Robust combinatorial optimization with cost uncertainty

Michael Poss

UMR CNRS 6599 Heudiasyc, Université de Technologie de Compiègne, Centre de Recherches de Royallieu, 60200 Compiègne, France.

Abstract

We present in this paper a new model for robust combinatorial optimization with cost uncertainty that generalizes the classical budgeted uncertainty set. We suppose here that the budget of uncertainty is given by a function of the problem variables, yielding an uncertainty multifunction. The new model is less conservative than the classical model and approximates better Value-at-Risk objective functions, especially for vectors with few non-zero components. Examples of budget functions are constructed from the probabilistic bounds computed by Bertsimas and Sim. We provide an asymptotically tight bound for the cost reduction obtained with the new model. We turn then to the tractability of the resulting optimization problems and provide a mixed-integer linear reformulation. We show that when the budget function is affine, the resulting optimization problems can be solved by solving $n + 1$ deterministic problems. We propose combinatorial algorithms to handle problems with more general budget functions. We also adapt existing dynamic programming algorithms to solve faster the robust counterparts of optimization problems, which can be applied both to the traditional budgeted uncertainty model and to our new model. We evaluate numerically the reduction in the price of robustness obtained with the new model on the shortest path problem and on a survivable network design problem.

Keywords: Combinatorial optimization; Robust Optimization; Dynamic Programming; Price of Robustness; Budgeted Uncertainty.

Email address: mjposs@gmail.com (Michael Poss)

1. Introduction

Let $X \subset \{0,1\}^n$ be the feasibility set of a combinatorial optimization problem and c be a cost vector in $\mathbb{R}^n$. We study in this paper optimization problems of the form

$$CO \quad \equiv \quad \min_{x \in X} c^T x, \tag{1}$$

in the situation where the cost c is uncertain. If the coefficients of c are described by random variables, a classical approach replaces CO by its Value-at-Risk version:

$$\min_{x \in X} \mathrm{VaR}_\epsilon(c^T x), \tag{2}$$

where $\mathrm{VaR}_\epsilon(c^T x) = \inf\{t | P(c^T x \leq t) \geq 1 - \epsilon\}$. Hence, (2) tries to find the best solution cost that is guaranteed with a probability of $1 - \epsilon$. While VaR is very popular in the financial industry (Cornuejols and Tutuncu, 2006), it suffers from several drawbacks, among which: (i) the probability distributions of the random parameters are often impossible to describe with precision and (ii) the resulting optimization problems are very hard to solve exactly unless strong assumptions are made on the probability distributions. The hard computational complexity of problem (2) holds even when X is a polyhedron.

In contrast, robust optimization relaxes the knowledge about the components of c by only assuming that c belongs to a convex, closed, and bounded set $\mathbf{U} \subset \mathbb{R}^n$. The problem turns then to looking for a feasible solution that minimize its worst-case cost:

$$\min_{x \in X} \max_{c \in \mathbf{U}} c^T x. \tag{3}$$

This approach has several advantages, the first of which is its numerical complexity. Considering more particularly the budgeted uncertainty polytope

$$\mathbf{U}^\Gamma := \left\{ c \in \mathbb{R}^n : c_i = \bar{c}_i + \delta_i \hat{c}_i, 0 \leq \delta_i \leq 1, \sum \delta_i \leq \Gamma \right\}, \tag{4}$$

Bertsimas and Sim (2003) prove that problem

$$CO^\Gamma \quad \equiv \quad \min_{x \in X} \max_{c \in \mathbf{U}^\Gamma} c^T x \tag{5}$$

pertains to the same complexity class as CO. For general uncertainty polytopes $\mathbf{U}$, the linear programming relaxation of problem (3) can be reformulated as a linear program, while the discrete problem (3) can be reformulated

as a mixed-integer linear program. These approaches have made possible to solve a large variety of robust combinatorial optimization problems.

Another advantage of robust optimization lies in its less specific assumptions on the uncertainty. In many applications, it is not realistic or not possible to describe the uncertainty on the parameters by unique probability distributions. In these settings, it is sometimes more convenient to restrict our knowledge to the description of the possible outcomes by providing an uncertainty set $\mathbf{U}$ without providing any probability weight. There exists also intermediary models that assume partial knowledge on the distribution of the uncertain parameters. These models are usually cited as ambiguous chance constraints (Erdogan and Iyengar, 2006) or distributionally robust optimization problems (pioneered by Žáčková (1966)).

Among the large literature on robust optimization, researchers have proposed uncertainty sets that allow a robust constraint to approximate a chance constraint or an ambiguous chance constraint in the following sense: any solution to the robust constraint will be feasible for the original chance constraint or ambiguous chance constraint. In particular, Bertsimas and Sim (2004) have proved that uncertainty set $\mathbf{U}^{\Gamma}$ approximates an ambiguous chance constraint where the coefficients are described by bounded random perturbations that are only assumed to be symmetrically and independently distributed around their means. Their result yields the following relation between problems (2) and (3): the optimal solution of problem (3) is an upper bound for the optimal solution of problem (2) when the coefficients of c can be any random variables symmetrically and independently distributed around their means.

In a recent work, Poss (2013) mentions that the bound provided by Bertsimas and Sim is too conservative for feasible solutions with few non-zero components. In fact, for problems whose optimal solutions have small cardinalities compared to n, the probabilistic bounds from Bertsimas and Sim can be meaningless because the bounds would prescribe values for Γ that are greater than the solutions cardinalities. This motivates the introduction by Poss (2013) of a more general uncertainty model where the uncertainty set is replaced by a multifunction. Given a non-negative function $\gamma(x)$, the author defines variable budgeted uncertainty as the following multifunction:

$$\mathcal{U}^{\gamma}(x) := \left\{ c \in \mathbb{R}^n : c_i = \bar{c}_i + \delta_i \hat{c}_i, 0 \leq \delta_i \leq 1, \sum \delta_i \leq \gamma(x) \right\}. \qquad (6)$$

The uncertainty model (6) is tested on the robust knapsack problem showing

a reduction of the price-of-robustness by an average factor of 18% for little increase in computational time.

In this paper, we apply uncertainty model $\mathcal{U}^\gamma$ to optimization problems with uncertain costs, yielding

$$CO^\gamma \quad \equiv \quad \min_{x \in X} \max_{c \in \mathcal{U}^\gamma(x)} c^T x. \tag{7}$$

We study the properties and the computational complexity of CO^γ and present numerical examples showing the cost reduction obtained when using model CO^γ instead of model CO^Γ.

In the rest of the paper, $\|x\| = \sum_{i=1}^n |x_i|$ denotes the usual ℓ^1-norm. We also denote by $z(CO^\gamma)$ and $z(CO^\Gamma)$ the optimal solutions costs of problems CO^γ and CO^Γ, respectively. Finally, we assume throughout that the function γ is defined on $[0, n]$ and satisfies the following:

1. The function γ is non-decreasing.
2. For any $x^1, x^2 \in \{0,1\}^n$ such that $\|x^1\| = \|x^2\|$, $\gamma(x^1) = \gamma(x^2)$. Hence, we sometimes commit an abuse of notation and denote $\gamma(k)$ with $k \in \mathbb{Z}$.
3. $\gamma(\|x\|) \leq \|x\|$ for each $x \in X$.

These assumptions are natural in our context since the introduction of the function γ should modulate the size of the uncertainty set according to the cardinality of x. In particular, the functions prescribed by the probabilistic bounds from Bertsimas and Sim (2004) and introduced in Section 2.1 satisfy these assumptions. Nevertheless, some of these assumptions could be relaxed for proving specific results of this paper.

1.1. Contributions and structure of the paper

Our contributions lie in the study of the properties and solution methods of problem CO^γ as well as on its numerical assessment on two combinatorial optimization problems from the literature. Section 2 starts with the study of three important properties of model CO^γ. We explain in Section 2.1 how we can choose functions γ that provide better bounds for $\min_{x \in X} \mathrm{VaR}_\epsilon(c^T x)$ than the classical budgeted uncertainty model. These functions are deduced from the probabilistic bounds computed by Bertsimas and Sim (2004). We provide in Section 2.2 an asymptotically tight bound (asymptotic with respect to n) for the maximum cost reduction one can obtain when solving CO^γ instead of CO^Γ.

In Sections 3–5, we turn to the study of the computational complexity of CO^γ when $X \subset \{0,1\}^n$. Section 3 provides a mixed-integer linear reformulation for the problem, based on the dualization technique. The approach is particularly useful for $\mathcal{NP}$-hard optimization problems for which no efficient combinatorial algorithms exist. We study in the subsequent sections combinatorial approaches to CO^γ. In Section 4, we reduces problem CO^γ to solving a sequence of problems CO or its cardinality-constrained version, which depends on whether γ is affine or not. The approach from Section 5 focuses on problems CO for which efficient dynamic programming algorithms are available. We adapt these algorithms to solve the robust versions CO^Γ and CO^γ. The contributions of Sections 4 and 5 are summarized in Table 1 where it is assumed that γ has been chosen in accordance with the simplest probabilistic bound from Bertsimas and Sim (2004), see also Section 2.1; τ and $\tau^{\leq}$ denote the times for solving CO and its cardinality-constrained version, respectively. Recalling from Bertsimas and Sim (2003) that CO^Γ can be solved in $O(n\tau)$, we see from Table 1 that when g is affine the solution times of CO^Γ and CO^γ are of the same order of magnitude.

Table 1: Complexity of CO^γ

Dynamic programming	Properties of g	Complexity
No	affine	$O(n\tau)$
No	none	$O(n^2\tau^{\leq})$
Yes	integer-valued	$O(n^{3/2}\tau)$
Yes	none	$O(n^{5/2}\tau)$

In addition to these results, our dynamic programming approach from Section 5 also applies to the classical robust model CO^Γ studied by Bertsimas and Sim (2003). We show that when $\Gamma \in \mathbb{Z}$, problem CO^Γ can be solved in $O(n^{1/2}\tau)$, improving over the solution time of $O(n\tau)$ from Bertsimas and Sim (2003). This result extends to a large class of dynamic programming algorithms the ideas proposed by Klopfenstein and Nace (2008); Monaci et al. (2013) for the robust knapsack problem, by using a more general description of dynamic programming.

In Section 6, we present a numerical comparison of models CO^Γ and CO^γ on the shortest path problem and on the hop-constrained path diversified network design problem. The main objective of our experiments is to evaluate the cost reduction obtained by using the new model. As a by-product, we

also compare some of the solution methods proposed in Sections 3–5. Finally, we conclude the paper in Section 7.

2. Main properties

2.1. Probabilistic motivation

The main motivation of model $\mathcal{U}^\gamma$ comes from the probabilistic bounds computed by Bertsimas and Sim (2004) and extended to variable uncertainty by Poss (2013). We show below how the the simplest of these bounds, which has a nice analytical expression, extends to the problem of minimizing the value-at-risk. Namely, we show that the optimal solution of robust model CO^γ provides a less conservative solution for the stochastic problem (2) than model CO^Γ. Let $\tilde{c}_i = \bar{c}_i + \eta_i \hat{c}_i$ be the random variable associated with parameter c_i and suppose that η_i, $i = 1, \ldots, n$, are arbitrary random variables independently and symmetrically distributed in $[-1, 1]$.

Proposition 1. *Let* $\alpha(x) = (-2\ln(\epsilon)\|x\|)^{1/2}$. *It holds that*

$$\min_{x \in X} \mathrm{VaR}_\epsilon(\tilde{c}^T x) \leq \min_{x \in X} \max_{c \in \mathcal{U}^\alpha(x)} c^T x. \tag{8}$$

Proof. The result follows directly from:

$$\begin{aligned}
\min_{x \in X} \mathrm{VaR}_\epsilon(\tilde{c}^T x) = & \min \quad t \\
& \text{s.t.} \quad P(\tilde{c}^T x \leq t) \geq 1 - \epsilon \\
& \qquad\; x \in X \\
\leq & \min \quad t \\
& \text{s.t.} \quad c^T x \leq t, \qquad c \in \mathcal{U}^\alpha(x) \qquad (9) \\
& \qquad\; x \in X \\
= & \min_{x \in X} \max_{c \in \mathcal{U}^\alpha(x)} c^T x,
\end{aligned}$$

where inequality (9) follows from (Poss, 2013, Corollary 1). □

Inequality (8) also holds in the case of budgeted uncertainty where $\alpha(x) = \alpha(n) = \Gamma$ for all $x \in X$. Since $\alpha(x) \leq \Gamma$ for all $x \in X$, then

$$\min_{x \in X} \max_{c \in \mathcal{U}^\alpha(x)} c^T x \leq \min_{x \in X} \max_{c \in \mathbf{U}^\Gamma(x)} c^T x \tag{10}$$

holds and the approximation of problem (2) provided by CO^α is tighter than the one provided by CO^Γ. In fact, for problems for which the solution

has a small cardinality compared to n, model CO^{Γ} provides a very crude approximation of problem (2) because the value of Γ given by $\alpha(n)$ will almost always be above the cardinality of the solution, regardless to the value of ϵ. This is likely to happen, for instance, in network design problems defined on $G = (V, E)$ where the cardinality of the solutions is typically in $O(|V|)$ while the number of variables is in $O(|E|) = O(|V|^2)$. Model CO^{α} provides a tighter bound by reducing the value of Γ as the cardinality of x decreases.

Bertsimas and Sim (2004) mention that the probabilistic bound used in Proposition 1 is not very tight. For this reason, they introduce more complex bounds that provide tighter approximations of the probability $P(\tilde{c}^T x \leq t)$. The strongest of these bounds states that any vector x that satisfies the robust constraint $c^T x \leq t$ for all $c \in \mathbf{U}^{\Gamma}$, satisfies

$$P\left(\tilde{c}^T x > t\right) \leq B(n, \Gamma) = \frac{1}{2^n}\left((1-\mu)\binom{n}{\lfloor \nu \rfloor} + \sum_{l=\lfloor \nu \rfloor + 1}^{n} \binom{n}{l}\right), \tag{11}$$

where $\nu = (\Gamma + n)/2$ and $\mu = \nu - \lfloor \nu \rfloor$. Their experiments show that the bound provided by $B(n, \Gamma)$ is one order of magnitude smaller than the bound used in Proposition 1 for $n = 100$ and $n = 2000$.

To use this bound in the context of $\mathcal{U}^{\gamma}$, we need to solve the following problem in variable Γ

$$\min \Gamma \text{ s.t. } B(\|x\|, \Gamma) \leq \epsilon. \tag{12}$$

Problem (12) may not always have a solution for small values of $\|x\|$ and ϵ. This is not an issue in practice because one can see (Poss, 2013) that the problem has a solution for ϵ equal to 0.01 and 0.05 as soon as $\|x^*\|$ is greater than or equal to 8 and 5, respectively. We state below the pendant of Proposition 1 for this bound, with a proof similar to the one of Proposition 1. We define β as follows

$$\beta(x) = \begin{cases} \text{solution to (12) if the problem is feasible} \\ \|x\| \text{ otherwise.} \end{cases} \tag{13}$$

Proposition 2. *Let $\beta(x)$ be as in* (13)*. It holds that*

$$\min_{x \in X} \mathrm{VaR}_{\epsilon}(\tilde{c}^T x) \leq \min_{x \in X} \max_{c \in \mathcal{U}^{\beta}(x)} c^T x. \tag{14}$$

Choosing function γ according to the probabilistic guarantees computed by Bertsimas and Sim (2004) lead to using non-linear functions α or β. There exist different ways to avoid this non-linearity. In Section 3, we show that we can replace a non-linear function γ by a piece-wise approximation. A better idea, which is used in our numerical experiments, relies on solving first model CO^{Γ} with $\Gamma = \gamma(n)$ and then solving model $CO^{\partial\gamma}$ where $\partial\gamma$ is the affine function that approximates γ at the optimal solution of CO^{Γ}.

We study in the next section a theoretical upper bound on the cost reduction provided by model $\mathcal{U}^{\gamma}$.

2.2. Cost reduction

Once we have established that model $\mathcal{U}^{\gamma}$ provides the same probabilistic guarantee as model $\mathbf{U}^{\Gamma}$, it is natural to wonder how much cheaper the solution provided by model $\mathcal{U}^{\gamma}$ can be. We show below that when a technical condition holds, the ratio $\frac{z(CO^{\gamma})}{z(CO^{\Gamma})}$ is never smaller that $\frac{\gamma(\lceil\Gamma\rceil)-1}{\lceil\Gamma\rceil}$ and the bound is asymptotically tight with respect to n. The next result requires that γ satisfies the following condition:

$$\text{there exists a positive integer } \underline{k} \text{ such that } \frac{\gamma(k)-1}{k} \text{ is non-increasing for all } k \geq \underline{k}. \tag{15}$$

Functions α and β introduced in Propositions 1 and 2 satisfy condition (15) for small values of $\underline{k}$. Namely, one can see that $\frac{\alpha(k)-1}{k}$ is non-increasing as soon as $k \geq \frac{-2}{\ln(\epsilon)}$, while a numerical verification shows that β satisfies condition (15) as soon as (12) has a solution. The proof of the next theorem is provided in the appendix.

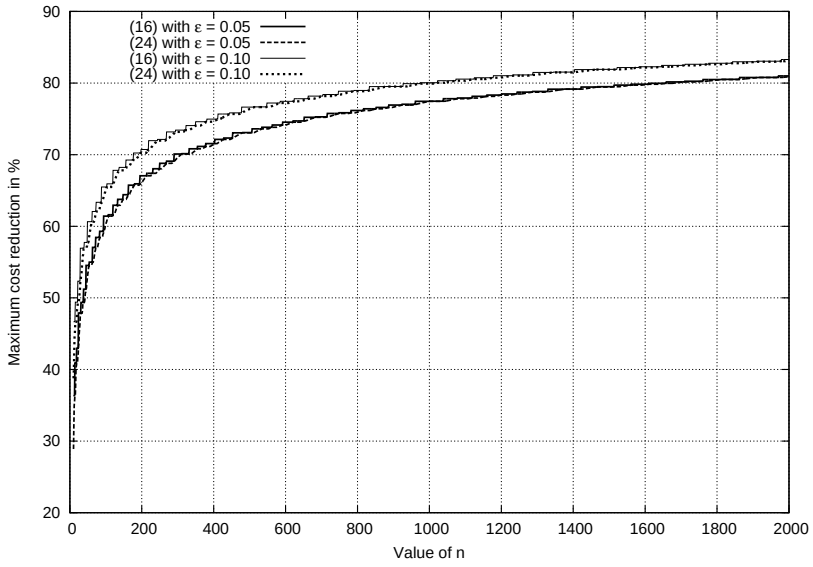
Theorem 1. *Let CO be any combinatorial optimization problem and $\Gamma = \gamma(n)$. Suppose that γ satisfies property* (15) *and that the optimal solution of CO^{γ} has a cardinality greater than or equal to $\underline{k}$. It holds that:*

$$\frac{z(CO^{\gamma})}{z(CO^{\Gamma})} \geq \frac{\gamma(\lceil\Gamma\rceil)-1}{\lceil\Gamma\rceil}. \tag{16}$$

Bound (16) *is asymptotically tight with respect to n for $\gamma = \alpha$, $\bar{c} = 0$, $\hat{c}_i = 1$ for each $i = 1, \ldots, n$, and $X \subset \{x, \|x\| \geq k\}$.*

It is much harder to verify analytically the tightness of (16) for function β given that the function does not have an analytical expression. Hence,

Figure 1: Maximal theoretical values for $\frac{z(CO^{\Gamma})-z(CO^{\beta})}{z(CO^{\Gamma})}$ expressed in percentage.



we illustrate numerically in Figure 1 the bounds provided by (16) and (A.8) for the cost reduction $\frac{z(CO^{\Gamma})-z(CO^{\beta})}{z(CO^{\Gamma})}$. Figure 1 suggests that bound (16) is very tight already for relatively small values of n. One can hardly expect to witness such cost reductions with practical problems since the theoretical bounds consider extreme cases where $\overline{c} = 0$ and $\hat{c}_i = 1$ for each $i = 1, \ldots, n$. Our numerical experiments from Section 6 realized on two combinatorial optimization problems with hundreds of variables exhibit cost reduction ranging from less than 1% up to 40%.

In the following three sections, we study the tractability of CO^{γ}. We provide in Section 3 a mixed-integer reformulation for CO^{γ} and we study the computational complexity of the problem in Sections 4 and 5. These approaches show that using model CO^{γ} instead of CO^{Γ} may not increase much the complexity of the resulting optimization problems. In particular when γ is affine, problem CO^{γ} belongs to the same complexity class as problem CO^{Γ}.

3. MILP reformulation

Robust combinatorial optimization problems under polyhedral uncertainty can be reformulated as mixed-integer linear programs by using linear programming duality (Ben-Tal and Nemirovski, 1999). Applying this technique to CO^γ leads to reformulations containing products between $\gamma(x)$ and the dual variables which can be difficult to handle, especially when γ is not affine. Hence, one ideally would like to consider only affine functions $\gamma(x)$ because this would lead to products of binary variables and real variables, easy to linearize with big-M coefficients. However, recall that some meaningful functions $\gamma(x)$, such as the ones proposed in Propositions 1 and 2, are not affine. To be able to approximate models defined by non-affine functions, we introduce an extension of $\mathcal{U}^\gamma$ where more than one function delimits the maximum number of simultaneous deviations. This allows us to approximates a non-affine function by an upper-bounding piece-wise affine function.

Let $\mathbf{g} : X \to \mathbb{R}^m$ be an affine mapping defined as

$$\mathbf{g}(x) = \mathbf{g}_0 + \sum_{i=1}^{n} \mathbf{g}_i x_i,$$

characterized by vectors $\mathbf{g}_0 \in \mathbb{R}^m$ and $\mathbf{g}_i \in \mathbb{R}^m$ for each $i = 1, \ldots, n$. We introduce below a multifunction delimited by $\mathbf{g}$:

$$\mathcal{U}^{\mathbf{g}}(x) := \left\{ c \in \mathbb{R}^n : c_i = \bar{c}_i + \delta_i \hat{c}_i, 0 \leq \delta_i \leq 1, \sum \delta_i \leq \mathbf{g}^j(x), j = 1 \ldots, m \right\}. \tag{17}$$

Multifunction $\mathcal{U}^{\mathbf{g}}$ with affine mapping $\mathbf{g} : X \to \mathbb{R}^m$ allows us to approximate $\mathcal{U}^\gamma$ when γ is non-affine. If $\mathbf{g}^j(x) \geq \gamma(x)$ holds for $j = 1, \ldots, m$, $\mathcal{U}^\gamma(x) \subseteq \mathcal{U}^{\mathbf{g}}(x)$ for any $x \in X$ and $\mathcal{U}^{\mathbf{g}}$ provides a conservative approximation of $\mathcal{U}^\gamma$.

Lemma 1. *Let $\mathbf{g} : X \to \mathbb{R}^m$ be an affine mapping of x such that $\mathbf{g}^j(x) \geq \gamma(x)$ for $j = 1, \ldots, m$ and all $x \in \{0,1\}^n$. It holds that*

$$\min_{x \in X} \max_{c \in \mathcal{U}^\gamma(x)} c^T x \leq \min_{x \in X} \max_{c \in \mathcal{U}^{\mathbf{g}}(x)} c^T x.$$

The next result shows how to handle the upper approximation provided by $\mathbf{g}$.

Theorem 2. *Let* $\mathbf{g} : X \to \mathbb{R}^m_+$ *be a non-negative affine mapping of* x*. Then,* $\min_{x\in X} \max_{c\in\mathcal{U}^{\mathbf{g}}(x)} c^T x$ *is equivalent to*

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} c_i x_i + \sum_{j=1}^{m} \left(\mathbf{g}_0^j z_j + \sum_{i=1}^{n} \mathbf{g}_i^j w_{ji} \right) + \sum_{i=1}^{n} p_i \\
s.t. \quad & \sum_{j=1}^{m} z_j + p_i \geq \hat{c}_i x_i, && i = 1, \ldots, n, \\
& w_{ji} - z_j \geq -\max_j(\hat{c}_j)(1 - x_i), && i = 1, \ldots, n, j = 1, \ldots, m, \\
& p, w, z \geq 0, \\
& x \in \{0, 1\}^n.
\end{aligned}
$$

Proof. Using an epigraph formulation, $\min_{x\in X} \max_{c\in\mathcal{U}^{\mathbf{g}}(x)} c^T x$ can be rewritten as

$$
\begin{aligned}
\min \quad & t \\
\text{s.t.} \quad & c^T x \leq t, \qquad c \in \mathcal{U}^{\mathbf{g}}(x) \\
& x \in X.
\end{aligned}
$$

The results follows from applying (Poss, 2013, Theorem 1) to the epigraph reformulation. □

The MILP formulation from Theorem 2 can be an efficient way to handle problems for which the deterministic version CO is $\mathcal{NP}$-hard in the strong sense. Computational results from Poss (2013) on the robust knapsack problem suggest that the formulation above is only slightly more complex than the dualization of the classical model using uncertainty set $\mathbf{U}^{\Gamma}$ and that the coefficients $\max_j(\hat{c}_j)$ may not decrease the quality of the bound provided by the linear relaxation. We further show in the computational results from Section 6.2 that this method can be the most efficient way to solve model CO^{γ} for hard network design problems. Nevertheless, when CO can be solved in polynomial time or pseudo-polynomial time, the next sections describe more efficient approaches than Theorem 2.

4. Sequence of deterministic problems

We study in this section the computational complexity of CO^{γ}. We show that when γ is affine, CO^{γ} belongs to same complexity class as the original

problem CO. We show then that for more general functions γ, CO^γ belongs to same complexity class as the cardinality-constrained version of CO.

We first consider the case where γ is an affine function of x:

$$\gamma(x) = \gamma_0 + \sum_{i=1}^{n} \gamma_i x_i. \tag{18}$$

We see next that surprisingly, the computational complexity of CO^γ for an affine function γ is essentially the same as the complexity of CO. The result below is an extension of (Bertsimas and Sim, 2003, Theorem 3) to the case of variable budgeted uncertainty. The proof is omitted because it is almost identical to the proof of (Bertsimas and Sim, 2003, Theorem 3). Without loss of generality, we suppose below that the coefficients the variables are ordered by decreasing order of $\hat{c}_i$ and we use the notation $\hat{c}_{n+1} = 0$.

Theorem 3. *When γ is an affine function, CO^γ can be solved by solving the $n+1$ nominal problems:*

$$\gamma_0 \hat{c}_l + \min_{x \in X} \sum_{i=1}^{n} (\overline{c}_i + \gamma_i \hat{c}_l) x_i + \sum_{i=1}^{l} (\hat{c}_i - \hat{c}_l) x_i, \tag{19}$$

for $l = 1, \ldots, n+1$ and taking the cheapest optimal solution.

The only difference between problem (19) and the problems solved in (Bertsimas and Sim, 2003, Theorem 3) lies in the costs coefficients of x. In problem (19), these coefficients depend on the components γ_i of the budget function γ while they are independent of Γ in (Bertsimas and Sim, 2003, Theorem 3).

We turn now to non-affine budget functions, such as the functions used in Propositions 1 and 2. We show below show that CO^γ is strongly related to the cardinality-constrained version of CO

$$CO_k^{\leq} \quad \equiv \quad \min_{x \in X, \|x\| \leq k} c^T x. \tag{20}$$

For a large class of combinatorial optimization problems, such as the knapsack problem or the shortest path problem, the original problem and its cardinality constrained version stay in the same complexity class. We show in the next results how solving CO^γ amounts to solve robust versions of $CO_k^{\leq}$.

Theorem 4. *Problem CO^γ can be solved by solving the n nominal problems:*

$$\min_{x\in X,\|x\|\le k} \max_{c\in \mathbf{U}^{\gamma(k)}} c^T x,$$

for $k = 1,\dots,n$ and taking the cheapest optimal solution.

Proof. We must prove that following equality holds

$$\min_{x\in X} \max_{c\in \mathcal{U}^\gamma(x)} c^T x = \min_{k=1,\dots,n} \min_{x\in X,\|x\|\le k} \max_{c\in \mathbf{U}^{\gamma(k)}} c^T x. \tag{21}$$

$\le$: For every $k = 1,\dots,n$, the following holds:

$$\begin{aligned} \min_{x\in X,\|x\|\le k} \max_{c\in \mathbf{U}^{\gamma(k)}} c^T x &\ge \min_{x\in X,\|x\|\le k} \max_{c\in \mathcal{U}^\gamma(x)} c^T x \qquad (22)\\ &\ge \min_{x\in X} \max_{c\in \mathcal{U}^\gamma(x)} c^T x, \end{aligned}$$

where inequality (22) follows from the fact that γ is non-decreasing.

$\ge$: Let x^* be the optimal solution of the left-hand-side of (21) and denote $k^* = \|x^*\|$. The following holds

$$\begin{aligned} \min_{x\in X} \max_{c\in \mathcal{U}^\gamma(x)} c^T x = \min_{x\in X} \max_{c\in \mathcal{U}^\gamma(x^*)} c^T x &\ge \min_{x\in X,\|x\|=k^*} \max_{c\in \mathbf{U}^{\gamma(k^*)}} c^T x \qquad (23)\\ &\ge \min_{x\in X,\|x\|\le k^*} \max_{c\in \mathbf{U}^{\gamma(k^*)}} c^T x, \end{aligned}$$

where $\mathbf{U}^{\gamma(k^*)} = \mathcal{U}^\gamma(x^*)$ was used to obtain inequality (23). □

Applying sequentially Theorem 4 and Theorem 3, we obtain that CO^Γ is polynomially solvable when $CO_k^\le$ is polynomially solvable.

Corollary 1. *Problem CO^γ can be solved by solving the $n(n+1)$ nominal problems:*

$$\gamma(k)\hat{c}_l + \min_{x\in X,\|x\|\le k} \sum_{i=1}^{n} (c_i + \gamma_i \hat{c}_l) x_i + \sum_{i=1}^{l} (\hat{c}_i - \hat{c}_l) x_i,$$

for $k = 1,\dots,n, l = 1,\dots,n+1$ and taking the cheapest optimal solution.

We show below that a cardinality constrained optimization problem can also be solved as a problem CO^{γ}. Hence, polynomially solvable optimization problems that become $\mathcal{NP}$-hard when adding a cardinality constraint, such as the minimum cut problem (Bentz et al., 2009), lead to $\mathcal{NP}$-hard problems CO^{γ} in general.

Theorem 5. *Problem $CO_k^{\leq}$ can be solved by solving CO^{γ} with*

$$\gamma(x) = \begin{cases} 0 & \text{if } \|x\| \leq k \\ 1 & \text{if } \|x\| > k \end{cases}, \tag{24}$$

and $\hat{c}_i = \sum_i c_i + 1$ for each $i = 1, \ldots, n$.

Proof. Consider an instance of $CO_k^{\leq}$ and create an associate instance of CO^{γ} by setting $\hat{c}_i = \sum_i c_i + 1$ for each $i = 1, \ldots, n$, and defining γ as in (24). Let x^* and c^* be, respectively, the optimal solution and the optimal solution cost to the instance of CO^{γ}. If $c^* \leq \sum_i c_i$, then x^* is also an optimal solution for the instance of $CO_k^{\leq}$. If $c^* > \sum_i c_i$, this means that the original instance of (20) is infeasible. □

Corollary 1 and Theorem 5 show that, given a combinatorial optimization problem defined by optimizing a linear cost function over the feasibility set X, problems CO^{γ} and $CO_k^{\leq}$ belong to the same complexity class. This equivalence does not hold if γ is restricted to affine functions since Theorem 3 states that the resulting problem CO^{γ} belongs to the same complexity class as CO.

5. Dynamic Programming

We propose in this section an alternative method for solving problems CO^{Γ} and CO^{γ}. Given a combinatorial optimization problem CO and a dynamic programming (DP) algorithm A to solve CO, our objective is to modify A to solve problems CO^{Γ} and CO^{γ} by taking into account the min max structure of these problems inside algorithm A. Hence, unlike the results of Section 4, the results of this section also provide faster algorithms for solving the classical robust optimization problem CO^{Γ}. Notice that not all DP algorithms can be used in our framework and, in fact, DP is a methodology that goes far beyond the scope of the static combinatorial optimization problems studied in this paper (Sniedovich, 2011). We define precisely Section 5.1 the type of functional equations we consider in this paper, which include many

well-known DP algorithms such as the Bellman-Ford algorithm the DP approach to the knapsack problem. We show then in Sections 5.2 and 5.3 how the aforementioned DP algorithms extend naturally to robust problems CO^{Γ} and CO^{γ}.

For CO^{Γ}, we explain next why this provides algorithms with better worst-time complexity than (Bertsimas and Sim, 2003, Theorem 3) when Γ is an integer chosen in accordance with Propositions 1 or 2. Consider a specific problem CO^{Γ} that can be solved by a DP algorithm A in $O(\tau)$. Theorem 3 from Bertsimas and Sim (2003) states that CO^{Γ} can be solved by employing $n+1$ times Algorithm A yielding a total time of $O(n\tau)$. We show below that if A is a DP algorithm, it can be modified to solve CO^{Γ} in $O(\Gamma\tau)$ instead. Guided by the probabilistic results of Bertsimas and Sim (2004), natural choices for Γ are of the order of $n^{1/2}$ or below. In this case, the running time of the modified A is $O(n^{1/2}\tau)$. Similarly, we show that solving CO^{γ} by the DP algorithm requires to use another modified version of Algorithm A with a time of $O(n^{3/2}\tau)$.

5.1. Problem CO

Consider an optimization problem CO with cost function c and suppose that there exists a DP algorithm A to solve CO. Hence, CO can be reformulated over a finite state space $\mathcal{S}$, which contains at least two special states: 0 is the initial state and N is the final state. For some problems, there exists several initial states, in which case $\mathcal{O} \subset \mathcal{S}$ denotes the set of initial states. The main idea of many DP algorithms is to fix variables x_i to 1 iteratively, using a recurrence loop. The optimal cost of state s is denoted by $F(s)$ while $F(N)$ denotes the optimal solution cost of CO. We denote by $q(s)$ the set of variables that can be fixed to 1 at state s and by $p(s,i) \in \mathcal{S}$ the previous state from s when choosing variable $i \in q(s)$. Set $q(s)$ may also include a dummy element d which means that the optimal choice at state s is that no variable is fixed to 1. Algorithm A solves problem CO by computing $F(N)$ via the functional equation below:

$$\begin{cases} F(s) &= \min\limits_{i \in q(s)} \{F(p(s,i)) + c_i\}, & s \in \mathcal{S} \backslash \mathcal{O} \\ F(s) &= 0, & s \in \mathcal{O}, \end{cases} \tag{25}$$

and all states that are not in $\mathcal{O}$ are initialized with the value $+\infty$. Let $Q = \max_{s \in \mathcal{S}} |q(s)|$. In the worst-case, functional equation (25) requires to visit all spaces of $\mathcal{S}$, yielding a total time of $O(Q|\mathcal{S}|)$. This approach leads to

an intuitive interpretation in term of shortest path in a directed and acyclic graph. Let $G = (V, A)$ be a directed graph with $V = \mathcal{S}$ and $A = \{(p(s,i), s) : s \in \mathcal{S}, i \in q(s)\}$ where the cost of arc $(p(s,i), s)$ is equal to c_i. Clearly, the value of the shortest path from the states in $\mathcal{O}$ to state N is equal to the optimal solution of problem CO. The interpretation of CO as a shortest path problem in the directed and acyclic graph defined by the state space is used in the next two sections to extend functional equation (25) to the robust cases.

We provide below two classical examples of DP algorithms that follow equation (25): the Bellman-Ford algorithm and the DP approach to the knapsack problem. Other applications can be found in the literature, such as the constrained shortest path problem (Righini and Salani, 2008), the traveling salesman problem (Held and Karp, 1962), the capital budgeting problem (Cornuejols and Tutuncu, 2006), and scheduling problems (P. Brucker, 2006). Notice, however, that this is not the case for all DP algorithms used in combinatorial optimization, since, for instance, the Floyd-Warshall algorithm does not select individual arcs iteratively.

Example 1 (Bellman-Ford algorithm)**.** *This algorithms looks for the shortest path between two nodes (denoted o and t) in a graph $G = (V, A)$ that does not contain negative cycles. Here, the state space is $\mathcal{S} = \{(i,k) : i \in V,\ k = 0, \ldots, |V| - 1\}$, with initial state space $\mathcal{O} = \{(o,k) : k = 0, \ldots, |V| - 1\}$ and the final state N is $(t, |V| - 1)$. Then, the possible choices at each state are $q(i,k) = \{(j,i) \in A\}$ if $k > 0$ where the predecessors are $p(i,k,(j,i)) = (j, k-1)$, and $q(i,0) = \emptyset$. Notice that while the original version of this algorithm (Bellman, 1958) relies on functional equation* (25)*, the faster version (see for instance (Bang-Jensen and Gutin, 2007)) has replaced the functional equation by a smaller loop which is no longer a DP algorithm. Nevertheless, the DP version can be used to solve the shortest path problem with hop constraint, which is not the case of the non-DP algorithm.*

Example 2 (Knapsack problem)**.** *We recall here the DP algorithm for the knapsack problem with n items, total capacity b, and individual weights w_i. The state space is $\mathcal{S} = \{(i,k) : i = 0, \ldots, n,\ k = 0, \ldots, b\}$, the initial state space is $\mathcal{O} = \{(0,k) : k = 0, \ldots, b\}$, and the final state is (n, b). The possible choices at each state are $q(i,k) = \{i\} \cup \{d\}$ and the predecessors are $p(i,k,i-1) = (i-1, k-w_i)$ and $p(i,k,d) = (i-1,k)$.*

5.2. Problem CO^{Γ}

We prove below that the solution cost of CO^{Γ} can be computed using a functional equation that is similar to equation (25).

Theorem 6. *Consider an instance of problem CO that can be solved in $O(\tau)$ by using functional equation* (25)*. Then, its robust version CO^{Γ} can be solved in $O(\Gamma\tau)$.*

Proof. We consider first the case where $\Gamma \in \mathbb{Z}$. Recalling the graph analogy, the optimal solution cost of CO^{Γ} is now equal to the minimum cost shortest path problem in G from $\mathcal{O}$ to N where the worst Γ arcs have extreme costs. Let us denote by $SPP(s,\alpha)$ the minimum cost shortest path problem in G from $\mathcal{O}$ to s where the worst α arcs have extreme costs. We show next by induction that $SPP(s,\alpha)$ is equal to $F(s,\alpha)$ defined by the functional equation

$$\begin{cases} F(s,\alpha) = \min\limits_{i\in q(s)}\{\max(F(p(s,i),\alpha)+\overline{c}_i, F(p(s,i),\alpha-1)+\overline{c}_i+\hat{c}_i)\}, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad s\in\mathcal{S}\backslash\mathcal{O}, 1\le\alpha\le\Gamma, \\ F(s,0) = \min\limits_{i\in q(s)}\{F(p(s,i),0)+\overline{c}_i\}, \qquad\qquad s\in\mathcal{S}\backslash\mathcal{O}, \\ F(s,\alpha) = 0, \qquad\qquad\qquad\qquad\qquad 0\le\alpha\le\Gamma, s\in\mathcal{O}. \end{cases} \tag{26}$$

First, $SPP(s,\alpha) = 0 = F(s,\alpha)$ for $0 \le \alpha \le \Gamma$ and each $s \in \mathcal{O}$. Suppose then that $F(p(s,i),\alpha) = SPP(p(s,i),\alpha)$ and $F(p(s,i),\alpha-1) = SPP(p(s,i),\alpha-1)$ for each $i \in q(s)$ and consider $F(s,\alpha)$. To compute $SPP(s,\alpha)$, we must consider adding the arc $(p(s,i),s)$ to the shortest path to $p(s,i)$, for each $i \in q(s)$, using the worst α extreme costs, and selecting the cheapest solution. Notice that when adding arc $(p(s,i),s)$ to an existing path, the arc contributes to the total cost of the current path either with its cost c_i or with its extreme cost $c_i+\hat{c}_i$, depending on which is worse. In the first case, the rest of the path considers the α worst extreme costs, while in the second case only $\alpha-1$ arcs have extreme costs in the rest of the path. In any case, $SPP(s,\alpha)$ chooses among the predecessor yielding the smallest total cost. This is exactly what describes the right-hand-side of the first equation in (26). The second equation in (26) is obtained when no extreme costs can be considered at s because $\alpha = 0$. Hence, $F(s,\alpha) = SPP(s,\alpha)$.

When $\Gamma \notin \mathbb{Z}$, we must consider two extreme cost values for each variable i, $\hat{c}_i$ and $(\Gamma - \lfloor\Gamma\rfloor)\hat{c}_i$. Accordingly, we add one dimension to the dynamic programming state space, equal to 1 if the extreme cost related to $\Gamma - \lfloor\Gamma\rfloor$

has been considered already in the computation of the shortest path, and equal to 0 otherwise. The optimal solution cost for this new model is denoted $F(N, \lfloor \Gamma \rfloor, 1)$, which is computed by the functional equation

$$\begin{cases} F(s,\alpha,1) &= \min\limits_{i\in q(s)}\{\max(F(p(s,i),\alpha,1)+\overline{c}_i, F(p(s,i),\alpha-1,1)+\overline{c}_i+\hat{c}_i, \\ & \quad F(p(s,i),\alpha,0)+\overline{c}_i+(\Gamma-\lfloor\Gamma\rfloor)\hat{c}_i)\}, \qquad s\in\mathcal{S}\backslash\mathcal{O}, 1\le\alpha\le\lfloor\Gamma\rfloor \\ F(s,\alpha,0) &= \min\limits_{i\in q(s)}\{\max(F(p(s,i),\alpha,0)+\overline{c}_i, F(p(s,i),\alpha-1,0)+\overline{c}_i+\hat{c}_i)\}, \\ & \quad s\in\mathcal{S}\backslash\mathcal{O}, 1\le\alpha\le\lfloor\Gamma\rfloor \\ F(s,0,1) &= \min\limits_{i\in q(s)}\{\max(F(p(s,i),0,1)+\overline{c}_i, F(p(s,i),0,0)+\overline{c}_i+(\Gamma-\lfloor\Gamma\rfloor)\hat{c}_i)\}, \\ & \quad s\in\mathcal{S}\backslash\mathcal{O} \\ F(s,0,0) &= \min\limits_{i\in q(s)}\{F(p(s,i),0,0)+\overline{c}_i\}, \qquad s\in\mathcal{S}\backslash\mathcal{O} \\ F(s,\alpha,f) &= 0, \qquad 0\le\alpha\le\lfloor\Gamma\rfloor, f\in\{0,1\}, s\in\mathcal{O} \end{cases} \tag{27}$$

The number of states in equations (26) and (27) are equal to $\Gamma|\mathcal{S}|$ and $2\Gamma|\mathcal{S}|$, respectively, and the computation of each state involves $O(Q)$ operations, with $Q = \max_{s\in\mathcal{S}}|q(s)|$. Therefore, the total solution times of the algorithms is $O(\Gamma Q|\mathcal{S}|) = O(\Gamma\tau)$. □

Let us illustrate Theorem 6 when Γ is chosen according to a probabilistic guarantee of $1-\epsilon$. The pendent of Proposition 1 in the classical case where Γ is independent of x imposes that $\Gamma = \left(-2\ln(\epsilon)n\right)^{1/2}$. In this case, the solution time of Theorem 6 becomes $O(n^{1/2}\tau)$. Using betters bounds for Γ, see (Bertsimas and Sim, 2004, Theorem 3), the solution time can be further decreased.

5.3. Problem CO^γ

Theorem 6 can be extended to CO^γ by keeping track of the size of the paths in G. Here, allowing fractional values of γ increases the computational complexity of the algorithm, contrasting with the case of CO^Γ. In the next result, we use the notation $\Gamma' = \min(n, \max\limits_{k=0,\dots,n}\lfloor\gamma(k)\rfloor)$.

Theorem 7. *Consider an instance of problem CO that can be solved in $O(\tau)$ by using functional equation (25). If $\gamma(k)\in\mathbb{Z}$ for each $k=0,\dots,n$, then CO^γ can be solved in $O(n\Gamma'\tau)$. Otherwise, CO^γ can be solved in $O(n^2\Gamma'\tau)$.*

Proof. We consider first the case where $\gamma(k)\in\mathbb{Z}$ for each $k=0,\dots,n$. Recalling once more the graph analogy, we compute shortest paths in G

while keeping track of the path lengths. Hence, we compute $F(s,\gamma(k),k)$ via

$$\begin{cases} F(s,\alpha,k) &= \min\limits_{i\in q(s)}\{\max(F(p(s,i),\alpha,k-1)+\overline{c}_i, F(p(s,i),\alpha-1,k-1)+\overline{c}_i+\hat{c}_i)\}, \\ & \qquad\qquad s\in\mathcal{S}\backslash\mathcal{O}, 1\leq\alpha\leq\Gamma', 1\leq k\leq n, \\ F(s,0,k) &= \min\limits_{i\in q(s)}\{F(p(s,i),0,k-1)+\overline{c}_i\}, \qquad s\in\mathcal{S}\backslash\mathcal{O}, 1\leq k\leq n, \\ F(s,\alpha,0) &= 0, \qquad s\in\mathcal{S}\backslash\mathcal{O}, 0\leq\alpha\leq\Gamma', \\ F(s,\alpha,k) &= 0, \qquad 0\leq\alpha\leq\Gamma', 0\leq k\leq n, s\in\mathcal{O}, \end{cases}$$

for each $l=k,\ldots,n$. This computation is done in $O(n\Gamma' Q|\mathcal{S}|)=O(n\Gamma'\tau)$ and the optimal solution to CO^γ is given by

$$\min_{k=1,\ldots,n} F(s,\gamma(k),k). \tag{28}$$

The difficulty when considering real-valued function γ is that the fractional part of $\gamma(k)$ may be different for each $k=1,\ldots,n$. Hence, we must solve a sequence of functional equations associated to the different fractional values of $\gamma(k)$. Namely, let $r_l=\gamma(l)-\lfloor\gamma(l)\rfloor$ for each $l=1,\ldots,n$. Then, for each $l=1,\ldots,n$, we compute $F^l(s,\alpha,l,1)$ via

$$\begin{cases} F^l(s,\alpha,k,1) &= \min\limits_{i\in q(s)}\{\max(F^k(p(s,i),\alpha,k-1,1)+\overline{c}_i, F^k(p(s,i),\alpha-1,k-1,1)+\overline{c}_i+\hat{c}_i, \\ & \quad F^k(p(s,i),\alpha,k-1,0)+\overline{c}_i+r_l\hat{c}_i)\}, \qquad s\in\mathcal{S}\backslash\mathcal{O}, 1\leq\alpha\leq\Gamma', 1\leq k\leq n, \\ F^l(s,\alpha,k,0) &= \min\limits_{i\in q(s)}\{\max(F^l(p(s,i),\alpha,k-1,0)+\overline{c}_i, F^l(p(s,i),\alpha-1,k-1,0)+\overline{c}_i+\hat{c}_i)\}, \\ & \qquad\qquad s\in\mathcal{S}\backslash\mathcal{O}, 1\leq\alpha\leq\Gamma', 1\leq k\leq n, \\ F^l(s,0,k,1) &= \min\limits_{i\in q(s)}\{\max(F^l(p(s,i),0,k-1,1)+\overline{c}_i, F^l(p(s,i),0,k-1,0)+\overline{c}_i+r_l\hat{c}_i)\}, \\ & \qquad\qquad s\in\mathcal{S}\backslash\mathcal{O}, 1\leq k\leq n, \\ F^l(s,0,k,0) &= \min\limits_{i\in q(s)}\{F^l(p(s,i),0,k-1,0)+\overline{c}_i\}, \qquad s\in\mathcal{S}\backslash\mathcal{O}, 1\leq k\leq n, \\ F^l(s,\alpha,0,f) &= 0, \qquad s\in\mathcal{S}\backslash\mathcal{O}, 0\leq\alpha\leq\Gamma', f\in\{0,1\}, \\ F^l(s,\alpha,k,f) &= 0, \qquad 0\leq\alpha\leq\Gamma', f\in\{0,1\}, 1\leq k\leq n, s\in\mathcal{O}. \end{cases}$$

For each $l=1,\ldots,n$, computing $F^l(s,\alpha,l,1)$ is done in $O(n\Gamma'\tau)$ yielding a total time of $O(n^2\Gamma\tau)$. The optimal solution to CO^γ is given by

$$\min_{l=1,\ldots,n} F^l(s,\lfloor\gamma(l)\rfloor,l,1).$$

□

6. Numerical experiments

In this section, we assess numerically our new model on two combinatorial optimization problems from the literature: the shortest path problem and the survivable network design problem. The main objective of our computational experiments is to evaluate the reduction of the *price of robustness* obtained from using CO^β instead of CO^Γ where β is given by Proposition 2 and $\Gamma = \beta(n)$ is chosen accordingly. This cost reduction is formally defined as $\frac{CO^\Gamma - CO^\beta}{CO^\Gamma}$. As a byproduct, we illustrate numerically the effiency of some of the solution methods proposed in Sections 3–5.

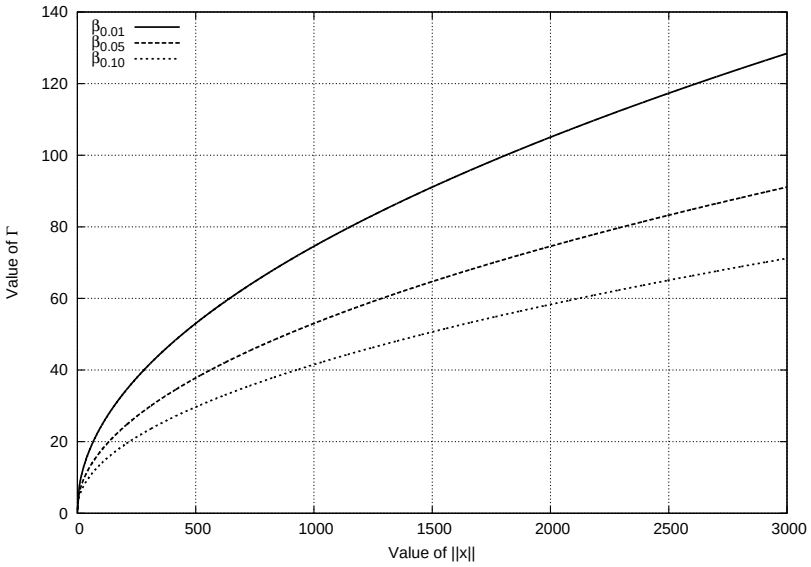
We have seen in Sections 3 and 4 that CO^γ is much easier to solve when γ is an affine function. For this reason, our experiments with variable uncertainty focus on an approach in two steps. First, we solve model CO^Γ with $\Gamma = \beta(n)$ and denote its optimal solution by x^*. Then, defining $\partial\beta$ as the affine function that approximates β at $\|x^*\|$, we solve model $CO^{\partial\beta}$. For the shortest path problem, we also evaluate the value of the bound provided by $UB(\mathcal{U}^\beta)$ equal to $\max_{\mathcal{U}^\beta} c^T x^*$.

The value of β was computed numerically with a precision of three decimals by solving problem (12). We could easily compute β for k going up to 3000, which was enough for our computational experiments. These values are illustrated in Figure 2 for three choices of ϵ. The experiments have been run a computer equipped with a processor Intel Core i7-3520M at 2.90 GHz and 8 GB of RAM. All algorithms were coded in JAVA and CPLEX (2013) was used as the MIP solver.

6.1. The shortest path problem

We illustrate in this section the gain of the new model CO^β compared to the classical model CO^Γ on the $o-t$ shortest path problem defined in graph $G = (V \cup \{o,t\}, A)$ that may contain cycles and where all costs are positive. The problem can be modeled as as follows. Let $|V| = n$ and $|A| = m$, and $c : A \to \mathbb{R}_+$ and $\hat{c} : A \to \mathbb{R}_+$ be two cost functions. Let also $b \in \mathbb{R}^n$ be such that $b_o = -1$, $b_t = 1$, and $b_i = 0$ for each $i \in V$, $\delta^+(i) = \{j \in V : \exists (i,j) = a \in A\}$, and $\delta^-(i) = \{j \in V : \exists (j,i) = a \in A\}$.

Figure 2: The value of β for $\epsilon = 0.01, 0.05$, and 0.10.



The formulation follows:

$$
\begin{aligned}
& & \min \quad & c^T x & & & (29)\\
(SP) & & \text{s.t.} \quad & \sum_{a \in \delta^-(i)} x_a - \sum_{a \in \delta^+(i)} x_a = b_i & & i \in V \cup \{o, t\}\\
& & & x_a \in \{0, 1\} & & a \in A.
\end{aligned}
$$

The formulation above can be adapted to cost varying in $\mathbf{U}^\Gamma$ and $\mathcal{U}^\beta$ by replacing objective function (29) with $\min \max_{c \in \mathbf{U}^\Gamma} c^T x$ and $\min \max_{c \in \mathcal{U}^\beta} c^T x$, respectively. Our experiments are based on four real road networks used in Zhan and Noon (1998), NE1, AL1, MN1, and IA1, whose main characteristics are reminded in Table 2. We generate $\hat{c}$ as Bertsimas and Sim (2003): for each $a \in A$, $\hat{c}_a$ is uniformly distributed in $[0, 8c_a]$.

While (SP) can handle relatively large deterministic problems, its dualized reformulations are ineffective for solving robust problems because the resulting matrix is not totally unimodular anymore and solving the problems enumerates many branch-and-bound nodes. Instead, using (Bertsimas and Sim, 2003, Theorem 3) and Theorem 3, models $\mathbf{U}^\Gamma$ and $\mathcal{U}^{\partial\beta}$ can be solved

by solving $n+1$ shortest path problems, which can in turn be solved by Dijkstra's algorithm. Unreported results show that solving the robust versions of (SP) by dualizing the mathematical program is orders of magnitudes slower than using $n+1$ Dijkstra's algorithm. Using Theorem 4, model $\mathcal{U}^\beta$ can also be solved to optimality by solving n cardinality-constrained robust shortest path problems which are solved by using the robust version of the Bellman-Ford algorithm described in Section 5.2.

Table 2: Characteristics of the networks taken from Zhan and Noon (1998).

Network name				Arc Length		
(abbreviation)	$\|V\|$	$\|A\|$	Arc/Node ratio	Maximum	Mean	Stnd. Dev.
Nebraska (NE1)	523	1646	3.14	0.874764	0.215551	0.142461
Alabama (AL1)	842	2506	2.98	0.650305	0.128870	0.114031
Minnesota (MN1)	951	2932	3.08	0.972436	0.175173	0.132083
Iowa (IA1)	1003	2684	2.68	0.573768	0.119900	0.113719

We provide in Table 3 the solution costs for each model, expressed as percentages of the deterministic solution cost. For each value of ϵ ranging from 0.01 to 0.1, we generate 5 instances and report the geometric averages. One can see from Table 3 that the cost of model $\mathbf{U}^\Gamma$ is identical for all $\epsilon \leq 0.8$. This is due to the fact that $\beta_\epsilon(2684)$ is greater than the cardinality of the optimal solution for all $\epsilon \leq 0.8$. Hence, choosing $\Gamma = \beta_\epsilon(2684)$, the optimal solution of model $\mathbf{U}^\Gamma$ is identical to the solution of the problem where all costs take simultaneously their maximum values with probability 1.

The solution times are very similar for models $\mathbf{U}^\Gamma$ and $\mathcal{U}^{\partial\beta}$ and all instances could be solved within a few seconds. Recall, however, that model $\mathcal{U}^{\partial\beta}$ requires to solve first model $\mathbf{U}^\Gamma$ to obtain the vector x^* at which β is linearized. Hence, the total solution time for model $\mathcal{U}^{\partial\beta}$ is more or less twice the solution time for model $\mathbf{U}^\Gamma$. Model $\mathcal{U}^\beta$ is much harder to solve since several minutes are needed to solve the problem to optimality. Moreover, we see from Table 3 that the additional gain of model $\mathcal{U}^\beta$ over model $\mathcal{U}^{\partial\beta}$ is very little in comparison to the gain of model $\mathcal{U}^{\partial\beta}$ over model $\mathbf{U}^\Gamma$. For this reason, we restrict our attention to the comparison of models $\mathbf{U}^\Gamma$ and $\mathcal{U}^{\partial\beta}$ in what follows. We report in Figure 3 the cost reduction when using model $\mathcal{U}^{\partial\beta}$ instead of model $\mathbf{U}^\Gamma$, expressed in percentage of the optimal solution cost of model $\mathbf{U}^\Gamma$.

Table 3: Price of robustness in % of deterministic cost for instance IA1.

ϵ	$\mathbf{U}^\Gamma$	$UB(\mathcal{U}^\beta)$	$\mathcal{U}^{\partial\beta}$	$\mathcal{U}^\beta$
0.01	408	319	311	310
0.02	408	306	297	295
0.03	408	297	287	285
0.04	408	289	279	278
0.05	408	283	273	271
0.06	408	278	267	265
0.07	408	273	261	260
0.08	408	268	257	255
0.09	407	264	252	251
0.10	407	260	248	246

6.2. The survivable network design problem

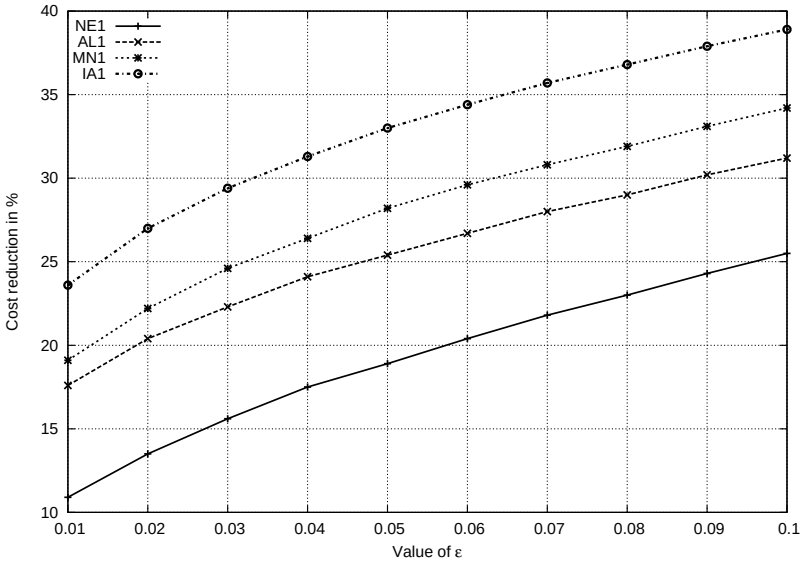
Given an undirected graph $G = (V, E)$ with nonnegative edge costs and node pairs Q, we consider the problem of selecting a minimum cost set of edges so that the induced subgraph contains at least K edge-disjoint paths containing at most L edges between each pair in Q. The problem is motivated by the need of developing survivable networks without loss of service quality, even in case of network failure (link or node failure). The problem has been a very active source of investigation in the last 10 years and it is only recently that researchers have introduced formulations valid for any positive integers K, L, see Botton et al. (2013); Mahjoub et al. (In press). These formulations use a small set of binary design variables, x, denoting which edges belong to the optimal solution, and a larger set of real flow variables defined on layered graphs. We denote by $X' \subseteq \{0,1\}^{|E|}$ the set of binary vectors x that define a subgraph feasible for the problem. Denoting by $c : E \to \mathbb{R}_+$ the cost of using the different edges of E, the deterministic version of the problem can thus be stated as

$$\min_{x \in X'} c^T x.$$

The robust versions of the problem can be further defined as $\min_{x\in X'} \max_{c\in \mathbf{U}^\Gamma} c^T x$ and $\min_{x\in X'} \max_{c\in \mathcal{U}^\beta} c^T x$, respectively.

In the following, we compare models $\mathbf{U}^\Gamma$ and $\mathcal{U}^{\partial\beta}$ on two instances taken from Botton et al. (2013); Mahjoub et al. (In press): TC-21-5 and TC-41-10. Both instances are defined by complete undirected graphs with 21 and 41 nodes, respectively, where 5 and 10 node pairs must be connected,

Figure 3: Cost reduction when using model $\mathcal{U}^{\partial\beta}$ instead of model $\mathbf{U}^{\Gamma}$.



respectively. We further suppose that $K = 2$ and $L = 3$ and consider two values of ϵ: 0.05 and 0.10. We define the extreme cost function $\hat{c} : E \to \mathbb{R}_+$ as follows. For each $\delta \in \{0.25, 0.5, 1, 2, 4, 8\}$, we create 5 instances where $\hat{c}_e$ is randomly and uniformly generated in $[0, \delta c_e]$ for each $e \in E$.

For these two instances, the comparison from Mahjoub et al. (In press) and unreported results show that the fastest solution method for the deterministic problem is the formulation from Botton et al. (2013), without using the Benders' decomposition algorithm. Based on this, we use the formulation from Botton et al. (2013) to compare the the solution times required by the dualization of Theorem 2 and those required by the method that consists in solving $n + 1$ deterministic problems, see Theorem 3. The deterministic version of the problem is solved in 0.34 seconds and 13 seconds, respectively. For each of the robust instances, we set a time limit of 5000 multiplied by the deterministic solution time. We report in Table 4 the geometric averages of these solution times expressed as multiples of the deterministic solution times. Average solution times exceeding the time limit are denoted by T. Solution times for model $\mathcal{U}^{\partial\beta}$ do not count the time needed to compute $\partial\beta$ by solving model $\mathbf{U}^{\Gamma}$ and approximating β at its optimal solution. Results

from Table 4 show that the solution times for solving problem $CO^{\partial\beta}$ by the dualization are strongly impacted by the value of δ. Still, the dualization approach is faster than the method that solves $n+1$ deterministic problems for all our instances, the speed-up factor being as large as 171 on average for instance TC-41-10 with $\delta = 0.25$ and $\epsilon = 0.05$.

We report in Figure 4 the cost reduction obtained when using model $\mathcal{U}^{\partial\beta}$ instead of model $\mathbf{U}^{\Gamma}$, expressed in percentage of the optimal solution cost of model $\mathbf{U}^{\Gamma}$. Results for instance TC-41-10 with $\delta = 8$ are not available because the instances could not be solved within the time limit.

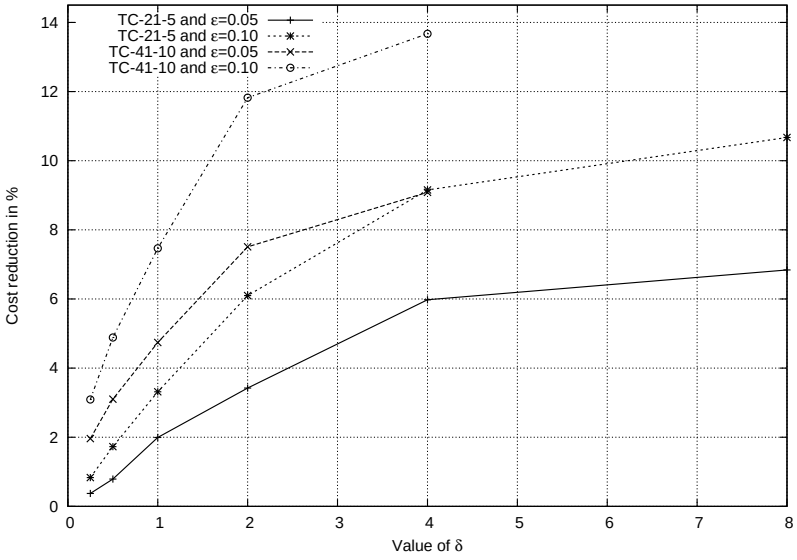
Table 4: Solution time divided by deterministic solution time.

δ	ϵ	TC-21-5				TC-41-10			
		$\mathbf{U}^{\Gamma}$		$\mathcal{U}^{\partial\beta}$		$\mathbf{U}^{\Gamma}$		$\mathcal{U}^{\partial\beta}$	
		Dual.	$n+1$ det.	Dual.	$n+1$ det.	Dual.	$n+1$ det.	Dual.	$n+1$ det.
0.25	0.05	1.3	127	2.3	121	13	654	4.4	754
	0.10	2.2	127	2.4	124	10	666	6.1	737
0.5	0.05	1.7	117	3.5	111	10	643	38	934
	0.10	2.3	118	3.2	112	14	642	26	833
1	0.05	2.7	108	6.7	105	25	536	118	1413
	0.10	4.2	109	6.8	105	32	547	81	1078
2	0.05	2.8	95	13	97	26	459	492	3460
	0.10	4.2	94	13	95	47	451	321	2066
4	0.05	2.1	90	27	95	55	389	2561	T
	0.10	3.7	90	26	93	82	392	1475	T
8	0.05	2.4	87	49	92	17	336	T	T
	0.10	2.6	87	48	90	14	336	T	T

7. Conclusion

We have presented in this paper a new approach to robust combinatorial optimization with cost uncertainty where the budget of uncertainty is given by a function γ, rather than a constant Γ as in the classical model. The use of a budget function γ allows us to better modulate the degree of conservatism required, according to cardinality of the optimal solution. While we stick to specific functions γ that are motivated by the probabilistic bounds from Bertsimas and Sim, our results can be extended to other budget functions.

Figure 4: Cost reduction when using model $\mathcal{U}^{\partial\beta}$ instead of model $\mathbf{U}^{\Gamma}$.



This would allow us to use different budget functions in practice if they capture better the uncertainty arising from a specific context.

Bang-Jensen, J., Gutin, G., 2007. Digraphs Theory, Algorithms and Applications. Springer-Verlag.

Bellman, R., 1958. On a routing problem. Quart. Appl. Math. 16, 87–90.

Ben-Tal, A., Nemirovski, A., 1999. Robust solutions of uncertain linear programs. Operations Research Letters 25, 1–13.

Bentz, C., Costa, M.-C., Derhy, N., Roupin, F., 2009. Cardinality constrained and multicriteria (multi)cut problems. Journal of Discrete Algorithms 7 (1), 102 – 111.

Bertsimas, D., Sim, M., 2003. Robust discrete optimization and network flows. Mathematical Programming 98, 49–71.

Bertsimas, D., Sim, M., 2004. The price of robustness. Operations Research 52, 35–53.

Botton, Q., Fortz, B., Gouveia, L., Poss, M., 2013. Benders decomposition for the hop-constrained survivable network design problem. INFORMS Journal on Computing 25 (1), 13–26.

Cornuejols, G., Tutuncu, R., 2006. Optimization methods in finance. Vol. 13. Cambridge University Press.

CPLEX, 2013. IBM-ILOG Cplex Optimizer.

Erdogan, E., Iyengar, G., 2006. Ambiguous chance constrained problems and robust optimization. Mathematical Programming 107, 37–61.

Held, M., Karp, R. M., 1962. A dynamic programming approach to sequencing problems. Journal of the Society for Industrial and Applied Mathematics 10 (1), 196–210.

Klopfenstein, O., Nace, D., 2008. A robust approach to the chance-constrained knapsack problem. Oper. Res. Lett. 36 (5), 628–632.

Mahjoub, A., Simonetti, L., Uchoa, E., In press. Hop-level flow formulation for the survivable network design with hop constraints problem. Networks.

Monaci, M., Pferschy, U., Serafini, P., 2013. Exact solution of the robust knapsack problem. Computers & OR 40 (11), 2625–2631.

P. Brucker, S. K., 2006. Complex Scheduling. Springer.

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

Righini, G., Salani, M., 2008. New dynamic programming algorithms for the resource constrained elementary shortest path problem. Netw. 51 (3), 155–170.

Sniedovich, M., 2011. Dynamic Programming Foundations and Principles 2nd. CRC.

Žáčková, J., 1966. On minimax solution of stochastic linear programming problems. Časopis pro Pěstování Matematiky 91, 423–430.

Zhan, F. B., Noon, C. E., 1998. Shortest path algorithms: An evaluation using real road networks. Transportation Science 32 (1), 65–73.

AppendixA. Proof of Theorem 1

We first prove that

$$\frac{z(CO^{\gamma})}{z(CO^{\Gamma})} \geq \frac{\gamma(\lceil\Gamma\rceil)-1}{\lceil\Gamma\rceil}.$$

Let x^* be the optimal solution of problem CO^{γ}. In what follows, we suppose without loss of generality that $\gamma(x^*) < \Gamma$. The following holds:

$$\frac{z(CO^{\gamma})}{z(CO^{\Gamma})} \geq \frac{\max\limits_{c\in\mathcal{U}^{\gamma}} c^T x^*}{\max\limits_{c\in\mathbf{U}^{\Gamma}} c^T x^*} = \frac{c^T x^* + \max\limits_{\delta\in[0,1]^n, \sum\delta_i\leq\gamma(x^*)} \sum\limits_i \delta_i\hat{c}_i x_i^*}{c^T x^* + \max\limits_{\delta\in[0,1]^n, \sum\delta_i\leq\Gamma} \sum\limits_i \delta_i\hat{c}_i x_i^*} \tag{A.1}$$

$$\geq \frac{\max\limits_{\delta\in[0,1]^n, \sum\delta_i\leq\gamma(x^*)} \sum\limits_i \delta_i\hat{c}_i x_i^*}{\max\limits_{\delta\in[0,1]^n, \sum\delta_i\leq\Gamma} \sum\limits_i \delta_i\hat{c}_i x_i^*}$$

$$\geq \frac{\max\limits_{\delta\in[0,1]^n, \sum\delta_i\leq\lfloor\gamma(x^*)\rfloor} \sum\limits_i \delta_i\hat{c}_i x_i^*}{\max\limits_{\delta\in[0,1]^n, \sum\delta_i\leq\lceil\Gamma\rceil} \sum\limits_i \delta_i\hat{c}_i x_i^*} \tag{A.2}$$

$$= \frac{\lfloor\gamma(x^*)\rfloor}{\min(\|x^*\|, \lceil\Gamma\rceil)}. \tag{A.3}$$

Inequality (A.1) holds because x^* may not be optimal for CO^{Γ} and inequality (A.3) follows from a more technical argument detailed below.

Because $\lfloor\gamma(x^*)\rfloor \in \mathbb{Z}$ and $\lceil\Gamma\rceil \in \mathbb{Z}$, we can define $\Delta^n \subset \{1,\ldots,n\}$ and $\Delta^d \subset \{1,\ldots,n\}$ as the sets of indices where δ_i is equal to 1 in the optimal solutions of the maximization problems involved in the numerator and the denominator of (A.2), respectively. Moreover, since $\lfloor\gamma(x^*)\rfloor \leq \lceil\Gamma\rceil$ we can further suppose that $\Delta^n \subseteq \Delta^d$ so that we can define $\Delta^* = \Delta^d \backslash \Delta^n$. Thus, (A.2) can be rewritten as

$$\frac{\sum\limits_{i\in\Delta^n} \hat{c}_i}{\sum\limits_{i\in\Delta^n} \hat{c}_i + \sum\limits_{i\in\Delta^*} \hat{c}_i}. \tag{A.4}$$

Let us define $\underline{\hat{c}} = \min_{i \in \Delta^n} \hat{c}_i$. Diving both members of fraction (A.4) by $\underline{\hat{c}}$, and noticing that $\hat{c}_i/\underline{\hat{c}} \leq 1$ for each $i \in \Delta^*$ we obtain

$$\frac{\sum_{i \in \Delta^n} \hat{c}_i/\underline{\hat{c}}}{\sum_{i \in \Delta^n} \hat{c}_i/\underline{\hat{c}} + \sum_{i \in \Delta^*} \hat{c}_i/\underline{\hat{c}}} \geq \frac{\sum_{i \in \Delta^n} \hat{c}_i/\underline{\hat{c}}}{\sum_{i \in \Delta^n} \hat{c}_i/\underline{\hat{c}} + \sum_{i \in \Delta^*} 1}. \tag{A.5}$$

Inequality (A.3) finally follows from subtracting $\sum_{i \in \Delta^n}(\hat{c}_i/\underline{\hat{c}} - 1)$ from the two members of the rhs of inequality (A.5) and recalling that $|\Delta^n| = \lfloor \gamma(x^*) \rfloor$ and $|\Delta^n| + |\Delta^*| = \min(\|x^*\|, \lceil \Gamma \rceil)$.

We have proven

$$\frac{z(CO^\gamma)}{z(CO^\Gamma)} \geq \frac{\lfloor \gamma(x^*) \rfloor}{\min(\|x^*\|, \lceil \Gamma \rceil)}. \tag{A.6}$$

Two cases are left to analyze to conclude the proof of validity of bound (16):

- if $\|x^*\| \geq \lceil \Gamma \rceil$ the rhs of (A.6) becomes $\frac{\lfloor \gamma(x^*) \rfloor}{\lceil \Gamma \rceil}$, which is greater than or equal to $\frac{\lfloor \gamma(\lceil \Gamma \rceil) \rfloor}{\lceil \Gamma \rceil}$ because γ is non-decreasing.
- if $\|x^*\| < \lceil \Gamma \rceil$ we obtain

  $$\frac{\lfloor \gamma(x^*) \rfloor}{\|x^*\|} \geq \frac{\gamma(x^*) - 1}{\|x^*\|} \geq \frac{\gamma(\lceil \Gamma \rceil) - 1}{\lceil \Gamma \rceil},$$

  where the second inequality follows from (15).

We prove next that bound (16) is asymptotically tight. Consider optimization problems such that $\overline{c} = 0$, $\hat{c}_i = 1$ for each $i = 1, \ldots, n$, and $X \subset \{x, \|x\| \geq k\}$. We see immediately that

$$\frac{z(CO^\gamma)}{z(CO^\Gamma)} = \frac{\gamma(k)}{\min(k, \Gamma)}. \tag{A.7}$$

Notice that since γ satisfies (15), we obtain that $\frac{\gamma(k)}{k}$ is non-increasing for all $k \geq \underline{k}$. Hence, because γ is non-decreasing, the minimum of (A.7) is reached at $k = \Gamma$ if Γ is integer. If Γ is not integer, the minimum of (A.7) is reached either at $k = \lfloor \Gamma \rfloor$ or at $k = \lceil \Gamma \rceil$, yielding the following value for (A.7):

$$\min\left(\frac{\gamma(\lfloor \Gamma \rfloor)}{\lfloor \Gamma \rfloor}, \frac{\gamma(\lceil \Gamma \rceil)}{\Gamma}\right), \tag{A.8}$$

which is not smaller than bound (16). Consider next the particular function $\gamma(x) = \alpha(x) = (-2\ln(\epsilon)\|x\|)^{\frac{1}{2}}$ and denote $(-2\ln(\epsilon))^{\frac{1}{2}}$ by $K > 0$ and $\alpha(n) = (-2\ln(\epsilon)n)^{\frac{1}{2}} = Kn^{\frac{1}{2}}$ by $m^{\frac{1}{2}}$. We show below that the value of (A.8) converges to the rhs of (16) as $m = K^2 n$ goes to infinity:

$$\begin{aligned}
\min\left(\frac{K(\lfloor m^{\frac{1}{2}}\rfloor)^{\frac{1}{2}}}{\lfloor m^{\frac{1}{2}}\rfloor}, \frac{K(\lceil m^{\frac{1}{2}}\rceil)^{\frac{1}{2}}}{m^{\frac{1}{2}}}\right) - \frac{K(\lceil m^{\frac{1}{2}}\rceil)^{\frac{1}{2}} - 1}{\lceil m^{\frac{1}{2}}\rceil} &\leq \frac{K(\lceil m^{\frac{1}{2}}\rceil)^{\frac{1}{2}}}{m^{\frac{1}{2}}} - \frac{K(\lceil m^{\frac{1}{2}}\rceil)^{\frac{1}{2}} - 1}{\lceil m^{\frac{1}{2}}\rceil} \\
&\leq \frac{K(\lceil m^{\frac{1}{2}}\rceil)^{\frac{1}{2}} + 1}{m^{\frac{1}{2}} + 1} - \frac{K(\lceil m^{\frac{1}{2}}\rceil)^{\frac{1}{2}} - 1}{m^{\frac{1}{2}} + 1} \\
&\leq \frac{2}{m^{\frac{1}{2}} + 1}.
\end{aligned}$$