{r}_a \middle| S \subseteq p_{\widetilde{W}(1)} \cup \ldots \cup p_{\widetilde{W}(n)} \cup p_{2n+1} \cup \ldots \cup p_{2n+h}, |S| = n(m+1) \right\} \leq \bar{b}_{n+h} \tag{12}$$

written for p_W are equivalent to $e_h = \{i, j\} \nsubseteq W$, for each $h = 1, \ldots, m$. Notice that in (11) and (12), $|S| = n(m+1)$ has been used instead of $|S| \leq n(m+1)$ because all components of $\hat{r}$ are positive and the paths considered in (11) and (12) contain not less than $n(m+1)$ arcs. Our construction below is made in two steps. First, we impose restrictions on $\hat{r}$ and $\bar{b}$ which ensure that the corresponding deadline constraints translate exactly into the constraints of IS. To this end we introduce additional parameters μ, ν, and σ_h for each $h \in \{1, \ldots, m\}$ to be specified later. Second, we provide in Appendix B an example of vector $(\mu, \nu, \sigma, \hat{r}, \bar{b})$ that satisfies the restrictions and that is composed of polynomial functions of n, m, and K.

Constraint (11) The number of arcs of subpath $p_{\widetilde{W}(1)} \cup p_{\widetilde{W}(2)} \cup \ldots \cup p_{\widetilde{W}(n)}$ is equal to $n(m+1)$, so that the unique choice of S in the maximization is the full subpath $p_{\widetilde{W}(1)} \cup p_{\widetilde{W}(2)} \cup \ldots \cup p_{\widetilde{W}(n)}$. Hence, (11) becomes

$$\sum_{i=1}^{n} \sum_{k=1}^{m+1} \hat{r}_{a(\widetilde{W}(i),k)} \leq \bar{b}_n. \tag{13}$$

Let $\hat{r}$ and $\bar{b}_n$ be such that

$$\begin{aligned} \sum_{k=1}^{m+1} \hat{r}_{a(2i,k)} &= \mu, & i \in \{1, \ldots, n\}, \\ \sum_{k=1}^{m+1} \hat{r}_{a(2i-1,k)} &= \mu + 1, & i \in \{1, \ldots, n\}, \\ \bar{b}_n &= n\mu + n - K. \end{aligned} \tag{14}$$

Plugging (14) into (13), we obtain that p_W cannot contain more than $n - K$ paths of the form p_{2i-1} for $i \in \{1, \ldots, n\}$, which is equivalent to $|V \setminus W| \leq n - K$, or more simply, $|W| \geq K$.

h-th constraint of (12) We propose next restrictions on $\hat{r}$ and $\bar{b}$ such that the maximization in the lhs of the h-th constraint in (12) becomes

$$\sum_{i=1}^{n} \sum_{k=h+1}^{m+1} \hat{r}_{a(\widetilde{W}(i),k)} + \sum_{i=2n+1}^{2n+h} \sum_{k=1}^{n} \hat{r}_{a(i,k)}. \tag{15}$$

One readily checks that the following set of constraints yields the desired (15)

$$\begin{aligned} \hat{r}_{a(i,k)} &< \hat{r}_{a(j,k+1)}, & i, j \in \{1, \ldots, 2n\}, k \in \{1, \ldots, m\}, \\ \nu &> \hat{r}_{a(i,m)}, & i \in \{1, \ldots, 2n\}, \\ \nu &= \hat{r}_{a(i,k)}, & i \in \{2n+1, \ldots, 2n+m\}, k \in \{1, \ldots, n\}. \end{aligned} \tag{16}$$

Once the lhs of the h-th constraint of (12) has been turned to (15), we impose the following additional restrictions to obtain the desired deadline constraint

$$\begin{aligned}
\sum_{k=h+1}^{m+1} \hat{r}_{(2i,k)} &= \begin{cases} \sigma_h + 1, & \text{if } e_h \text{ is adjacent to vertex } i, \\ \sigma_h, & \text{otherwise.} \end{cases}, \quad i \in \{1, \ldots, n\}, \\
\sum_{k=h+1}^{m+1} \hat{r}_{(2i-1,k)} &= \sigma_h, \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad i \in \{1, \ldots, n\}, \\
\bar{b}_{n+k} &= 1 + n\sigma_h + hn\nu.
\end{aligned} \tag{17}$$

Plugging restrictions (16) and (17) into the h-th constraint of (12), we obtain that p_W cannot contain more than a single subpath p_{2i} such that i is adjacent to edge e_h. Hence, the number of vertices in W adjacent to e_h must not be greater than 1, which is equivalent to $\{i, j\} \nsubseteq W$ for $e_h = \{i, j\}$. $\square$

Since $\mathcal{U}^\Gamma$-PD is a special case of the decision problem associated to $\mathcal{U}^\Gamma$-$TWSP$, we obtain immediately the following result, which contrasts with the pseudo-polynomial algorithms proposed for $\mathcal{U}^\Gamma$-CSP in the previous section.

Corollary 3. *$\mathcal{U}^\Gamma$-$TWSP$ and $\mathcal{U}^\gamma$-$TWSP$ are $\mathcal{NP}$-hard in the strong sense.*

5 Label-setting algorithm

We show in this section how to extend the classical labels used in dynamic programming and label-setting algorithms (e.g. [10]) to the case of uncertain weights. We focus on the definition of robust labels and discuss dominance rules in case of budgeted uncertainty. Our description is presented for time windows; one can readily modify the algorithms presented below to a capacity constraint instead of time windows. Notice, however, that the computational complexity of the algorithm presented below is exponential in Γ, which is in accordance with the $\mathcal{NP}$-hardness of $\mathcal{U}^\Gamma$-$TWSP$.

In Section 5.1, we sketch the label-setting algorithm, recall how labels are defined in the deterministic setting and how dominance applies. We refer the interested reader to [8], among others, for a detailed description of the algorithm. In Section 5.2, we consider set $\mathcal{U}^\Gamma$ and suppose that Γ is integer. If it is not the case, we can always round up Γ to obtain a slightly more conservative model. In the (unlikely) situation where the fractional part of Γ really matters, one can always extend the algorithm described next in a way similar to the dynamic programming algorithms described in [22]. In Section 5.3, we consider $\mathcal{U}^\gamma$. In the following, we denote the cost of any path p shortly by $\kappa(p) = \sum_{a \in p} \kappa_a$.

5.1 Deterministic labels

We outline next the basic principles of the label-setting algorithm for $TWSP$. For each path p from node o to node i, the classical label-setting algorithm considers label $(\kappa(p), t_{|p|}(p))$ that records the cost of the path and its arrival time at node i. To avoid constructing all labels, the algorithm selects and extends labels in a special order. Two sets of labels are considered: permanent labels and non-permanent labels. At each iteration, the algorithm selects the smallest non-permanent label according to the lexicographical order and mark the label as permanent. Then, for each direct descendant of the node associated to the current label, we create a new non-permanent label and check whether it satisfies the time-windows and is not dominated by a permanent label. If one of the condition fails to hold, the label is immediately removed from the set of non-permanent labels. The algorithm ends when the next selected label corresponds to d.

A crucial phase in the label-setting algorithm is the removal of dominated labels, which reduces significantly the total number of labels searched in the course of the algorithm. Given two labels y and y' associated to paths p and p' ending at the same node, we say that label y' is dominated by label y if the following condition holds: *if path p' belongs to an optimal solution of $TWSP$, then path p belongs to an optimal solution of $TWSP$.* Dominated labels can be discarded from the search that occurs during the label-setting algorithm. The next result is well-known and is presented without proof.

Lemma 1. *Consider the TWSP and let $y = (\kappa, t)$ and $y' = (\kappa', t')$ be two labels associated to paths p and p' ending at the same node. Assume the following conditions are verified:*

1. $\kappa \leq \kappa'$
2. $t \leq t'$
3. *and at least one inequality is strict.*

Then, label y' is dominated by label y.

5.2 Robust labels

Next, we consider robust time windows constraints (4) recalled below for uncertainty set $\mathcal{U}^\Gamma$

$$t_j(p, r) \leq \bar{b}_{i_j}, \quad j = 1, \ldots, l, \; r \in \mathcal{U}^\Gamma, \tag{18}$$

where $p = (i_0, i_1, \ldots, i_l)$, and $l = |p|$. The naive approach to $\mathcal{U}^\Gamma$-$TWSP$ would express the problem as a shortest path problem with $|\mathcal{U}^\Gamma|$ time-windows constraints, one for each $r \in \mathcal{U}^\Gamma$. Defining $s = |\mathcal{U}^\Gamma|$ and $\mathcal{U}^\Gamma = \{r^1, \ldots, r^s\}$, the naive approach would associate to node i and each path p from o to i a label with s resources,

$$\left(\kappa(p), t_{|p|}(p, r^1), \ldots, t_{|p|}(p, r^s)\right), \tag{19}$$

and extend the label through arc (i, j) with the classical formula

$$\left(\kappa(p) + \kappa_{ij}, \max(\underline{b}_i, t_{|p|}(p, r^1) + r^1_{ij}), \ldots, \max(\underline{b}_i, t_{|p|}(p, r^s) + r^s_{ij})\right). \tag{20}$$

Using labels (19) and appropriate data structures, one can solve $\mathcal{U}^\Gamma$-$TWSP$ by the label-setting algorithm in solution time $O(s|A|B^s)$, where $s = \sum_{k=1}^{\Gamma} \binom{|A|}{k}$. We show below how to reduce the solution time to $O(\Gamma|A|B^{\Gamma+1})$ by using a more compact description of the labels. Notice that this is a significant improvement since Γ is always much smaller than s. To be more precise, this improvement leads to pseudo-polynomial algorithms when Γ is fixed, which happens in applications where we are interested to be protected only against a small number of deviations, regardless to the size of the instances. However, if we choose Γ according to probabilistic guarantees, then $\Gamma \in O(|A|^{1/2})$ (see [7, 21]). In the latter case, the resulting computing time of the label-setting algorithm would be $O(|A|^{3/2} B^{O(|A|^{1/2})})$ which is not polynomial, although asymptotically smaller than $O(s|A|B^s)$.

Remark 4. *The author of [22] has shown how generic dynamic programming algorithms for combinatorial optimization problems can be extended to the robust versions of the problems when only costs are uncertain and the uncertainty set is $\mathcal{U}^\Gamma$. His work shows that the number of states of the robust dynamic programming algorithms needs only to be multiplied by Γ, and therefore, polynomial algorithms stay polynomial in the robust case. This is in sharp contrast with the robust label-setting algorithm presented herein, which multiplies the number of labels by $B^{\Gamma+1}$. Unfortunately, the approach of [22] cannot be applied to the problems studied in this paper because herein the uncertainty lies in the constraints while only costs are uncertain in [22].*

Before explaining how the running-time can be decreased to $O(\Gamma|A|B^{\Gamma+1})$, we present without proof the well-known extension of Lemma 1 to multiple-resource label (19).

Lemma 2. *Consider the $\mathcal{U}^\Gamma$-$TWSP$ and let $y = (\kappa, t^1, \ldots, t^s)$ and $y' = (\kappa', t'^1, \ldots, t'^s)$ be two labels associated to paths p and p' ending at the same node. Assume the following conditions are verified:*

1. $\kappa \leq \kappa'$
2. $t^j \leq t'^j$, *for each* $j = 1, \ldots, s$
3. *and at least one inequality is strict.*

Then, label y' is dominated by label y.

The key idea to reduce the running-time of the robust label-setting algorithm is based on rewriting time windows (18) as

$$\max_{r \in \mathcal{U}^\Gamma} t_j(p,r) \leq \overline{b}_{i_j}, \quad j = 1, \ldots, l. \tag{21}$$

Our objective is to define *robust labels* that contain only the necessary information to test whether the current path is feasible when considering its maximum travel time over $\mathcal{U}^\Gamma$. Rather than considering all possible travel times $r \in \mathcal{U}^\Gamma$ that could be used along path $p = (i_0, \ldots, i_l)$ from $o = i_0$ to $i = i_l$, we can define the label attached to node i and path p as

$$\left(\kappa(p), \tau_l^0(p), \ldots, \tau_l^\Gamma(p)\right), \tag{22}$$

where, for each $j = 1, \ldots, l$, $\tau_j^g(p)$ is defined as the maximum arrival at node i_j when considering up to $g \in \{0, \ldots, \Gamma\}$ deviations when $g \leq |p|$ and is equal 0 otherwise, that is,

$$\tau_j^g(p) = \begin{cases} \max\limits_{r \in \mathcal{U}^g} t_j(p,r), & \text{for each } g \in \{0, \ldots, \min(|p|, \Gamma)\}, \\ 0, & \text{for each } g \in \{|p|+1, \ldots, \Gamma\}. \end{cases} \tag{23}$$

Then, we extend the label through arc (i,k), generating a new label for node $k = i_{l+1}$, with formula

$$\begin{cases} \kappa = \kappa(p) + \kappa_{ij}, \\ \tau_{l+1}^0 = \max\left(\underline{b}_j, \tau_l^0(p) + \overline{r}_{ij}\right), \\ \tau_{l+1}^g = \max\left(\underline{b}_j, \tau_l^{g-1}(p) + \overline{r}_{ij} + \hat{r}_{ij}, \tau_l^g(p) + \overline{r}_{ij}\right), & \text{for each } g \in \{1, \ldots, \bar{g}\}, \\ \tau_{l+1}^{\bar{g}+1} = \max\left(\underline{b}_j, \tau_l^{\bar{g}-1}(p) + \overline{r}_{ij} + \hat{r}_{ij}\right), \end{cases} \tag{24}$$

where $\bar{g} = \min\left(|p|, \Gamma - 1\right)$ and the extended label is feasible if $\tau^{\bar{g}+1}$ is less than or equal to $\overline{b}_j$. It is easy to see by induction that extending label $(0, 0 \ldots, 0)$, that corresponds to the empty path, iteratively through formula (24) leads exactly to definition (23). One readily sees that the solution time of the label-setting algorithm based on label (22) is reduced to $O(\Gamma|A|B^{\Gamma+1})$ since these labels contain $\Gamma + 1$ resources and their extension through new arcs can be done in $O(\Gamma)$ with the help of formula (24). Finally, the next result states the new robust dominance rule.

Lemma 3. *Consider the $\mathcal{U}^\Gamma$-TWSP and let $z = (\kappa, \tau^0, \ldots, \tau^\Gamma)$ and $z' = (\kappa', \tau'^0, \ldots, \tau'^\Gamma)$ be two labels associated to paths p and p' ending at the same node. Assume the following conditions are verified:*

1. *$\kappa \leq \kappa'$*
2. *$\tau^j \leq \tau'^j$, for each $j = 0, \ldots, \Gamma$*
3. *and at least one inequality is strict.*

Then, label z' is dominated by label z.

Proof. We prove the result by contradiction. Namely, suppose that p' belongs to an optimal solution to $\mathcal{U}^\Gamma$-$TWSP$, denoted p'', while p does not belong to an optimal solution to $\mathcal{U}^\Gamma$-$TWSP$. Consider then path $p^* = p \cup (p'' \backslash p')$. One readily sees that path p^* is optimal, yielding a contradiction. □

5.3 Variable budgeted uncertainty

Let us now address the case of variable budgeted uncertainty $\mathcal{U}^\gamma$-$TWSP$. Again, we suppose that γ is an integer-valued function, and if this is not case, we consider instead the rounded function $\lceil \gamma \rceil$. Notice that while the rounded function $\lceil \gamma \rceil$ is non-decreasing (because γ is non-decreasing), $\lceil \gamma \rceil$ is usually not affine. This is fortunately not a requirement for the algorithm described in this section.

The difference between $\mathcal{U}^\Gamma$-$TWSP$ and $\mathcal{U}^\gamma$-$TWSP$ is that the budget of uncertainty against which the solution must be protected depends on the cardinality of the solution. Hence, we define *variable robust labels* as robust labels to which we append the cardinality of the current path $k = |p|$, that is, $(\kappa, \tau^0, \ldots, \tau^\Gamma, k)$, where τ^g is defined as before (Eq. (23)) for each $g \in \{0, \ldots, \Gamma\}$. Then, we extend labels through arc (i,j) similarly to formula (24), and increasing k by one unit.

With variable budgeted uncertainty, smaller paths suffer less deviations than longer ones. Namely, to check the feasibility of label $(\kappa, \tau^0, \ldots, \tau^\Gamma, k)$, we only need to check that time windows are satisfied by τ^g for $g = \lceil \gamma(k) \rceil$. In opposition, the check for robust labels must be done for $g = \Gamma$ for classical robust labels. This implies that smaller paths must be preferred over longer ones, as formalized in the lemma below.

Lemma 4. *Consider the $\mathcal{U}^\gamma$-$TWSP$ and let $w = (\kappa, \tau^0, \ldots, \tau^\Gamma, k)$ and $w' = (\kappa', \tau'^0, \ldots, \tau'^\Gamma, k')$ be two labels associated to paths p and p' ending at the same node. Assume the following conditions are verified:*

1. $\kappa \leq \kappa'$
2. $\tau^j \leq \tau'^j$, *for each* $j = 0, \ldots, \Gamma$
3. $k \leq k'$
4. *and at least one inequality is strict.*

Then, label w' is dominated by label w.

Proof. The proof follows directly from the fact that γ is non-decreasing. □

One readily sees that the classical label-setting algorithm can be applied to variable robust labels. With appropriated data structures, the complexity of the resulting algorithm is $O(\Gamma|A||N|B^{\gamma(|N|)+1})$.

6 Computational experiments

We present next our numerical assessment of the algorithms discussed in the paper for problems $\mathcal{U}^\Gamma$-$TWSP$ and $\mathcal{U}^\gamma$-$TWSP$. The proposed solution approaches have been coded in JAVA and our numerical experiments have been carried out on an Intel(R) Core(TM) i7 CPU M 620, 2.67 GHz, 4 GB RAM machine, under microsoft operating system. The experimental results underline that the robust version of the $TWSP$ is not much harder to solve than its deterministic counterpart. Furthermore, the results show that, while the dynamic programming approach for $\mathcal{U}^\gamma$-$TWSP$ requires more computational effort than for solving $\mathcal{U}^\Gamma$-$TWSP$, $\mathcal{U}^\gamma$-$TWSP$ provides cheaper optimal solution than $\mathcal{U}^\Gamma$-$TWSP$. More details about the computational results are reported in what follows. First, we describe the instances used to assess the behavior of the proposed dynamic programming approaches. Then, we discuss the numerical results.

6.1 Test problems

test	nodes	arcs	density
G1	625	2400	3.84
G2	2500	9800	3.92
G3	5625	22200	3.95
G4	15625	62000	3.97
G5	22500	89400	3.97
G6	30625	121800	3.98
G7	40000	159200	3.98

Table 1: Characteristics of the test problems proposed in [13].

Our experiments are based on the grid networks from Class 6 used in [13]. The characteristic of these networks are reported in Table 1. For each network, we generate a set of different instances by varying the values of Γ (or γ) and $\rho = \hat{r}_a/\overline{r}_a$ as follows. Ratio ρ takes each value in $\{0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. For the value of Γ, we consider two sets of instances $S1$ and $S2$ defined below.

Generating time windows We describe next how we generate $\underline{b}$ and $\overline{b}$ such that all instances always have a feasible solution. We let $\hat{T}$ and $\bar{T}$ represent the shortest paths trees with respect to $\overline{r}+\hat{r}$ and $\overline{r}$, respectively. We let $\hat{p}$ denote the path from node o to node d in $\hat{T}$ and the width of time windows is denoted by w (and set to value in $\{40, 100, 200\}$). The values of $\underline{b}_i$ and $\overline{b}_i$ are set according to the following rule

$$\begin{cases} \underline{b}_i = \hat{T}_i - w; \overline{b}_i = \hat{T}_i, & \text{for each } i \in \hat{p}, \\ \underline{b}_i = \hat{T}_i - w; \overline{b}_i = \hat{T}_i, & \text{for each } i \notin \hat{p} \textit{ and rand } < 0.5, \\ \underline{b}_i = \bar{T}_i; \overline{b}_i = \bar{T}_i + w, & \text{for each } i \notin \hat{p} \textit{ and rand } \geq 0.5, \end{cases} \tag{25}$$

where $\hat{T}_i$ and $\bar{T}_i$ are the costs of the paths from o to i in $\hat{T}$ and $\bar{T}$, respectively, and *rand* is chosen randomly through the uniform distribution $[0, 1]$.

Instances $S1$ The set contains instances with no connections to the probabilistic bounds discussed in Section 2.2. To build these instances, we let Γ take each value in $\{3, 6, 12, 25, 50\}$. These values are such that the cardinality of the optimal paths are less than Γ.

Instances $S2$ The set contains instances for which γ and Γ are generated according to the probabilistic bounds. Namely, we compute function γ for each probability level ϵ in $\{0.01, 0.05, 0.1\}$ following the construction outlined in Appendix A. For Γ, we compute first an upper bound UB on the longest feasible path for the problem and then set $\Gamma = \gamma(UB)$, see Remark 3. Bound UB is computed as follows: for each node in $i \in V$, we let $\underline{r}_i$ be the value of the lowest resource arc outgoing from i, $\underline{r}_i = \min_{(i,j)\in A} \overline{r}_{ij}$, and we sort the nodes in V by non-decreasing values of $\underline{r}$. Then, UB is equal to

$$\max_{UB \in \mathbb{Z}_+} \left\{ UB : \sum_{i=1}^{UB} \underline{r}_i \leq b_{UB} \right\}.$$

Table 2 reports the values of UB and the time needed to compute the bounds for the $TWSP$. The values of Γ are shown in the last three columns for different values of ϵ.

Price of robustness In our results, we report the optimal solution cost for the deterministic problems, while the solution costs for robust problems are provided under the form of prices of robustness. The latter can be computed as follows

$$\text{PoR} = \frac{\text{opt}(P) - \text{opt}(TWSP)}{\text{opt}(TWSP)},$$

where $\text{opt}(P)$ is the optimal solution cost of problem $P \in \{\mathcal{U}^\Gamma\text{-}TWSP, \mathcal{U}^\gamma\text{-}TWSP\}$. PoR represents the additional cost required to obtain solutions robust with a prescribed guarantee. Notice, however, that in real applications this cost is significantly smaller than the cost needed to repair the deterministic solutions that become infeasible in the presence of uncertainty.

6.2 Numerical results

Our computational results are presented in Tables 3–5 where we provide the solution times in seconds, numbers of generated labels, solution costs, and prices of robustness. Since for set $S2$ we have a unique value of Γ, each row of Table 5 shows the average results over ρ only. Each row of Table 4 averages the results for the different values of ρ and Γ.

Instances in $S1$ The computational results for instances in $S1$ are presented in Tables 3 and 4. Table 3 provides average results for $TWSP$ and $\mathcal{U}^\Gamma$-$TWSP$ taken over the different values of ρ and Γ, while Table 4 shows the variations observed for the different values of Γ. We see that for the instances tested, $\mathcal{U}^\Gamma$-$TWSP$ does not seem much harder to solve that $TWSP$. This may be surprising given the exponential (in Γ) worst-case running time of the robust label-setting algorithm. However, the label-setting algorithm is well-known for generating only a subset of all possible labels.

				$\lceil\gamma(UB)\rceil$		
test problems	$\hat{r}_a/\overline{r}_a$	UB	time	0.01	0.05	0.1
G1	0.5	143	8.00	29	21	16
	0.6	148	5.00	29	21	17
	0.7	154	3.00	30	22	17
	0.8	160	3.00	30	22	17
	0.9	165	3.00	31	22	18
	1	173	3.00	32	23	18
G2	0.5	298	46.00	41	29	23
	0.6	309	41.00	42	30	24
	0.7	320	46.00	43	30	24
	0.8	333	44.00	43	31	24
	0.9	344	45.00	44	32	25
	1	360	43.00	45	32	25
G3	0.5	450	217.00	50	36	28
	0.6	466	208.00	51	37	29
	0.7	484	215.00	52	37	29
	0.8	502	206.00	53	38	30
	0.9	518	227.00	54	38	30
	1	544	208.00	55	39	31
G4	0.5	734	1623.00	64	46	36
	0.6	761	1613.00	65	46	36
	0.7	790	1568.00	66	47	37
	0.8	819	1624.00	68	48	38
	0.9	845	1613.00	69	49	38
	1	887	1566.00	70	50	39
G5	0.5	887	3346.00	70	50	39
	0.6	918	3140.00	71	51	40
	0.7	954	3254.00	73	52	41
	0.8	989	3341.00	74	53	41
	0.9	1021	3251.00	75	54	42
	1	1071	3112.00	77	55	43
G6	0.5	1028	6192.00	76	54	42
	0.6	1064	6138.00	77	55	43
	0.7	1106	6253.00	78	56	44
	0.8	1146	6412.00	80	57	44
	0.9	1183	6004.00	81	58	45
	1	1243	6086.00	83	59	46
G7	0.5	1187	10339.00	81	58	45
	0.6	1230	10225.00	83	59	46
	0.7	1276	10787.00	84	60	47
	0.8	1324	10484.00	86	61	48
	0.9	1367	10307.00	87	62	48
	1	1434	10390.00	89	63	50
AVG		741.31	3100.93	61.45	43.88	34.36

Table 2: Computation of UB and Γ for $\epsilon \in \{0.01, 0.05, 0.1\}$.

Still, Table 4 confirms that increasing Γ tends to increase solution times, which is particularly visible when Γ changes from 12 to 25 or 50. Another observation one can make is that widening time windows increases the difficulty of solving all problems, and that the increase in solution times is more marked for $\mathcal{U}^{\Gamma}$-$TWSP$ than for $TWSP$. Indeed, comparing $w = 40$ with $w = 100$, the average solution times for the $TWSP$ are multiplied by 1.7 while they are multiplied by 2.9 for the $\mathcal{U}^{\Gamma}$-$TWSP$. This, again, can easily be explained by the larger dimension of the labels in the robust case, which makes more room for creating more labels when time windows are widened. Regarding the optimal solution costs (and the related prices of robustness), the results show that, as expected, increasing the value of Γ raises the price of robustness, while widening time windows reduces the price of robustness.

Instances in $S2$ Table 5 shows that the robust $TWSP$ with variable budgeted uncertainty is more difficult to solve than $\mathcal{U}^{\Gamma}$-$TWSP$. Indeed, the computational cost of the label-setting algorithm for solving the $\mathcal{U}^{\gamma}$-$TWSP$ is, on average, 1.52 times higher than the time required for solving the $\mathcal{U}^{\Gamma}$-$TWSP$ (see Table 3). This behavior is justified by the number of generated labels. Indeed, the labels generated when solving $\mathcal{U}^{\gamma}$-$TWSP$ are 1.54 times higher than that processed by the label-setting algorithm to solve $\mathcal{U}^{\Gamma}$-$TWSP$. The higher number of labels is due to the fact that the feasibility check is done for a subset of the element of the labels. It follows, that unfeasible labels for the $\mathcal{U}^{\Gamma}$-$TWSP$ are feasible for the $\mathcal{U}^{\gamma}$-$TWSP$. The table also shows that model $\mathcal{U}^{\gamma}$-$TWSP$ leads to significant reduction in the price of robustness obtained by model $\mathcal{U}^{\Gamma}$-$TWSP$.

		$TWSP$				$\mathcal{U}^{\Gamma}$-$TWSP$			
w	test problems	time	#labels	cost	$\lvert p \rvert$	time	#labels	PoR %	$\lvert p \rvert$
40	G1	0.02	159.83	4427.00	48.00	0.02	130.17	5.64	48.00
	G2	0.17	318.00	8707.00	98.33	0.18	360.11	8.79	98.00
	G3	0.47	442.33	12663.67	148.00	0.77	746.92	3.38	148.00
	G4	2.50	851.67	21675.33	246.00	4.11	1374.36	4.01	245.00
	G5	3.95	912.83	15226.33	293.33	6.37	1458.81	3.39	293.00
	G6	6.50	1086.00	24798.00	340.00	14.08	2284.78	4.06	339.00
	G7	9.54	1213.83	22215.33	392.33	16.07	1929.17	3.77	392.00
	AVG	3.31	712.07	15673.24	223.71	5.94	1183.47	4.72	223.29
100	G1	0.05	292.67	4423.67	49.00	0.03	201.00	5.43	48.28
	G2	0.24	555.83	8198.00	98.67	0.40	824.61	5.46	98.00
	G3	0.70	705.83	12569.33	149.33	1.65	1567.22	2.11	148.28
	G4	3.66	1283.00	20689.67	248.00	13.60	3889.14	2.02	245.83
	G5	6.70	1538.17	14812.00	294.33	25.59	4994.25	0.91	293.39
	G6	11.01	1763.67	24557.67	342.00	32.43	5125.14	2.11	339.78
	G7	15.81	1953.50	21485.67	395.33	46.05	5460.56	2.19	392.83
	AVG	5.45	1156.10	15248.00	225.24	17.11	3151.70	2.89	223.77
200	G1	0.09	766.83	4142.00	48.00	0.06	530.47	12.54	48.06
	G2	0.55	1321.83	8225.00	100.33	0.60	1385.25	4.72	98.44
	G3	1.27	1410.00	12357.00	152.33	2.34	2451.89	2.50	148.44
	G4	4.92	1867.00	21256.00	247.67	19.05	5424.03	0.37	246.83
	G5	8.92	2335.50	13870.67	295.33	25.93	5618.75	0.69	295.11
	G6	13.84	2577.67	24845.00	347.00	47.86	7254.61	2.15	342.33
	G7	25.43	3463.83	21728.00	397.00	136.68	11863.39	0.62	396.17
	AVG	7.86	1963.24	15203.38	226.81	33.22	4932.63	3.37	225.06

Table 3: Average computational results for the instances belonging to set $S1$.

	40				100				200			
Γ	time	#labels	PoR %	$\lvert p \rvert$	time	#labels	PoR %	$\lvert p \rvert$	time	#labels	PoR %	$\lvert p \rvert$
50	12.72	2470.57	8.09	223.29	50.50	9726.38	6.31	223.29	95.15	15407.14	4.03	223.67
25	7.47	1480.98	6.16	223.29	20.51	3817.83	3.46	223.29	32.44	5240.19	2.37	224.29
12	5.00	1014.05	4.52	223.29	10.19	1879.74	2.21	223.29	18.12	2336.24	1.94	224.95
6	3.91	798.95	3.48	223.29	7.68	1268.55	1.48	223.48	18.85	2425.29	0.89	225.29
3	3.44	691.52	2.36	223.29	6.79	1114.86	0.51	224.19	18.49	2187.29	0.52	225.76
AVG	6.51	1291.21	4.92	223.29	19.13	3561.47	2.79	223.50	36.61	5519.23	1.95	224.79

Table 4: Average computational results for each value of Γ when solving the instances belonging to the set $S1$.

		$TWSP$				$\mathcal{U}^{\Gamma}$-$TWSP$				$\mathcal{U}^{\gamma}$-$TWSP$			
ϵ	test problems	time	#labels	cost	$\|p\|$	time	#labels	PoR %	$\|p\|$	time	#labels	PoR %	$\|p\|$
0.01	G1	0.02	159.83	4427.00	48.00	0.01	108.50	7.30	48.00	0.05	261.50	4.96	48.00
	G2	0.17	318.00	8707.00	98.33	0.20	489.00	12.60	98.00	0.61	1323.33	10.69	98.00
	G3	0.47	442.33	12663.67	148.00	1.22	1388.00	7.57	148.00	3.09	3108.17	5.35	148.00
	G4	2.50	851.67	21675.33	246.00	7.78	3127.33	8.00	245.00	21.24	7358.00	5.59	245.00
	G5	3.95	912.83	15226.33	293.33	14.64	4034.17	6.16	293.00	40.11	9551.67	5.30	293.00
	G6	6.50	1086.00	24798.00	340.00	51.04	9271.33	10.03	339.00	124.94	20666.50	6.52	339.00
	G7	9.54	1213.83	22215.33	392.33	39.25	4929.17	8.78	392.00	360.57	15381.50	6.42	392.00
	AVG	3.31	712.07	15673.24	223.71	16.31	3335.36	8.63	223.29	78.66	8235.81	6.40	223.29
0.05	G1	0.02	159.83	4427.00	48.00	0.02	141.33	7.25	48.00	0.03	257.83	4.88	48.00
	G2	0.17	318.00	8707.00	98.33	0.22	455.00	12.12	98.00	0.44	984.50	10.24	98.00
	G3	0.47	442.33	12663.67	148.00	1.26	1218.83	6.29	148.00	1.95	2059.33	3.63	148.00
	G4	2.50	851.67	21675.33	246.00	7.64	2750.83	7.42	245.00	14.33	5335.83	5.15	245.00
	G5	3.95	912.83	15226.33	293.33	12.63	3092.50	5.92	293.00	23.81	5901.17	4.58	293.00
	G6	6.50	1086.00	24798.00	340.00	37.52	6324.83	8.80	339.00	70.92	12058.33	6.06	339.00
	G7	9.54	1213.83	22215.33	392.33	34.54	4316.00	7.75	392.00	80.05	10176.50	5.70	392.00
	AVG	3.31	712.07	15673.24	223.71	13.41	2614.19	7.93	223.29	27.36	5253.36	5.75	223.29
0.1	G1	0.02	159.83	4427.00	48.00	0.02	144.83	6.22	48.00	0.03	226.17	4.88	48.00
	G2	0.17	318.00	8707.00	98.33	0.21	441.00	11.57	98.00	0.41	899.50	9.72	98.00
	G3	0.47	442.33	12663.67	148.00	1.04	1039.83	5.91	148.00	1.57	1661.00	2.56	148.00
	G4	2.50	851.67	21675.33	246.00	6.57	2210.67	5.80	245.00	12.13	4039.00	4.46	245.00
	G5	3.95	912.83	15226.33	293.33	10.58	2517.67	5.50	293.00	19.15	4417.33	4.36	293.00
	G6	6.50	1086.00	24798.00	340.00	29.03	4934.33	8.14	339.00	46.49	8107.50	5.52	339.00
	G7	9.54	1213.83	22215.33	392.33	28.48	3580.33	7.34	392.00	57.93	7778.50	5.15	392.00
	AVG	3.31	712.07	15673.24	223.71	10.85	2124.10	7.21	223.29	19.67	3875.57	5.24	223.29

Table 5: Average computational results for the instances belonging to set $S2$.

7 Concluding remarks

In this paper, we have addressed a variant of the constrained shortest path problem. In particular, we have assumed that the resource consumption is not known with precision, rather, it belongs to an uncertainty set. We have proved that the problem is $\mathcal{NP}$-hard in the strong sense for unbounded uncertainty sets. To overcome this issue, we have considered the budgeted uncertainty set and the variable budgeted uncertainty set. Indeed, the considered sets have been successfully applied to combinatorial optimization problems, and in many cases, the complexity of the robust counterparts are similar to the complexity of the deterministic problem.

We have analyzed two types of resource constraints, that is, time windows and capacity. We have proved that the latter version of the problem can be solved in pseudo-polynomial time, thus, the robust version maintains the complexity of the deterministic counterpart. Interestingly, the complexity of the robust version of the time windows shortest path problem increases. Indeed, we have proved it is $\mathcal{NP}$-hard in the strong sense, using a reduction from the independent set problem. This is the first example of a weakly $\mathcal{NP}$-hard combinatorial optimization problem whose robust version (under budgeted uncertainty) is $\mathcal{NP}$-hard in the strong sense.

A dynamic programming-based solution approach has been devised in order to solve the problems at hand. Robust labels have been defined and the well-known dominance relations have been extended to address the robustness. The developed label-setting algorithm has been tested by considering instances inspired from the scientific literature. Several scenarios have been considered in order to assess the behavior of the proposed solution strategy. The computational results underline the effectiveness of the algorithm in solving the robust version of the time windows shortest path problem with budgeted uncertainty set. In addition, we have shown that using the variable budgeted uncertainty set a reduction in the price of robustness is observed.

It is worth observing that the defined label-setting algorithm is able to solve the robust shortest path problem with capacity constraint. However, speed up techniques have to be considered, that is, use of lower and upper bounds on both the cost and the resource consumption and pre-processing procedures in order to reduce the dimension of the networks by removing nodes and arcs. The extension of the proposed dynamic programming-based solution approach to the problem with capacity constraint is an interesting subject of future investigation.

References

[1] A. Agra, M. Christiansen, R. Figueiredo, L. M. Hvattum, M. Poss, and C. Requejo. Layered formulation for the robust vehicle routing problem with time windows. In *ISCO*, pages 249–260, 2012.

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

[3] H. Aissi, C. Bazgan, and D. Vanderpooten. Min–max and min–max regret versions of combinatorial optimization problems: A survey. *European journal of operational research*, 197(2):427–438, 2009.

[4] E. Álvarez-Miranda, I. Ljubić, and P. Toth. A note on the bertsimas & sim algorithm for robust combinatorial optimization problems. *4OR*, 11(4):349–360, 2013.

[5] A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[6] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98:49–71, 2003.

[7] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52:35–53, 2004.

[8] N. Boland, J. Dethridge, and I. Dumitrescu. Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Operations Research Letters*, 34(1):58 – 68, 2006.

[9] D. Catanzaro, M. Labbé, and M. Salazar-Neumann. Reduction approaches for robust shortest path problems. *Computers & OR*, 38(11):1610–1619, 2011.

[10] M. Desrochers and F. Soumis. A generalized permanent labelling algorithm for the shortest path problem with time windows. *INFOR*, 26:191–212, 1988.

[11] L. Di Puglia Pugliese and F. Guerriero. A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*, 62(3):183–200, 2013.

[12] M. Dror. Note on the complexity of the shortest path models for column generation in vrptw. *Operations Research*, 42(5):977–978, 1994.

[13] I. Dumitrescu and N. Boland. Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem. *Networks*, 42:135 – 153, 2003.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[15] K.-S. Goetzmann, S. Stiller, and C. Telha. Optimization over integers with robustness in cost and few constraints. In *WAOA*, pages 89–101, 2011.

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

[17] S. Irnich and G. Desaulniers. *Shortest Path Problems with Resource Constraints*, chapter 2, pages 33–65. GERAD 25th Anniversary Series. Springer, 2005.

[18] B. Jaumard, F. Semet, and T. Vovor. A two-phase resource constrained shortest path algorithm for acyclic graphs. *Cahiers du GERAD*, 1996.

[19] H. C. Joksch. The shortest route problem with constraints. *Journal of Mathematical Analysis and Applications*, 14(2):191 – 197, 1966.

[20] P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*, volume 14. Springer, 1997.

[21] M. Poss. Robust combinatorial optimization with variable budgeted uncertainty. *4OR*, 11(1):75–92, 2013.

[22] M. Poss. Robust combinatorial optimization with variable cost uncertainty. *European Journal of Operational Research*, 237(3):836 – 845, 2014.

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

[24] F. Talla Nobibon and R. Leus. Complexity results and exact algorithms for robust knapsack problems. *Journal of Optimization Theory and Applications*, 161(2):533–552, 2014.

A Probabilistic bounds

Bertismas and Sim [7] present various examples of bounds $\delta(\Gamma, |A|)$. In our computational results, we use bound

$$\delta(\Gamma, |A|) = \frac{1}{2^{|A|}} \left((1-\mu)\binom{|A|}{\lfloor \nu \rfloor} + \sum_{l=\lfloor \nu \rfloor + 1}^{|A|} \binom{|A|}{l} \right) \tag{26}$$

where $\nu = (\Gamma + |A|)/2$ and $\mu = \nu - \lfloor \nu \rfloor$. From (26), we see that the smallest value of Γ such that the resource constraint are satisfied with probability $1 - \epsilon$ is given by the solution of minimization problem

$$\min \Gamma \quad \text{s.t.} \quad \delta(\Gamma, |A|) \leq \epsilon, \tag{27}$$

which can be solved by a bisection method. We can then compute γ in two steps. First, we compute function $\hat{\Gamma} : \{1, \ldots, |A|\} \to \Gamma$ by solving (27) when $|A|$ is replaced by k which takes any value in $\{1, \ldots, |A|\}$. Then, we compute an affine upper-bounding approximation of the resulting function. The reader is redirected to [21] for more details on a similar construction.

B Existence of a solution $(\mu, \nu, \sigma, \hat{r}, \overline{b})$ to (14),(16), and (17)

Let $\hat{r}$ be defined as follows. For each $i \in \{1, \ldots, n\}$ and $k \in \{2, \ldots, m\}$, we have

$$\hat{r}_{a(2i,k)} = \begin{cases} 4(k-1)+1, & \text{if } e_{k-1} \text{ is adjacent to } i \text{ and } e_k \text{ is not,} \\ 4(k-1)-1, & \text{if } e_k \text{ is adjacent to } i \text{ and } e_{k-1} \text{ is not,} \\ 4(k-1), & \text{otherwise,} \end{cases}$$

and for each $i \in \{1, \ldots, n\}$, we have $\hat{r}_{a(2i-1,1)} = 2$, and

$$\hat{r}_{a(2i,1)} = \begin{cases} 0, & \text{if } e_1 \text{ is adjacent to vertex } i, \\ 1, & \text{otherwise.} \end{cases}$$

Moreover,

$$\hat{r}_{a(2i,m+1)} = \begin{cases} 4m+1, & \text{if } e_m \text{ is adjacent to } i, \\ 4m, & \text{otherwise,} \end{cases}$$

and $\hat{r}_{a(2i-1,k)} = 4(k-1)$, for each $k \in \{2, \ldots, m+1\}$ and $i \in \{1, \ldots, n\}$.

Parameters μ and σ follow from these definitions. Namely,

$$\sigma_h = 2(m+h)(m-h+1),$$

for each $h \in \{1, \ldots, m\}$ and $\mu = \sigma_1 + 1$. Then we must fix ν to any integer greater than $4(m-1)+1$.

Finally, deadlines $\overline{b}_n$ and $\overline{b}_{n+h}$ for each $h \in \{1, \ldots, m\}$ are chosen according to the equations in (14) and (17), respectively, while other deadlines can be set, for instance, to $n(\mu+1)$.