



HAL
open science

Robust combinatorial optimization with variable cost uncertainty

Michael Poss

► **To cite this version:**

Michael Poss. Robust combinatorial optimization with variable cost uncertainty. European Journal of Operational Research, 2014, 237 (3), pp.836-845. 10.1016/j.ejor.2014.02.060 . hal-01099569

HAL Id: hal-01099569

<https://inria.hal.science/hal-01099569>

Submitted on 4 Jan 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.



Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Robust combinatorial optimization with variable cost uncertainty

Michael Poss*

UMR CNRS 7253 Heudiasyc, Université de Technologie de Compiègne, Centre de Recherches de Royallieu, 60200 Compiègne, France

ARTICLE INFO

Article history:

Received 26 July 2013

Accepted 27 February 2014

Available online xxx

Keywords:

Combinatorial optimization

Robust optimization

Dynamic programming

Price of robustness

Budgeted uncertainty

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. An example of budget function is 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. 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.

© 2014 Elsevier B.V. All rights reserved.

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

$$CO \equiv \min_{x \in X} c^T x \quad (1)$$

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:

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

where $\text{VaR}_\epsilon(c^T x) = \inf\{t | P(c^T x \leq t) \geq 1 - \epsilon\}$. 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 & 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.

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 minimizes its worst-case cost:

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

This approach has several advantages, the first of which is its numerical complexity. Given a positive parameter Γ and bounds \bar{c}_i and $\bar{c}_i + \hat{c}_i$ for each $i = 1, \dots, n$, Bertsimas and Sim (2003) consider the budgeted uncertainty polytope

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

and prove that problem

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

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 it possible to solve a large variety of robust combinatorial optimization problems. Recent works (Büsing & D'Andreagiovanni, 2012; Mattia, 2012) have independently extended the properties of \mathbf{U}^Γ to richer classes of uncertainty polytopes.

Another advantage of robust optimization lies in its less specific assumptions on the uncertainty. In many applications, it is

* Tel.: +33 3226505880.

E-mail address: mjposs@gmail.com

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 exist also intermediary models that assume partial knowledge on the distribution of the uncertain parameters. These models are usually cited as ambiguous chance constraints (Erdogan & 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}^f 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 amount of uncertainty is not bound by a static number but by a non-negative function $\gamma(x)$ defined on the feasibility region X . Namely, the author defines variable budgeted uncertainty as multifunction $\mathcal{U}^\gamma : \Rightarrow \mathbb{R}^n$, defined as follows:

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

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}^γ to optimization problems with uncertain costs, yielding

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

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^f .

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 $z(CO^f)$ the optimal solutions costs of problems CO^γ and CO^f , respectively. Finally, we assume throughout that function γ satisfies the following:

1. The function γ is non-decreasing.
2. $0 \leq \gamma(\|x\|) \leq \|x\|$ for each $x \in X$.
3. For any $x^1, x^2 \in \{0, 1\}^n$ such that $\|x^1\| = \|x^2\|$, $\gamma(x^1) = \gamma(x^2)$.

In view of the third property above, we sometimes commit an abuse of notation and denote $\gamma(k)$ with $k \in \mathbb{Z}$. 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.

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 two important properties of model CO^γ . Section 2.1 applies the techniques from Poss (2013) to explain how we can define γ to obtain a better bound for $\min_{x \in X} \text{VaR}_\epsilon(c^T x)$ than the classical budgeted uncertainty model. We provide then 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^f . Up to our knowledge, Monaci and Pferschy (2013) is the only prior work where a theoretical study of price of robustness is proposed.

In Sections 3 and 4, we turn to the study of the computational complexity of CO^γ . In Section 3, we reduce problem CO^γ to solving a sequence of problems CO or its cardinality-constrained version, which depends on whether γ is affine or not. In Section 4, we focus on problems CO for which efficient dynamic programming algorithms are available. We adapt these algorithms to solve the robust versions CO^f and CO^γ . The contributions of Sections 3 and 4 are summarized in Table 1 where we provide the complexity of CO^γ depending on the existence of a dynamic programming algorithm for CO and on the type of function γ (affine or integer-valued). The values reported in the column “Complexity” assume that γ has been chosen in accordance with the simplest probabilistic bound from Bertsimas and Sim (2004); τ and τ^\leq denote the times for solving CO and its cardinality-constrained version, respectively. Recalling from Bertsimas and Sim (2003) that CO^f can be solved in $O(n\tau)$, we see from Table 1 that when γ is affine the solution times of CO^f and CO^γ are of the same order of magnitude.

In addition to these results, our dynamic programming approach from Section 4 also applies to the classical robust model CO^f 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). Hence, our result extends to a large class of dynamic programming algorithms the ideas proposed by Klopfenstein and Nace (2008) and Monaci, Pferschy, and Serafini (2013) for the robust knapsack problem, by using a more general description of dynamic programming.

In Section 5, we present a numerical comparison of models CO^f 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. Finally, we conclude the paper in Section 6, and present a MILP formulation CO^γ and technical proofs in Appendices A–D.

2. Main properties

2.1. Probabilistic motivation

The main motivation of model \mathcal{U}^γ comes from the probabilistic bounds computed by Bertsimas and Sim (2004) and extended to variable uncertainty by Poss (2013). We recall below how the strongest of these bounds applies to the problem of minimizing the value-at-risk. Namely, we show that for a properly chosen

Table 1
Summary of our results.

Dynamic programming algorithm available?	Type of function γ	Complexity
No	Affine	$O(n\tau)$
No	–	$O(n^2\tau^\leq)$
Yes	Integer-valued	$O(n^{3/2}\tau)$
Yes	–	$O(n^{5/2}\tau)$

function, denoted β in what follows, the optimal solution of robust model CO^β provides a less conservative solution for the stochastic problem (2) than model CO^Γ . We sketch next how to construct function β , which does not have an analytical expression. The full details of the construction can be found in Poss (2013, Section 3), where the author also introduces another function, α , which has an analytical expression but leads to a more conservative model than β .

Let $\tilde{c}_i = \bar{c}_i + \eta_i \hat{c}_i$ be a random variable associated with parameter c_i and suppose that $\eta_i, i = 1, \dots, n$, are arbitrary random variables independently and symmetrically distributed in $[-1, 1]$. Bertsimas and Sim (2004) and Poss (2013) show that any vector x that satisfies the robust constraint $c^T x \leq t$ for all $c \in \mathbf{U}^T$, satisfies

$$P(\tilde{c}^T x > t) \leq B(\|x\|, \Gamma) = \frac{1}{2^{\|x\|}} \left((1 - \mu(x)) \binom{\|x\|}{\lfloor v(x) \rfloor} + \sum_{l=\lfloor v(x) \rfloor+1}^{\|x\|} \binom{\|x\|}{l} \right), \tag{8}$$

where $v(x) = (\Gamma + \|x\|)/2$ and $\mu(x) = v(x) - \lfloor v(x) \rfloor$. Inequality (8) implies that satisfying probabilistic constraint

$$P(\tilde{c}^T x > t) \leq \epsilon \tag{9}$$

with $\epsilon \in (0, 1)$ can be done by choosing a proper value of Γ , which depends on $\|x\|$. Namely, when $\|x\|$ are ϵ are given, the smallest Γ that ensures the satisfaction of inequality (9) is given by

$$\min \Gamma \text{ s.t. } B(\|x\|, \Gamma) \leq \epsilon. \tag{10}$$

Problem (10) 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 define function β as follows

$$\beta(x) = \begin{cases} \text{solution to (10) if the problem is feasible} \\ \|x\| \text{ otherwise.} \end{cases} \tag{11}$$

The key property of CO^β is presented next.

Theorem 1. Let $\beta(x)$ be as in (11) and $\epsilon \in (0, 1)$ be a probability level. It holds that

$$\min_{x \in X} \text{Var}_\epsilon(\tilde{c}^T x) \leq \min_{x \in X} \max_{c \in \mathbf{U}^\beta(x)} c^T x. \tag{12}$$

Inequality (12) also holds for the classical model of budgeted uncertainty where Γ is fixed and equal to $\beta(n)$. Since $\beta(x) \leq \Gamma$ for all $x \in X$, then

$$\min_{x \in X} \max_{c \in \mathbf{U}^\beta(x)} c^T x \leq \min_{x \in X} \max_{c \in \mathbf{U}^\Gamma(x)} c^T x \tag{13}$$

holds and the approximation of problem (2) provided by CO^β 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 $\beta(n)$ holds independently of the cardinality of any solution. In the extreme situation, Γ may even lie above the cardinality of the optimal 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 $O(|E|) = O(|V|^2)$. This situation is illustrated in our numerical experiments. In contrast to this, model CO^β provides a tighter bound by reducing the value of Γ as the cardinality of x decreases.

Choosing function γ according to the probabilistic guarantees computed by Bertsimas and Sim (2004) leads to using non-linear functions such as β and α described in Poss (2013). There exist different ways to avoid this non-linearity. Poss (2013) suggests to replace a non-linear function γ by a piece-wise upper 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 over-approximates γ as a function of k at the optimal solution of CO^Γ .

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

2.2. Cost reduction

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

there exists a positive integer \underline{k} such that $\frac{\gamma(k) - 1}{k}$ is non-increasing for all $k \geq \underline{k}$. (14)

One can check numerically that function β introduced in Theorem 1 satisfies condition (14). The proof of the next theorem is provided in Appendix B.

Theorem 2. Let CO be any combinatorial optimization problem and $\Gamma = \gamma(n)$. Suppose that γ satisfies property (14) and that the optimal solution of CO^γ has a cardinality greater than or equal to \underline{k} . It holds that:

$$\frac{z(CO^\gamma)}{z(CO^\Gamma)} \geq \frac{\gamma(\lceil \Gamma \rceil) - 1}{\lceil \Gamma \rceil}. \tag{15}$$

Bound (15) is asymptotically tight with respect to n for $\gamma = (-2 \ln(\epsilon) \|x\|)^{\frac{1}{2}}$, $\bar{c} = 0$, $\hat{c}_i = 1$ for each $i = 1, \dots, n$, and $X \subset \{x, \|x\| \geq \underline{k}\}$.

Function $(-2 \ln(\epsilon) \|x\|)^{\frac{1}{2}}$ mentioned in Theorem 2 has been introduced in Poss (2013) where the author shows how the function can also be used to decrease the conservatism of \mathbf{U}^T . However, it is not an easy task to verify analytically the tightness of (15) for function β given that the function does not have an analytical expression. Hence, we illustrate numerically in Fig. 1 the bounds provided by (15) for β . Fig. 1 suggests that bound (15) 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 $\bar{c} = 0$ and $\hat{c}_i = 1$ for each $i = 1, \dots, n$. Our numerical experiments from Section 5 realized on two combinatorial optimization problems with hundreds of variables exhibit cost reduction ranging from less than 1% up to 40%.

In the following two sections, we study the computational complexity of problem CO^γ .

3. 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 :

$$\gamma(x) = \gamma_0 + \sum_{i=1}^n \gamma_i x_i. \tag{16}$$

We see next that surprisingly, the computational complexity of CO^γ for an affine function γ is essentially the same as the complexity of

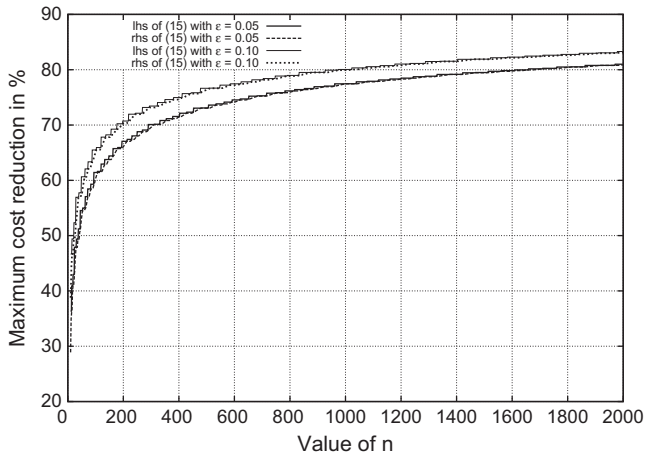


Fig. 1. Maximal theoretical values for $\frac{z(CO^T) - z(CO^\gamma)}{z(CO^T)}$ expressed in percentage.

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 of 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:

$$\gamma_0 \hat{c}_l + \min_{x \in X} \sum_{i=1}^n (\hat{c}_i + \gamma_i \hat{c}_i) x_i + \sum_{i=1}^l (\hat{c}_i - \hat{c}_i) x_i, \quad (17)$$

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

The only difference between problem (17) and the problems solved in Bertsimas and Sim (2003, Theorem 3) lies in the costs coefficients of x . In problem (17), 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 mention that when Γ is integer, Álvarez-Miranda, Ljubic, and Toth (2013) and Lee, Lee, Park, and Park (2012) show how the number of problems solved in Bertsimas and Sim (2003, Theorem 3) can be reduced to $n - \Gamma + 2$. The reduction is not useful in our context because, even if we assume that γ is integer-valued, the reduction would lead to solving $n - \gamma(k) + 2$ problems where k is the cardinality of the optimal solution. Since we do not know that cardinality, we should use a conservative approach, leading to $n - \gamma(0) + 2$ problems to be solved where $\gamma(0) = 0$.

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

$$CO_k^\leq \equiv \min_{x \in X, \|x\| \leq k} c^T x. \quad (18)$$

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:

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

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

Proof. We must prove that following equality holds

$$\min_{x \in X} \max_{c \in U^{\gamma(x)}} c^T x = \min_{k=1, \dots, n} \min_{x \in X, \|x\| \leq k} \max_{c \in U^{\gamma(k)}} c^T x. \quad (19)$$

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

$$\min_{x \in X, \|x\| \leq k} \max_{c \in U^{\gamma(k)}} c^T x \geq \min_{x \in X, \|x\| \leq k} \max_{c \in U^{\gamma(x)}} c^T x \geq \min_{x \in X} \max_{c \in U^{\gamma(x)}} c^T x, \quad (20)$$

where inequality (20) follows from the fact that γ is non-decreasing.

≥: Let x^* be the optimal solution of the left-hand-side of (19) and denote $k^* = \|x^*\|$. The following holds

$$\begin{aligned} \min_{x \in X} \max_{c \in U^{\gamma(x)}} c^T x &= \min_{x \in X} \max_{c \in U^{\gamma(x^*)}} c^T x \geq \min_{x \in X, \|x\| = k^*} \max_{c \in U^{\gamma(k^*)}} c^T x \\ &\geq \min_{x \in X, \|x\| \leq k^*} \max_{c \in U^{\gamma(k^*)}} c^T x, \end{aligned} \quad (21)$$

where $U^{\gamma(k^*)} = U^{\gamma(x^*)}$ was used to obtain inequality (21). □

Applying sequentially Theorems 4 and 3, we obtain that CO^Γ is polynomially solvable when CO_k^\leq is polynomially solvable.

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

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

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, Costa, Derhy, & Roupin, 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

$$\gamma(x) = \begin{cases} 0 & \text{if } \|x\| \leq k \\ 1 & \text{if } \|x\| > k \end{cases}, \quad (22)$$

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

Proof. Consider an instance of CO_k^\leq and create an associated instance of CO^γ by setting $\hat{c}_i = \sum_i c_i + 1$ for each $i = 1, \dots, n$, and defining γ as in (22). 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 (18) 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.

4. 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 3, the results of this section also provide faster algorithms for solving the

classical robust optimization problem CO^F . 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 (Snedovich, 2011). We define precisely in Section 4.1 the type of functional equations we consider in this paper, which include many well-known DP algorithms such as the Bellman–Ford algorithm and the DP approach to the knapsack problem. We show then in Sections 4.2 and 4.3 how the aforementioned DP algorithms extend naturally to robust problems CO^F and CO^γ .

4.1. 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 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 $\mathcal{O} \subset S$ 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 $p(s, i) \in S$ 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:

$$\begin{cases} F(s) = \min_{i \in q(s)} \{F(p(s, i)) + c_i\}, & s \in S \setminus \mathcal{O} \\ F(s) = 0, & s \in \mathcal{O}, \end{cases} \quad (23)$$

and all states that are not in \mathcal{O} are initialized with the value $+\infty$. Let $Q = \max_{s \in S} |q(s)|$. In the worst-case, functional Eq. (23) requires to visit all spaces of S , yielding a total time of $O(Q|S|)$, thus yielding a non-polynomial running-time in general. 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 $V = S$ and $A = \{(p(s, i), s) : s \in S, i \in q(s)\}$ 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 Eq. (23) to the robust cases.

We provide below a classical example of DP algorithm that follows Eq. (23): the Bellman–Ford algorithm. Other applications can be found in the literature, such as the constrained shortest path problem (Righini & Salani, 2008), the traveling salesman problem (Held & Karp, 1962), the capital budgeting problem (Cornuejols & Tutuncu, 2006), and scheduling problems (Brucker & Knust, 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 $S = \{(i, k) : i \in V, k = 0, \dots, |V| - 1\}$, with initial state space $\mathcal{O} = \{(o, k) : k = 0, \dots, |V| - 1\}$ 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 $q(i, 0) = \emptyset$.

4.2. Problem CO^F

We show below that the solution cost of CO^F can be computed using a functional equation that is similar to Eq. (23). The proof is omitted and can be found in Appendix C.

Theorem 6. Consider an instance of problem CO that can be solved in $O(\tau)$ by using functional Eq. (23). Then, its robust version CO^F can be solved in $O(\Gamma\tau)$.

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

4.3. 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^F . In the next result, we use the notation $\Gamma^\gamma = \min(n, \max_{k=0, \dots, n} \lfloor \gamma(k) \rfloor)$. The proof is omitted and can be found in Appendix D.

Theorem 7. Consider an instance of problem CO that can be solved in $O(\tau)$ by using functional Eq. (23). If $\gamma(k) \in \mathbb{Z}$ for each $k = 0, \dots, n$, then CO^γ can be solved in $O(n\Gamma^\gamma\tau)$. Otherwise, CO^γ can be solved in $O(n^2\Gamma^\gamma\tau)$.

5. 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^F where β is given by Theorem 1 and $\Gamma = \beta(n)$ is chosen accordingly. This cost reduction is formally defined as $\frac{CO^F - CO^\beta}{CO^F}$. As a byproduct, we compare numerically the efficiency of some of the solution methods proposed in Section 3 and the dualization proposed by Poss (2013) and recalled in Appendix A.

We have seen in Section 3 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^F 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 $UB(CO^\beta)$ equal to $\max_{x \in \mathcal{X}} c^T x^*$.

The value of β was computed numerically with a precision of three decimals by solving problem (10) by enumeration. We could easily compute β for k going up to 3000, which was enough for our computational experiments. These values are illustrated in Fig. 2 for three choices of ϵ . The experiments have been run on a computer equipped with a processor Intel Core i7-3520 M at 2.90 GHz and 8 GB of RAM. All algorithms were coded in JAVA and CPLEX (2013) 12.4 was used as the MIP solver with standard parameter settings.

5.1. The shortest path problem

We illustrate in this section the gain of the new model CO^β compared to the classical model CO^F on the $o - t$ shortest path problem defined in graph $G = (V \cup \{o, t\}, A)$ that may contain cycles and

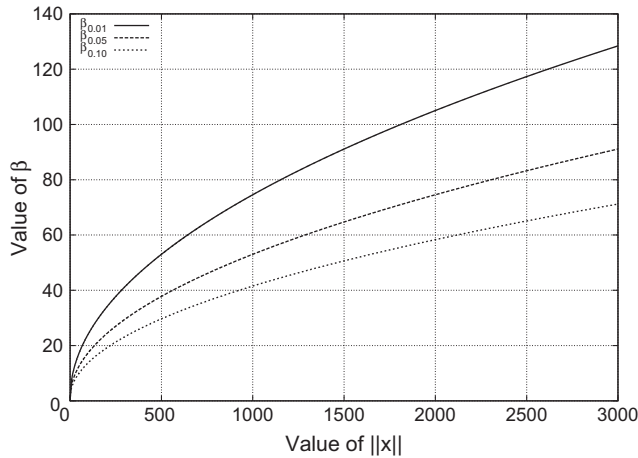


Fig. 2. The value of β for $\epsilon = 0.01, 0.05,$ and 0.10 .

where all costs are positive. The problem can be modeled as follows. Let $|V| = n$ and $|A| = m$, and $c : A \rightarrow \mathbb{R}_+$ and $\hat{c} : A \rightarrow \mathbb{R}_+$ 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 $\delta^-(i) = \{j \in V : \exists(j, i) = a \in A\}$. The formulation follows:

$$\begin{aligned} \min \quad & c^T x & (24) \\ \text{(SP)} \quad & \text{s.t.} \quad \sum_{a \in \delta^-(i)} x_a - \sum_{a \in \delta^+(i)} x_a = b_i \quad i \in V \cup \{o, t\} \\ & x_a \in \{0, 1\} \quad a \in A. \end{aligned}$$

The formulation above can be adapted to cost varying in \mathbf{U}^r and \mathbf{U}^β by replacing objective function (24) with $\min_{c \in \mathbf{U}^r} c^T x$ and $\min_{c \in \mathbf{U}^\beta} c^T x$, respectively. Our experiments are based on four real road networks used in Zhan and Noon (1998), NE1, AL1, MN1, and IA1, and the main characteristics of the networks are presented 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, problems CO^r and CO^{β} can be solved by solving $n + 1$ shortest path problems, respectively, 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 a sequence of Dijkstra’s algorithm. Using Theorem 4, problem CO^β 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 4.2.

We provide in Table 3 the solution costs for each model, expressed as percentages of the deterministic solution cost. For each value of ϵ in $\{0.01, 0.02, \dots, 0.1\}$, we generate 5 instances and report the geometric averages. One can see from Table 3 that the cost

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

ϵ	CO^r	$UB(CO^\beta)$	CO^{β}	CO^β
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

of problem CO^r 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 problem CO^r 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 problems CO^r and CO^{β} and each instance could be solved in around 2.5 seconds. Recall, however, that problem CO^{β} requires to solve first problem CO^r to obtain the vector x^* at which β is linearized. Hence, the total solution time for problem CO^{β} around 5 seconds. Problem CO^β 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 problem CO^β over problem CO^{β} is very little in comparison to the gain of problem CO^{β} over problem CO^r . For this reason, we restrict our attention to the comparison of problems CO^r and CO^{β} in what follows. We report in Fig. 3 the cost reduction when using problem CO^{β} instead of problem CO^r , expressed in percentage of the optimal solution cost of problem CO^r .

5.2. 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

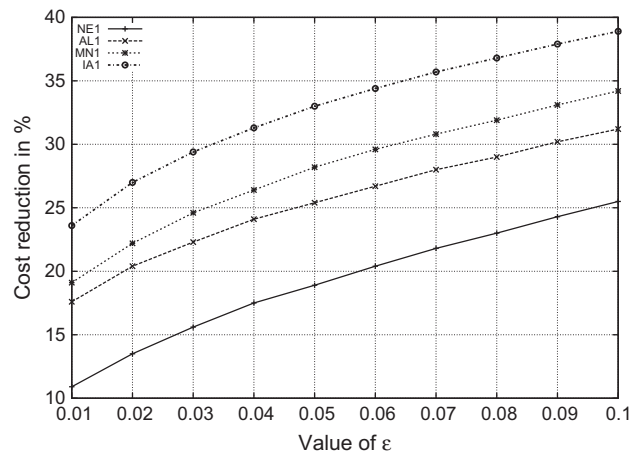


Fig. 3. Cost reduction when using problem CO^{β} instead of problem CO^r .

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

Network name (abbreviation)	V	A	Arc/node ratio	Arc Length		
				Maximum	Mean	Std. 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

Table 4
Solution time divided by deterministic solution time.

δ	ϵ	TC-21-5				TC-41-10			
		CO^T		$CO^{\partial\beta}$		CO^T		$CO^{\partial\beta}$	
		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

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, Fortz, Gouveia, and Poss \(2013\)](#), [Mahjoub, Simonetti, and Uchoa \(2013\)](#). 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 $c : E \rightarrow \mathbb{R}_+$ 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 $\min_{x \in X'} \max_{c \in U'} c^T x$ and $\min_{x \in X'} \max_{c \in U^{\partial\beta}} c^T x$, respectively. Notice that other versions of robust network design problems have been studied in the literature, see [Koster, Kutschka, and Raack \(2013\)](#) and the references therein.

In the following, we compare problems CO^T and $CO^{\partial\beta}$ on two instances taken from [Botton et al. \(2013\)](#), [Mahjoub et al. \(2013\)](#): 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, respectively. We further suppose that $K = 2$ and $L = 3$ and consider two values of $\epsilon : 0.05$ and 0.10 . We define the extreme cost function $\hat{c} : E \rightarrow \mathbb{R}_+$ 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. \(2013\)](#) 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 solution times required by the dualization of [Theorem 8](#) and those required by the method that consists in solving several 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 problem $CO^{\partial\beta}$ do not include the time needed to compute $\partial\beta$ by solving problem CO^T 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 [Fig. 4](#) the cost reduction obtained when using problem $CO^{\partial\beta}$ instead of problem CO^T , expressed in percentage of the optimal solution cost of problem CO^T . Results for instance TC-41-10 with $\delta = 8$ are not available because the instances could not be solved within the time limit.

6. 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. Our theoretical and computational results suggest that in many situations the practical applicability of our new approach is comparable to the one of the classical model. 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. This would allow us to use different budget functions in practice if they capture better the uncertainty arising from a specific context.

Appendix A. MILP reformulation

In this appendix, we recall from [Poss \(2013\)](#) how to solve CO^γ as a mixed-integer linear program whenever γ is an affine function.

Theorem 8. Let $\gamma : X \rightarrow \mathbb{R}^+$ be an affine mapping of x . Then, $\min_{x \in X} \max_{c \in U^\gamma(x)} c^T x$ is equivalent to

$$\begin{aligned} \min \quad & \sum_{i=1}^n c_i x_i + \gamma_0 z + \sum_{i=1}^n \gamma_i w_i + \sum_{i=1}^n p_i \\ \text{s.t.} \quad & z + p_i \geq \hat{c}_i x_i, \quad i = 1, \dots, n, \\ & w_i - z \geq -\hat{c}_i (1 - x_i), \quad i = 1, \dots, n, \\ & p, w, z \geq 0, \\ & x \in \{0, 1\}^n. \end{aligned}$$

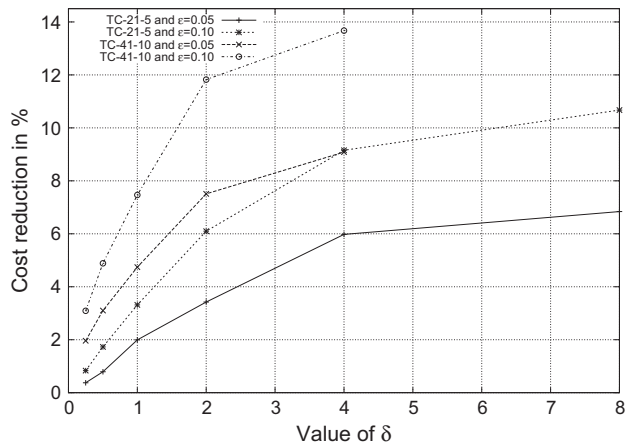


Fig. 4. Cost reduction when using problem $CO^{\theta\beta}$ instead of problem CO^F .

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

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

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

Appendix B. Proof of Theorem 2

We first prove that

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

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:

$$\frac{z(CO^\gamma)}{z(CO^F)} \geq \frac{\max_{c \in \mathcal{U}^l} c^T x^*}{\max_{c \in \mathcal{U}^f} c^T x^*} = \frac{c^T x^* + \max_{\delta \in [0,1]^n, \sum \delta_i \leq \gamma(x^*)} \sum \delta_i \hat{c}_i x_i^*}{c^T x^* + \max_{\delta \in [0,1]^n, \sum \delta_i \leq \Gamma} \sum \delta_i \hat{c}_i x_i^*} \tag{B.1}$$

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

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

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

Inequality (B.1) holds because x^* may not be optimal for CO^F and inequality (B.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 $\Delta^n \subset \{1, \dots, n\}$ and $\Delta^d \subset \{1, \dots, n\}$ 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 (B.2), respectively. Notice that these sets can easily be computed by reordering the items according to the decreasingly value of $\hat{c}_i x_i^*$, and taking in Δ^n (resp. Δ^d) the first $\lfloor \gamma(x^*) \rfloor$ (resp. $\lceil \Gamma \rceil$) elements. Hence, $\lfloor \gamma(x^*) \rfloor \leq \lceil \Gamma \rceil$ implies that $\Delta^n \subseteq \Delta^d$ so that we can define $\Delta^* = \Delta^d \setminus \Delta^n$. Thus, (B.2) can be rewritten as

$$\frac{\sum_{i \in \Delta^n} \hat{c}_i}{\sum_{i \in \Delta^n} \hat{c}_i + \sum_{i \in \Delta^*} \hat{c}_i}. \tag{B.4}$$

Let us define $\hat{c} = \min_{i \in \Delta^n} \hat{c}_i$, which is equal to $\hat{c}_{\lceil \Gamma \rceil}$ according to the reordering just mentioned. By definition, $\hat{c}_i / \hat{c} \leq 1$ for each $i \in \Delta^*$. Hence, dividing both members of fraction (B.4) we obtain

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

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

We have provens

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

Two cases are left to analyze to conclude the proof of validity of bound (15):

- if $\|x^*\| \geq \lceil \Gamma \rceil$ the rhs of (B.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 (14).

We prove next that bound (15) is asymptotically tight. Consider optimization problems such that $\bar{c} = 0$, $\hat{c}_i = 1$ for each $i = 1, \dots, n$, and $X \subset \{x, \|x\| \geq k\}$. We see immediately that

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

Notice that since γ satisfies (14), we obtain that $\frac{\gamma(k)}{k}$ is non-increasing for all $k \geq k$. Hence, because γ is non-decreasing, the minimum of (B.7) is reached at $k = \Gamma$ if Γ is integer. If Γ is not integer, the minimum of (B.7) is reached either at $k = \lfloor \Gamma \rfloor$ or at $k = \lceil \Gamma \rceil$, yielding the following value for (B.7):

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

which is not smaller than bound (15). Consider next the particular function $\gamma(x) = (-2 \ln(\epsilon) \|x\|)^{\frac{1}{2}}$ and denote $(-2 \ln(\epsilon))^{\frac{1}{2}}$ by $K > 0$ and $\gamma(n) = (-2 \ln(\epsilon) n)^{\frac{1}{2}} = Kn^{\frac{1}{2}}$ by $m^{\frac{1}{2}}$. We show below that the value of (B.8) converges to the rhs of (15) as $m = K^2 n$ goes to infinity:

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

Appendix C. Proof of Theorem 6

We consider first the case where $\Gamma \in \mathbb{Z}$. Recalling the graph analogy, the optimal solution cost of CO^F 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

$$\left\{ \begin{array}{l} F(s, \alpha) = \min_{i \in q(s)} \{ \max(F(p(s, i), \alpha) + \bar{c}_i, F(p(s, i), \alpha - 1) + \bar{c}_i + \hat{c}_i) \}, \\ \quad s \in \mathcal{S} \setminus \mathcal{O}, \quad 1 \leq \alpha \leq \Gamma, \\ F(s, 0) = \min_{i \in q(s)} \{ F(p(s, i), 0) + \bar{c}_i \}, \\ \quad s \in \mathcal{S} \setminus \mathcal{O}, \\ F(s, \alpha) = 0, \\ \quad 0 \leq \alpha \leq \Gamma, \quad s \in \mathcal{O}. \end{array} \right. \quad (C.1)$$

First, $SPP(s, \alpha) = 0 = F(s, \alpha)$ for $0 \leq \alpha \leq \Gamma$ 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 (C.1). The second equation in (C.1) 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 , \bar{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

$$\left\{ \begin{array}{l} F(s, \alpha, 1) = \min_{i \in q(s)} \{ \max(F(p(s, i), \alpha, 1) + \bar{c}_i, F(p(s, i), \alpha - 1, 1) \\ \quad + \bar{c}_i + \hat{c}_i, F(p(s, i), \alpha, 0) + \bar{c}_i + (\Gamma - \lfloor \Gamma \rfloor)\hat{c}_i) \}, \\ \quad s \in \mathcal{S} \setminus \mathcal{O}, \quad 1 \leq \alpha \leq \lfloor \Gamma \rfloor \\ F(s, \alpha, 0) = \min_{i \in q(s)} \{ \max(F(p(s, i), \alpha, 0) + \bar{c}_i, F(p(s, i), \alpha - 1, 0) + \bar{c}_i + \hat{c}_i) \}, \\ \quad s \in \mathcal{S} \setminus \mathcal{O}, \quad 1 \leq \alpha \leq \lfloor \Gamma \rfloor \\ F(s, 0, 1) = \min_{i \in q(s)} \{ \max(F(p(s, i), 0, 1) + \bar{c}_i, F(p(s, i), 0, 0) + \bar{c}_i \\ \quad + (\Gamma - \lfloor \Gamma \rfloor)\hat{c}_i) \}, \quad s \in \mathcal{S} \setminus \mathcal{O} \\ F(s, 0, 0) = \min_{i \in q(s)} \{ F(p(s, i), 0, 0) + \bar{c}_i \}, \quad s \in \mathcal{S} \setminus \mathcal{O} \\ F(s, \alpha, f) = 0, \\ \quad 0 \leq \alpha \leq \lfloor \Gamma \rfloor, \quad f \in \{0, 1\}, \quad s \in \mathcal{O} \end{array} \right. \quad (C.2)$$

The number of states in Eqs. (C.1) and (C.2) are equal to $\lfloor \Gamma \rfloor |\mathcal{S}|$ and $2\lfloor \Gamma \rfloor |\mathcal{S}|$, 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 $O(\Gamma Q |\mathcal{S}|) = O(\Gamma \tau)$.

Appendix D. Proof of Theorem 7

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

$$\left\{ \begin{array}{l} F(s, \alpha, k) = \min_{i \in q(s)} \{ \max(F(p(s, i), \alpha, k - 1) + \bar{c}_i, F(p(s, i), \alpha - 1, k - 1) + \bar{c}_i + \hat{c}_i) \}, \\ \quad s \in \mathcal{S} \setminus \mathcal{O}, \quad 1 \leq \alpha \leq \Gamma', \quad 1 \leq k \leq n, \\ F(s, 0, k) = \min_{i \in q(s)} \{ F(p(s, i), 0, k - 1) + \bar{c}_i \}, \\ \quad s \in \mathcal{S} \setminus \mathcal{O}, \quad 1 \leq k \leq n, \\ F(s, \alpha, 0) = 0, \\ \quad s \in \mathcal{S} \setminus \mathcal{O}, \quad 0 \leq \alpha \leq \Gamma', \\ F(s, \alpha, k) = 0, \\ \quad 0 \leq \alpha \leq \Gamma', \quad 0 \leq k \leq n, \quad s \in \mathcal{O}, \end{array} \right.$$

for each $l = k, \dots, n$. 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

$$\min_{k=1, \dots, n} F(s, \gamma(k), k). \quad (D.1)$$

The difficulty when considering real-valued function γ is that the fractional part of $\gamma(k)$ may be different for each $k = 1, \dots, n$. 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 $l = 1, \dots, n$. Then, for each $l = 1, \dots, n$, we compute $F^l(s, \alpha, l, 1)$ via

$$\left\{ \begin{array}{l} F^l(s, \alpha, k, 1) = \min_{i \in q(s)} \{ \max(F^l(p(s, i), \alpha, k - 1, 1) \\ \quad + \bar{c}_i, F^l(p(s, i), \alpha - 1, k - 1, 1) + \bar{c}_i + \hat{c}_i, \\ \quad F^l(p(s, i), \alpha, k - 1, 0) + \bar{c}_i + r_l \hat{c}_i) \}, \\ \quad s \in \mathcal{S} \setminus \mathcal{O}, \quad 1 \leq \alpha \leq \Gamma', \quad 1 \leq k \leq n, \\ F^l(s, \alpha, k, 0) = \min_{i \in q(s)} \{ \max(F^l(p(s, i), \alpha, k - 1, 0) \\ \quad + \bar{c}_i, F^l(p(s, i), \alpha - 1, k - 1, 0) + \bar{c}_i + \hat{c}_i) \}, \\ \quad s \in \mathcal{S} \setminus \mathcal{O}, \quad 1 \leq \alpha \leq \Gamma', \quad 1 \leq k \leq n, \\ F^l(s, 0, k, 1) = \min_{i \in q(s)} \{ \max(F^l(p(s, i), 0, k - 1, 1) \\ \quad + \bar{c}_i, F^l(p(s, i), 0, k - 1, 0) + \bar{c}_i + r_l \hat{c}_i) \}, \\ \quad s \in \mathcal{S} \setminus \mathcal{O}, \quad 1 \leq k \leq n, \\ F^l(s, 0, k, 0) = \min_{i \in q(s)} \{ F^l(p(s, i), 0, k - 1, 0) + \bar{c}_i \}, \\ \quad s \in \mathcal{S} \setminus \mathcal{O}, \quad 1 \leq k \leq n, \\ F^l(s, \alpha, 0, f) = 0, \\ \quad s \in \mathcal{S} \setminus \mathcal{O}, \quad 0 \leq \alpha \leq \Gamma', \quad f \in \{0, 1\}, \\ F^l(s, \alpha, k, f) = 0, \\ \quad 0 \leq \alpha \leq \Gamma', \quad f \in \{0, 1\}, \quad 1 \leq k \leq n, \quad s \in \mathcal{O}. \end{array} \right.$$

For each $l = 1, \dots, n$, 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

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

References

- Álvarez-Miranda, E., Ljubic, I., & Toth, P. (2013). A note on the bertsimas & sim algorithm for robust combinatorial optimization problems. *4OR*, 11(4), 349–360.
- 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.
- Brucker, P., & Knust, S. (2006). *Complex scheduling*. Springer.
- Büsing, C., & D'Andreagiovanni, F. (2012). New results about multi-band uncertainty in robust optimization. In *SEA* (pp. 63–74).
- 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. *Operations Research Letters*, 36(5), 628–632.
- Koster, A. M. C. A., Kutschka, M., & Raack, C. (2013). Robust network design: Formulations, valid inequalities, and computations. *Networks*, 2(61), 128–149.
- Lee, C., Lee, K., Park, K., & Park, S. (2012). Technical note – Branch-and-price-and-cut approach to the robust network design problem without flow bifurcations. *Operations Research*, 60(3), 604–610.
- Mahjoub, A., Simonetti, L., & Uchoa, E. (2013). Hop-level flow formulation for the survivable network design with hop constraints problem. *Networks*, 61(2), 171–179.
- Mattia, S. (2012). *Robust optimization with multiple intervals*. Tech. Rep. 12-07, IASI-CNR.
- Monaci, M., & Pferschy, U. (2013). On the robust knapsack problem. *SIAM Journal on Optimization*, 23(4), 1956–1982.
- Monaci, M., Pferschy, U., & Serafini, P. (2013). Exact solution of the robust knapsack problem. *Computers and OR*, 40(11), 2625–2631.
- 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. *Networks*, 51(3), 155–170.
- Sniedovich, M. (2011). *Dynamic programming foundations and principles* (2nd ed.). 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.