Adaptive robust optimization with objective uncertainty

Boris Detienne, Henri Lefebvre, Enrico Malaguti, Michele Monaci

Adaptive robust optimization with objective uncertainty

Boris Detienne[1], Henri Lefebvre[2], Enrico Malaguti[2] and Michele Monaci[2*]

[1]IMB UMR CNRS 5251, Inria Bordeaux Sud-Ouest, Université de Bordeaux, 200 Avenue de la Vieille Tour, Talence, 33405, France.
[2*]Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione "Guglielmo Marconi", Università di Bologna, Viale del Risorgimento, 2, Bologna, 40136, BO, Italy.

*Corresponding author(s). E-mail(s): michele.monaci@unibo.it; Contributing authors: boris.detienne@u-bordeaux.fr; henri.lefebvre@unibo.it; enrico.malaguti@unibo.it;

Abstract

In this work, we study optimization problems where some cost parameters are not known at decision time and the decision flow is modeled as a two-stage process within a robust optimization setting. We address general problems in which all constraints (including those linking the first and the second stages) are defined by convex functions and involve mixed-integer variables, thus extending the existing literature to a much wider class of problems. We show how these problems can be reformulated using Fenchel duality, allowing to derive an enumerative exact algorithm, for which we prove ϵ-convergence in a finite number of operations. An implementation of the resulting algorithm, embedding a column generation scheme, is then computationally evaluated on two different problems, using instances that are derived starting from the existing literature. To the best of our knowledge, this is the first approach providing results on the practical solution of this class of problems.

Keywords: two-stage robust optimization, reformulation, Fenchel duality, column generation, branch-and-bound, computational experiments

1 Introduction

Robust Optimization (RO) has emerged as a solution approach to deal with uncertainty in optimization problems. Contrary to stochastic optimization, another popular approach that relies on probability distributions, robust optimization considers an uncertainty set for the unknown parameters, against which the taken decision should be immune. In that sense, constraints have to be respected in every possible realization of the parameters and the objective function evaluated in the least advantageous case. The concept was first introduced in Soyster (1973) and received considerable attention in the scientific literature. Recent advances in RO can be found in Bertsimas et al (2010), Hassene et al (2009), Ben-Tal et al (2009), Leyffer et al (2020) and Yanıkoğlu et al (2019), among others.

More formally, a basic robust optimization problem can be cast as follows:

$$\begin{aligned} \inf_{\boldsymbol{z}} \ & \sup_{\boldsymbol{\xi}\in\Xi} f(\boldsymbol{\xi},\boldsymbol{z}) \\ \text{subject to } \ & \boldsymbol{g}(\boldsymbol{\xi},\boldsymbol{z}) \le \boldsymbol{0} \qquad \forall \boldsymbol{\xi}\in\Xi \\ & \boldsymbol{z}\in Z \end{aligned} \tag{1SR-P}$$

Here, the unknown data is represented by variables $\boldsymbol{\xi}$ that belong to the so-called *uncertainty set* Ξ. As mentioned above, decision $\boldsymbol{z}$ has to be feasible in every possible occurrence of the uncertainty, hence robust solutions tend to be overly conservative. To tackle this drawback, so-called *adjustable robust optimization* Ben-Tal et al (2004), also known as *two-stage robust optimization*, was introduced. As its name suggests, in a two-stage context, part of the decisions are made in a *here-and-now* phase (i.e., before uncertainty reveals), while recourse decisions can be taken in a *wait-and-see* phase (i.e., once the actual values of the uncertain data are known) as an attempt to react to the outcome of the uncertain process. Typically, the feasible space of (1SR-P) can, indeed, be recast to embed a two-stage decision process by splitting variables $\boldsymbol{z}$ in $(\boldsymbol{x},\boldsymbol{y})$ and defining set Z as $\mathcal{X}\times\mathcal{Y}$ accordingly. With the convention that the minimum objective function value for an infeasible problem is $+\infty$, a two-stage robust problem can be formulated as follows:

$$\inf_{\boldsymbol{x}\in\mathcal{X}} \sup_{\boldsymbol{\xi}\in\Xi} \inf_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x},\boldsymbol{\xi})} f(\boldsymbol{\xi},\boldsymbol{x},\boldsymbol{y}) \tag{2SR-P}$$

where $\mathcal{Y}(\boldsymbol{x},\boldsymbol{\xi}) = \{\boldsymbol{y} : \boldsymbol{y}\in\mathcal{Y},\ \boldsymbol{g}(\boldsymbol{\xi},\boldsymbol{x},\boldsymbol{y}) \le \boldsymbol{0}\}$, and $\boldsymbol{g}(\boldsymbol{\xi},\boldsymbol{x},\boldsymbol{y}) \le \boldsymbol{0}$ are the so-called *linking constraints*. Set $\mathcal{X}$ is now referred to as the first-stage feasible space. Given $\overline{\boldsymbol{x}}\in\mathcal{X}$ and $\overline{\boldsymbol{\xi}}\in\Xi$, the corresponding second-stage feasible space is $\mathcal{Y}(\overline{\boldsymbol{x}},\overline{\boldsymbol{\xi}})$, and the second-stage problem is $\inf\{f(\overline{\boldsymbol{\xi}},\overline{\boldsymbol{x}},\boldsymbol{y}) : \boldsymbol{y}\in\mathcal{Y}(\overline{\boldsymbol{x}},\overline{\boldsymbol{\xi}})\}$. It is known Ben-Tal et al (2004) that problems which can be cast as two-stage robust problems often are at least NP-hard, even in the case where first and second stage variables are continuous and all the involved functions are linear. Several approaches have been developed to tackle this class of problems. Assuming that the second stage is continuous and exhibits strong duality, it can be replaced by its dual. This way, the inner maximization problem can be reformulated using its epigraph, leading to a constraint-generation algorithm in the spirit of Benders' decomposition (see, e.g., Terry et al (2009), Bertsimas et al (2013), Jiang et al (2014) and Gabrel et al (2011)). A column-and-constraint-generation scheme has been proposed in Zeng and Zhao (2013), which consists in adding one set of recourse decision variables and the corresponding second-stage constraints associated with a realization of the uncertainty. These realizations are dynamically generated by solving a bilevel problem. Later, the same approach was used in Ayoub and Poss (2016), where the constraint-generation problem was modelled as a mixed integer program exploiting a description of the uncertainty set in terms of its extreme points. Note that this method can handle mixed-integer second-stage decisions, which is not the case for classical Benders-type approaches. Unfortunately, this method seems to be of practical relevance only when a small number of variables has to be added for reaching optimality.

The inherent difficulty of this class of problems motivated the development of approximate solution methods. In the *affine decision rule* approach proposed in Ben-Tal et al (2004), the recourse decisions are expressed as affine functions of the uncertainty. Another relevant approach, introduced in Bertsimas and Caramanis (2010), is the *finite adaptability* (also known as K-adaptability) in which the number of recourse decisions is restricted to some finite number. An MILP formulation for the case of binary second-stage decisions and objective uncertainty was proposed in Hanasusanto et al (2015) and a branch-and-bound algorithm was later proposed in Subramanyam et al (2019) to address cases with uncertain linear constraints.

An important special case of (2SR-P) arises when uncertainty affects the objective function only, i.e., $\mathcal{Y}(\boldsymbol{x},\boldsymbol{\xi}) = \mathcal{Y}(\boldsymbol{x}), \forall \boldsymbol{\xi} \in \Xi$. For this specific case, Kämmerling and Kurtz (2020) proposed an oracle-based algorithm relying on a hull relaxation combining the first- and second-stage feasible spaces embedded within a branch-and-bound framework. However, this approach applies to purely binary variables and linear constraints only. On the other hand, Arslan and Detienne (2021) proposed an exact MILP reformulation of the problem in case of linear linking constraints that involve binary variables only. Besides solving the problem by means of a branch-and-price algorithm, a further contribution of Arslan and Detienne (2021) is proving the NP-completeness of the problem in this setting.

In the setting where uncertainty affects the objective function only, our analysis shows that further effort is needed to tackle with more general cases, in particular when linking constraints are defined by nonlinear functions or involve both integer and continuous variables. Similarly, to the best of our knowledge, the case in which the objective function is nonlinear has not been considered yet. This paper contributes in filling this gap, as we consider two-stage robust problems with objective uncertainty, convex constraints and mixed-integer first and second stage. By extending in a non-trivial way some recent results from the two-stage stochastic optimization literature (see

Sherali and Fraticelli (2002), Sherali and Zhu (2006) and Li and Grossmann (2019)), we obtain a relaxation of the problem, and analyze its tightness for different special cases. This relaxation can be embedded within a branch-and-bound scheme thus producing an exact solution approach, for which we prove finite ε-convergence. Besides the theoretical analysis, we also show that, from a computational viewpoint, the proposed algorithm is able to solve instances of practical relevance arising from two different applications. We also point out that the class of problems which can be addressed by our solution approach is quite large since we only require mild assumptions on the nature of the involved optimization problem.

The article is organized as follows. In Section 2 we formally introduce the class of problems we are considering throughout this work, whereas Section 2.2 describes in greater details the algorithmic solution approach proposed in Arslan and Detienne (2021) for a special case of our problem. In Section 3 we present a relaxation of the problem, and an effective algorithm for its solution. We then derive sufficient conditions for the relaxation to coincide with the original problem in a mixed-integer context. In the purely binary case, the equivalence between problem (2SRO-P) and our lower-bounding problem is established without any condition. We then present a branch-and-bound algorithm able to close the optimality gap with finite ε-convergence assuming that the lower-bounding problem can be finitely solved with ε tolerance. In section 3.4, we propose a column-generation algorithm to solve the lower-bounding problem with such property. Finally, section 4 applies the proposed algorithm to two problems: a capital budgeting problem and a capacitated facility location problem.

Notations Throughout this paper, matrices and vectors are written in bold case, e.g., $\boldsymbol{x} \in \mathbb{R}^n$ or $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, while components are written in normal font, e.g., x_i or a_{ij}. Columns of $\boldsymbol{A}$ are written in bold case with exponent indexing, e.g., $\boldsymbol{a}^i$. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a given function with $\text{dom}(f) = \{\boldsymbol{x} \in \mathbb{R}^n : f(\boldsymbol{x}) < +\infty\}$; its convex conjugate is denoted by $f^* : \mathbb{R}^n \to \mathbb{R}$ and is given by

$$f^*(\boldsymbol{\pi}) = \sup_{\boldsymbol{x}\in\text{dom}(f)} \{\boldsymbol{\pi}^T\boldsymbol{x} - f(\boldsymbol{x})\}$$

Similarly, we denote by f_* the concave conjugate of f. Let $X \subseteq \mathbb{R}^n \times \{0,1\}^{n-p}$ be a given set, we denote $\overline{X}$ its continuous relaxation, i.e., $X = \overline{X} \cap \mathbb{R}^n \times \{0,1\}^{n-p}$, $\text{conv}\,(X)$ its convex hull, i.e., the smallest convex set C satisfying $X \subseteq C$.

The indicator function of X is noted $\delta(\cdot|X)$ and equals zero if its argument belongs to X and $+\infty$ otherwise. Its convex conjugate is therefore given by $\delta^*(\boldsymbol{\pi}|X) = \sup\{\boldsymbol{\pi}^T\boldsymbol{x} : \boldsymbol{x} \in X\}$. If X is a convex polytope, we note $\text{vert}\,(X)$ the set of its extreme points. Finally, for a logical proposition $\mathcal{E}$, function $\mathbf{1}(\mathcal{E})$ equals one if $\mathcal{E}$ is true and zero otherwise.

2 Problem description

2.1 General setting

As anticipated, our goal is to solve problem (2SR-P) with objective uncertainty, convex constraints and mixed-integer first and second stages.

For the sake of clarity, let us first introduce several sets. Set $I = \{1, \ldots, n_1\}$ denotes the set of indices for the first-stage variables, and is partitioned into two sets I_B and I_C: variables whose index belongs to I_B are required to take binary values, while those whose index belongs to I_C are continuous variables, i.e., wlog, $\mathcal{X} \subset \mathbb{R}^{|I_C|} \times \{0,1\}^{|I_B|}$. Similarly, we introduce set $J = \{1, \ldots, n_2\}$ as the indices for the second-stage variables and partition this set into J_B and J_C, i.e., wlog, $\mathcal{Y} \subset \mathbb{R}^{|J_C|} \times \{0,1\}^{|J_B|}$. Sets I_B and J_B, which correspond to binary first and second-stage binary variables, may be defined in the same way in case of general integer variables; all the results presented in the paper directly extend to the integer case as well. Finally, we introduce set $U = \{1, \ldots, n_3\}$ as the index set for the uncertain variables, i.e., $\Xi \subset \mathbb{R}^{n_3}$.

We now explicit some assumptions on the problem.

Assumption 1 (Objective uncertainty) *For all* $\boldsymbol{\xi} \in \Xi$ *and* $\boldsymbol{x} \in \overline{\mathcal{X}}$, $\mathcal{Y}(\boldsymbol{\xi}, \boldsymbol{x}) = \mathcal{Y}(\boldsymbol{x})$.

Assumption 2 (Convexity)

1. $\overline{\mathcal{X}}$ *is compact and convex;*
2. *The uncertainty set* Ξ *is a finite-dimensional, bounded convex set;*
3. *For all* $\boldsymbol{x} \in \overline{\mathcal{X}}$, $\overline{\mathcal{Y}}(\boldsymbol{x})$ *is a finite-dimensional, bounded convex set;*
4. *The objective function* f *is a concave function of the uncertainty and a convex function of the first- and second-stage decisions, i.e.,* $f_{\boldsymbol{x},\boldsymbol{y}} : \boldsymbol{\xi} \mapsto f(\boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{y})$ *is a concave function for all fixed* $\boldsymbol{x} \in \overline{\mathcal{X}}$ *and* $\boldsymbol{y} \in \overline{\mathcal{Y}}(\boldsymbol{x})$ *and* $f_{\boldsymbol{\xi}} : (\boldsymbol{x}, \boldsymbol{y}) \mapsto f(\boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{y})$ *is a convex function for all fixed* $\boldsymbol{\xi} \in \Xi$.

Assumption 3 (Complete recourse) *For every (relaxed) first-stage decision, there exists at least one feasible second-stage decision, i.e., for every* $\boldsymbol{x} \in \overline{\mathcal{X}}$, $\mathcal{Y}(\boldsymbol{x})$ *is a non-empty set.*

Assumption 4 (Boundedness)

1. *The objective function* f *is bounded over the first- and second-stage feasible space, i.e., for all fixed* $\boldsymbol{\xi} \in \Xi$, $\{(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x} \in \overline{\mathcal{X}}, \boldsymbol{y} \in \overline{\mathcal{Y}}(\boldsymbol{x})\} \subseteq \mathrm{dom}\,(f_{\boldsymbol{\xi}})$
2. *For all* $(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x} \in \overline{\mathcal{X}}$ *and* $\boldsymbol{y} \in \overline{\mathcal{Y}}(\boldsymbol{x})$, $\mathrm{relint}(\Xi) \cap \mathrm{dom}\,(f_{\boldsymbol{x},\boldsymbol{y}}) \neq \emptyset$

Assumption 5 (Separability) *Let* $Q = \{1, \ldots, q\}$.

1. *The objective function* f *can be expressed as a sum of* q *functions, i.e., there exist* q *functions* $(\psi_i : \mathbb{R}^{|U|+|I|+|J|} \to \mathbb{R})_{i \in Q}$ *such that* $f(\boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{y}) = \sum_{i \in Q} \psi_i(\boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{y})$ *for all* $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})$ *and all* $\boldsymbol{\xi} \in \Xi$.

2. *For all $i \in Q$, ψ_i is separable in $\boldsymbol{\xi}$ and $(\boldsymbol{x}, \boldsymbol{y})$ meaning that there exists functions $(w_i : \mathbb{R}^{|U|} \to \mathbb{R})_{i\in Q}$ and $(\varphi_i : \mathbb{R}^{|I|+|J|} \to \mathbb{R})_{i\in Q}$ such that $\psi_i(\boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{y}) = w_i(\boldsymbol{\xi})\varphi_i(\boldsymbol{x}, \boldsymbol{y})$. In addition, we assume that $w_i(\cdot)$ is a concave function and $\varphi_i(\cdot)$ is a convex function.*

A few remarks regarding these assumptions are necessary. First, note that Assumptions 1 and 2 are here to define what we refer to as *convex mixed-integer robust problems with objective uncertainty*. It is important to highlight that the word "convex" is here to suggest that all involved functions are convex with respect to the first- and second-stage variables. Yet, in general, even under these assumptions, problem (2SR-P) may fail to have a straightforward convex MINLP formulation, as function $h : \boldsymbol{x} \mapsto \max_{\boldsymbol{\xi}\in\Xi} \min_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} f(\boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{y})$ is not necessarily a convex function over the continuous relaxation of $\mathcal{X}$. We give here a small example.

Example 1 (nonconvex MINLP) Consider the following first- and second-stage feasible regions:

$$\mathcal{X} = [0,1] \text{ and } \mathcal{Y}(x) = \left\{ \boldsymbol{y} \in \{0,1\}^2 \;\middle|\; \begin{array}{rl} y_1 + y_2 & \leq 1 \\ y_1 & \leq 1 - x \end{array} \right\}$$

By inspection, we have that $(y_1, y_2) = (0,0)$ and $(y_1, y_2) = (0,1)$ are always feasible second-stage solutions, while $(y_1, y_2) = (1,0)$ is feasible only when $x = 0$. Fixing the uncertainty set $\Xi = [0,1]$, we take interest in the following *convex mixed-integer two-stage robust problem*:

$$\min_{x\in[0,1]} h(x) \text{ with } h : x \mapsto \max_{\xi\in[0,1]} \min_{(y_1,y_2)\in\mathcal{Y}(x)} \xi(-2y_1 + y_2 + 1)$$

Though every involved functions are convex (in fact, affine), the following holds:

$$h(x) = \begin{cases} \max_{\xi\in[0,1]} \min\{\xi;\ 2\xi;\ -\xi\} = 0 & \text{if } x = 0 \\ \max_{\xi\in[0,1]} \min\{\xi;\ 2\xi\} = 1 & \text{if } x > 0 \end{cases} = \mathbf{1}(x > 0)$$

Clearly, h fails to be convex over $[0,1]$ which ends our example.

Assumption 3 is a standard assumption in the two-stage optimization literature, and is known to be easy to enforce as soon as the considered problem is bounded, which is implied by Assumption 4.1. Assumption 4.2 is not restrictive in practice, and will be used in the proof of lemma (2).

Finally, Assumption 5 is structural to our work, and implies the following remark.

Remark 1 Without loss of generality, for all $i \in Q$, we can assume that $\varphi_i(\cdot)$ is a convex function (at most affine).

Proof Let $i \in Q$ such that $\varphi_i(\cdot)$ is concave, then, to fulfill Assumption 2.1, $w_i(\boldsymbol{\xi})$ must be negative forall $\boldsymbol{\xi} \in \Xi$. Thus, one may equivalently replace $w_i(\cdot)$ by $-w_i(\cdot)$ and $\varphi_i(\cdot)$ by $-\varphi_i(\cdot)$. □

Remark 2 For all $i \in Q$ such that $\varphi_i(\cdot)$ (resp. $w_i(\cdot)$) is not single-signed, then $w_i(\cdot)$ (resp. $\varphi_i(\cdot)$) is affine.

Remark 3 For all $i \in Q$ such that $\varphi_i(\cdot)$ (resp. $w_i(\cdot)$) is not affine, then $w_i(\cdot)$ (resp. $\varphi_i(\cdot)$) is a non-negative function.

Note that Assumption 5 could be relaxed to address situations in which, for $i \in Q$ such that $\varphi_i(\cdot)$ (resp. $w_i(\cdot)$) is affine, there is no restriction on the concavity (resp. convexity) of the associated $w_i(\cdot)$ (resp. $\varphi_i(\cdot)$).

Example 2 (Fulfilling Assumption 5) We give here some examples of functions which satisfy Assumption 5. For simplicity, we denote $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$.

- Uncertain linear functions of the form $(\boldsymbol{\xi}, \boldsymbol{z}) \mapsto \boldsymbol{\xi} \boldsymbol{A} \boldsymbol{z}$ where $\boldsymbol{A}$ is a given real matrix;
- Diagonal uncertain convex quadratic form $(\boldsymbol{\xi}, \boldsymbol{z}) \mapsto \boldsymbol{z}^T \mathrm{diag}(\boldsymbol{\xi}) \boldsymbol{z}$ where $\boldsymbol{\xi} \geq 0$;
- Uncertain positively weighted sum of convex functions of the form $(\boldsymbol{\xi}, \boldsymbol{z}) \mapsto \sum_{i \in Q} \xi_i \varphi_i(\boldsymbol{z})$ with $\Xi \subset \mathbb{R}_+^{|U|}$, e.g., $(\boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{y}) \mapsto \sum_{i \in Q} \xi_i x_i^2 / y_i$ with $\boldsymbol{y} \geq \boldsymbol{0}$.

Example 3 (Violating Assumption 5) We give here some examples of functions which do not satisfy Assumption 5.

- Non-concave functions of the uncertainty, e.g., $(\boldsymbol{\xi}, \boldsymbol{z}) \mapsto ||\boldsymbol{z} - \boldsymbol{\xi}||$ for any given norm;
- General uncertain quadratic form $(\boldsymbol{\Sigma}, \boldsymbol{z}) \mapsto \boldsymbol{z}^T \boldsymbol{\Sigma} \boldsymbol{z}$ even with $\boldsymbol{\Sigma} \succeq 0$ (unless $\Xi \cap \mathbb{R}_-^{|U|} = \emptyset$)

In the following lemma, we finally state the class of problems we consider.

Lemma 1 *Under Assumptions (1)-(5), there exists* $[\boldsymbol{l}, \boldsymbol{u}] \subset \mathbb{R}^{|I|+|J|}$ *such that* (2SR-P) *is equivalent to the following problem:*

$$\inf_{\boldsymbol{x} \in \mathcal{X} \cap [\boldsymbol{l}, \boldsymbol{u}]} \sup_{\boldsymbol{\xi} \in \Xi} \inf_{(\boldsymbol{t}, \boldsymbol{y}) \in \mathcal{Y}'(\boldsymbol{x})} \sum_{i \in Q} w_i(\boldsymbol{\xi}) t_i \tag{2SRO-P}$$

with $\mathcal{Y}'(\boldsymbol{x})$ *such that* $\mathcal{Y}(\boldsymbol{x}) = \mathrm{proj}_{\boldsymbol{y}}(\mathcal{Y}'(\boldsymbol{x}))$ *and* $\overline{\mathcal{Y}}'(\boldsymbol{x})$ *is a convex and finite-dimensional set.*

Proof The existence of the hyper-rectangle $[\boldsymbol{l}, \boldsymbol{u}]$ is trivial as $\mathcal{X}$ is assumed to be bounded (Assumption 2.1). Moreover, the following equality holds:

$$\inf_{\boldsymbol{y}} \left\{ \sum_{i\in Q} w_i(\boldsymbol{\xi})\varphi_i(\boldsymbol{x},\boldsymbol{y}) : \boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x}) \right\} = \inf_{\boldsymbol{y},\boldsymbol{t}} \left\{ \sum_{i\in Q} w_i(\boldsymbol{\xi}) t_i : \boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x}), t_i = \varphi_i(\boldsymbol{x},\boldsymbol{y}), \forall i \in Q \right\}$$

However, the optimization problem on the right side of the equality may fail to be convex if there exists $i \in Q$ such that φ_i is not affine. Let $Q^A \subseteq Q$ be the set of indices for which φ_i is affine. By Assumption 5, for all $i \in Q \backslash Q^A$, we have $w_i(\cdot) \geq 0$ and thus constraint "$t_i = \varphi_i(\boldsymbol{x},\boldsymbol{y})$" may be equivalently replaced by "$t_i \geq \varphi_i(\boldsymbol{x},\boldsymbol{y})$", which is convex. We therefore can choose

$$\mathcal{Y}'(\boldsymbol{x}) = \left\{ (\boldsymbol{t},\boldsymbol{y}) : \begin{array}{c} \boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x}) \\ t_i = \varphi_i(\boldsymbol{x},\boldsymbol{y}) \;\; \forall i \in Q^A \\ t_i \geq \varphi_i(\boldsymbol{x},\boldsymbol{y}) \;\; \forall i \in Q\backslash Q^A \end{array} \right\}$$

For every $\boldsymbol{x} \in \overline{\mathcal{X}}$, the continuous relaxation of $\mathcal{Y}'(\boldsymbol{x})$ is convex and non-empty (Assumption 3); by construction, it is also finite dimensional. □

In what remains, we will assume to know a hyper-rectangle $[\boldsymbol{l}, \boldsymbol{u}]$ as described in lemma 1.

2.2 Special case: linear and binary setting

We complete the introduction by discussing the special case of (2SRO-P) under the following additional assumptions:

1. $\mathcal{X}, \Xi$ and $\boldsymbol{x} \mapsto \mathcal{Y}(\boldsymbol{x})$ are defined by linear constraints;
2. there exists a matrix $\boldsymbol{A} \in \mathbb{R}^{|U|\times|Q|}$ such that $\forall i \in Q, w_i(\boldsymbol{\xi}) = \boldsymbol{\xi}^T \boldsymbol{a}^i$; and
3. linking constraints are defined by functions $g(\boldsymbol{x},\boldsymbol{y})$ that do not depend on first-stage variables in I_c.

In a recent paper Arslan and Detienne (2021), the authors observed that, for this variant of the problem, the inner minimization $\min_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} \boldsymbol{\xi}^T \boldsymbol{A}\boldsymbol{y}$ can be equivalently replaced by $\min_{\boldsymbol{y}\in\text{conv}(\mathcal{Y}(\boldsymbol{x}))} \boldsymbol{\xi}^T \boldsymbol{A}\boldsymbol{y}$, i.e., the second-stage feasible space can be substituted by its convex hull. This allows to transform the min-max-min problem into a min-max problem using the well known minimax theorem for convex sets. Assuming that Ξ is expressed as $\{\boldsymbol{\xi} \in \mathbb{R}_+^{|U|} : \boldsymbol{F}\boldsymbol{\xi} \leq \boldsymbol{d}\}$, the inner maximization problem is dualized so as to obtain the following equivalent problem:

$$\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{\lambda}} \boldsymbol{d}^T \boldsymbol{\lambda} \tag{1}$$
$$\text{subject to } \boldsymbol{x} \in \mathcal{X} \tag{2}$$
$$\boldsymbol{y} \in \text{conv}\left(\mathcal{Y}(\boldsymbol{x})\right) \tag{3}$$
$$\boldsymbol{F}^T \boldsymbol{\lambda} \geq \boldsymbol{A}\boldsymbol{y} \tag{4}$$
$$\boldsymbol{\lambda} \geq \boldsymbol{0}, \tag{5}$$

where $\boldsymbol{\lambda}$ are the dual variables associated to the inner maximization problem. Note that, besides the integrality requirements on the variables, the only non-convex constraint is (3). By exploiting a reformulation already used in Sherali and Fraticelli (2002), Sherali and Zhu (2006) and Li and Grossmann (2019) for two-stage stochastic optimization problems with mixed-integer first and second stage, Arslan and Detienne (2021) showed that this constraint may be equivalently enforced as

$$(\boldsymbol{x}, \boldsymbol{y}) \in \operatorname{conv}(S) \cap \{(\boldsymbol{x}', \boldsymbol{y}') : \boldsymbol{x} = \boldsymbol{x}'\}, \tag{6}$$

where $S = \{(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x} \in \{0,1\}, \boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})\}$. The obtained reformulation is then solved by means of a branch-and-price algorithm where the convex hull of set S is expressed in terms of convex combinations of its extreme points, and branching is performed on the first-stage variables only.

3 A hull-relaxation-based branch-and-bound algorithm

In this section we present our main contribution and its theoretical foundations. In the same spirit as in Arslan and Detienne (2021), we first turn problem (2SRO-P) from a min-max-min problem to a min-max problem in our mixed-integer and convex context. Then, since linear duality does not apply in our setting, we resort to Fenchel duality to obtain a reformulation of the problem. Similarly to the linear and binary case, we then replace the counterpart of constraint (3) by (6), though this only provides a relaxation of the problem in the general setting. This relaxation is thus embedded into a branch-and-bound scheme to obtain an optimal solution of (2SRO-P).

3.1 Problem reformulation

The following lemma extends to the mixed-integer and convex context the result given in Arslan and Detienne (2021).

Lemma 2 (Single-stage reformulation) *Problem* (2SRO-P) *is equivalent to the following problem:*

$$\inf_{(\boldsymbol{x},\boldsymbol{t},\boldsymbol{y}) \in F} \sup_{\boldsymbol{\xi} \in \Xi} \sum_{i \in Q} w_i(\boldsymbol{\xi}) t_i \tag{7}$$

with $F = \{(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}) : \boldsymbol{x} \in \mathcal{X} \cap [\boldsymbol{l}, \boldsymbol{u}], (\boldsymbol{t}, \boldsymbol{y}) \in \operatorname{conv}\left(\mathcal{Y}'(\boldsymbol{x})\right)\}$.

Proof This lemma relies on the same arguments as those employed in Arslan and Detienne (2021): first, the feasible space of the inner minimization problem is replaced by its convex hull by linearity of the objective function and convexity of the feasible region. By assumption 2.2 and Lemma 2, both Ξ and $\operatorname{conv}\left(\mathcal{Y}'(\boldsymbol{x})\right)$ (for all $\boldsymbol{x} \in \mathcal{X}$) are convex and finite dimensional. Thus, the result in Perchet and Vigeral (2015) can be used to turn the inner $\sup - \inf$ into an $\inf - \sup$ problem. This achieves the proof. □

The inner maximization problem may be turned into a minimization problem by use of Fenchel duality, as done in Ben-Tal et al (2009). In the following proposition, we therefore derive a general reformulation of problem (2SRO-P).

Proposition 1 (Deterministic reformulation) *Problem* (2SRO-P) *is equivalent to the following problem:*

$$\inf_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{t},(\boldsymbol{v}^i)_{i\in Q},\boldsymbol{\xi}} \delta^*(\boldsymbol{\xi}|\Xi) - \sum_{i\in Q}(t_i w_i)_* \left(\boldsymbol{v}^i\right) \tag{8}$$

$$\text{subject to } \boldsymbol{x} \in \mathcal{X} \cap [\boldsymbol{l}, \boldsymbol{u}] \tag{9}$$

$$(\boldsymbol{t}, \boldsymbol{y}) \in \text{conv}\left(\mathcal{Y}'(\boldsymbol{x})\right) \tag{10}$$

$$\sum_{i\in Q} \boldsymbol{v}^i = \boldsymbol{\xi} \tag{11}$$

$$\boldsymbol{v}^i \in \mathbb{R}^{|U|} \qquad \forall i \in Q \tag{12}$$

Proof By a direct application of Fenchel duality and some conjugate calculus results, the following holds

$$\begin{aligned}
\sup_{\boldsymbol{\xi}\in\Xi} \sum_{i\in Q} t_i w_i(\boldsymbol{\xi}) &= \sup_{\boldsymbol{\xi}\in\mathbb{R}^{|U|}} \left\{ \sum_{i\in Q} t_i w_i(\boldsymbol{\xi}) - \delta(\boldsymbol{\xi}|\Xi) \right\} = \inf_{\boldsymbol{\xi}\in\mathbb{R}^{|U|}} \left\{ \delta^*(\boldsymbol{\xi}|\Xi) - \left(\sum_{i\in Q} t_i w_i(\boldsymbol{\xi}) \right)_* \right\} \\
&= \inf_{\boldsymbol{\xi}\in\mathbb{R}^{|U|}} \left\{ \delta^*(\boldsymbol{\xi}|\Xi) - \sup_{\boldsymbol{v}^i\in\mathbb{R}^{|U|}, i\in Q} \left\{ \sum_{i\in Q} (t_i w_i)_* \left(\boldsymbol{v}^i\right) : \sum_{i\in Q} \boldsymbol{v}^i = \boldsymbol{\xi} \right\} \right\} \\
&= \inf \left\{ \delta^*(\boldsymbol{\xi}|\Xi) - \sum_{i\in Q} (t_i w_i)_* \left(\boldsymbol{v}^i\right) : \sum_{i\in Q} \boldsymbol{v}^i = \boldsymbol{\xi}, \boldsymbol{v}^i \in \mathbb{R}^{|U|}, i \in Q, \boldsymbol{\xi} \in \mathbb{R}^{|U|} \right\}
\end{aligned}$$

See also appendix A for more details on conjugate calculus. □

Remark 4 Assume wlog that $|Q| = |U|$. If, for all $i \in Q$, $w_i(\boldsymbol{\xi}) = w_i(\xi_i)$, then problem (2SRO-P) is equivalent to

$$\inf_{(\boldsymbol{x},\boldsymbol{t},\boldsymbol{y})\in F} \left\{ \delta^*(\boldsymbol{\xi}|\Xi) - \sum_{i\in Q} (t_i w_i)_* \left(\boldsymbol{\xi}\right) \right\} \tag{13}$$

Remark 5 Let $i \in Q$ such that $w_i(\cdot)$ is affine, i.e., $w_i(\boldsymbol{\xi}) = (\boldsymbol{r}^i)^T\boldsymbol{\xi} + r_{i0}$. Problem (2SRO-P) is equivalent to

$$\inf_{(\boldsymbol{x},\boldsymbol{t},\boldsymbol{y})\in F} \left\{ \delta^*(\boldsymbol{R}\boldsymbol{t}|\Xi) + \boldsymbol{r}_0^T \boldsymbol{t} \right\} \tag{14}$$

Proof Indeed, we have

$$(t_i w_i)_*(\boldsymbol{v}) = \inf_{\boldsymbol{\xi}\in\mathbb{R}^{|U|}} \{ \boldsymbol{v}^T\boldsymbol{\xi} - t_i((\boldsymbol{r}^i)^T\boldsymbol{\xi} + r_{i0}) \} = \begin{cases} -t_i r_{i0} & \text{if } \boldsymbol{v} = t_i \boldsymbol{r}^i \\ -\infty & \text{otherwise.} \end{cases}$$

□

These results show that although the reformulation for the general case adds $|Q| \times |U|$ continuous variables, for some relevant cases these additional variables can be omitted. In particular this is true in case all the $w_i(\cdot)$ functions are either separable or affine.

3.2 Relaxation

Note that the deterministic reformulation presented above still is not, in general, a convex MINLP and that no tractable, compact form is known in the general case. To overcome this drawback, we replace constraint (10) by the following requirement:

$$(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}) \in \operatorname{conv}(S) \cap \{(\boldsymbol{x}', \boldsymbol{t}', \boldsymbol{y}') : \boldsymbol{x} = \boldsymbol{x}'\}$$
$$\text{with } S = \left\{ (\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}) : \begin{array}{ll} l_j \leq x_j \leq u_j & \forall j \in I \\ x_j \in \{0,1\} & \forall j \in I_B \\ (\boldsymbol{t}, \boldsymbol{y}) \in \mathcal{Y}'(\boldsymbol{x}) & \end{array} \right\} \quad (15)$$

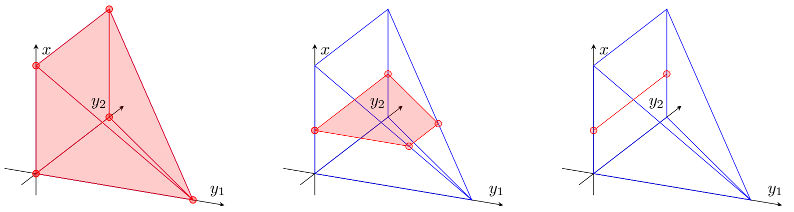
The problem obtained from this substitution is thus

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}, \boldsymbol{y}, (\boldsymbol{v}^i)_{i \in Q}, \boldsymbol{\xi}} & \delta^*(\boldsymbol{\xi} | \Xi) - \sum\limits_{i \in Q} (t_i w_i)_* \left(\boldsymbol{v}^i\right) \\ \text{subject to} & \boldsymbol{x} \in \mathcal{X} \cap [\boldsymbol{l}, \boldsymbol{u}] \\ & (\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}) \in \operatorname{conv}(S) \\ & \sum\limits_{i \in Q} \boldsymbol{v}^i = \boldsymbol{\xi} \\ & \boldsymbol{v}^i \in \mathbb{R}^{|U|} \qquad \forall i \in Q \\ & \boldsymbol{\xi} \in \mathbb{R}^{|U|} \end{array} \quad \text{(P)}$$

It is clear that, for any fixed $\bar{\boldsymbol{x}} \in \mathcal{X}$, we have $\{\bar{\boldsymbol{x}}\} \times \mathcal{Y}'(\bar{\boldsymbol{x}}) = S \cap \{(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}) : \boldsymbol{x} = \bar{\boldsymbol{x}}\}$, and that the same holds even for $\bar{\boldsymbol{x}} \in \overline{\mathcal{X}}$. However, as shown, e.g., in Sherali and Zhu (2006), the convexified counterpart does not hold, in the sense that the inclusion "$\{\bar{\boldsymbol{x}}\} \times \operatorname{conv}(\mathcal{Y}(\bar{\boldsymbol{x}})) \subseteq \operatorname{conv}(S) \cap \{(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}) : \boldsymbol{x} = \bar{\boldsymbol{x}}\}$" may be strict. Example 4 below illustrates this case.

Example 4 (Hull relaxation) We consider the first- and second-stage feasible sets introduced in Example 1. In Figure (1a), we represent the convex hull of S. For a fixed first-stage decision $\bar{x}$ (here, $\bar{x} = 0.4$), Figure (1b) reports the feasible points for constraint (15), whereas Figure (1c) describes the exact shape of $\operatorname{conv}(\mathcal{Y}(\bar{x}))$. The figure shows an example in which inclusion is strict. In addition, note that, whenever $\bar{x}$ attains its bounds (i.e., $\bar{x} \in \{0,1\}$), $\{\bar{x}\} \times \operatorname{conv}(\mathcal{Y}(\bar{x})) = \operatorname{conv}(S) \cap \{(x, \boldsymbol{y}) : x = \bar{x}\}$ holds.

The following Lemma follows from the considerations above.



(a) conv (S) (b) conv $(S) \cap \{x = \bar{x}\}$ (c) $\{\bar{x}\} \times$ conv $(\mathcal{Y}(\bar{x}))$

Fig. 1: Graphical representation of different sets from example 1

Lemma 3 (Lower-bounding property) *Denoting by $v(\bullet)$ the optimal objective value of problem $\bullet$, we have:*

$$v(\text{P}) \leq v(\text{2SRO-P})$$

In other words, (P) is a relaxation of (2SRO-P). In the next proposition, we introduce a condition under which a feasible solution for problems (P) is feasible for problem (2SRO-P) as well.

Proposition 2 *If $\bar{\boldsymbol{x}} \in \text{vert}([\boldsymbol{l}, \boldsymbol{u}])$, then*

$$\{\bar{\boldsymbol{x}}\} \times \text{conv}\left(\mathcal{Y}'(\bar{\boldsymbol{x}})\right) = \text{conv}(S) \cap \{(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}) : \boldsymbol{x} = \bar{\boldsymbol{x}}\}$$

Proof Let $\bar{\boldsymbol{x}} \in \text{vert}([\boldsymbol{l}, \boldsymbol{u}])$ and let $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{y}}) \in \text{conv}(S) \cap \{(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}) : \boldsymbol{x} = \bar{\boldsymbol{x}}\}$. Then, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{y}})$ can be expressed as a (finite) convex combination of points of conv (S) (Carathéodory's theorem), i.e.,

$$(\hat{\boldsymbol{x}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{y}}) = \sum_{e \in E} (\bar{\mathbf{x}}^e, \bar{\mathbf{t}}^e, \bar{\mathbf{y}}^e)\alpha_e$$

where E is a given index list of such elements of conv (S). Assume that there exists $j \in I$ and $i \in E$ such that $\bar{\mathrm{x}}^i_j \neq \bar{x}_j$. If $\bar{\mathrm{x}}^i_j > \bar{x}_j$, condition $\bar{\mathbf{x}}^i \in \text{conv}(S)$ implies that $\bar{x}_j = l_j$. Hence, $\alpha_i = 0$ since $\bar{\mathrm{x}}^k_j \geq l_j \ \forall k \in E$. The same argument shows that $\bar{\mathrm{x}}^i_j < \bar{x}_j$ implies $\alpha_i = 0$. Thus, for each $e \in E$ such that $\alpha_e > 0$, we must have $\bar{\mathbf{x}}^e = \bar{\boldsymbol{x}}$. This implies that $(\bar{\mathbf{t}}^e, \bar{\mathbf{y}}^e) \in \mathcal{Y}'(\bar{\boldsymbol{x}})$ and thus $\sum_{e \in E} (\bar{\mathbf{t}}^e, \bar{\mathbf{y}}^e)\alpha_e \in \text{conv}\left(\mathcal{Y}'(\bar{\boldsymbol{x}})\right)$. □

Corollary 1 (Tightness condition) *Let X^* be the set of optimal first-stage decisions of problem* (P)*. Then:*

$$X^* \cap \text{vert}([\boldsymbol{l}, \boldsymbol{u}]) \neq \emptyset \Rightarrow v(\text{P}) = v(\text{2SRO-P})$$

Proof Let $(\boldsymbol{x}^*, \boldsymbol{t}^*, \boldsymbol{y}^*)$ be an optimal solution of (P) with $\boldsymbol{x}^* \in \text{vert}([\boldsymbol{l}, \boldsymbol{u}])$. From Proposition 2, it is also feasible for problem (2SRO-P). Thus, Lemma 3 implies optimality for problem (2SRO-P). □

This result directly implies Corollary 2 which states that, in the special case where the first-stage variables are all binary, problem (P) is always an exact reformulation of (2SRO-P).

Corollary 2 (Tightness condition/binary case) *If the first-stage decisions are all binary, i.e.,* $I_C = \emptyset$*, then*

$$v(\text{P}) = v(\text{2SRO-P})$$

Proof In this case, $[\boldsymbol{l}, \boldsymbol{u}] = [\boldsymbol{0}, \boldsymbol{1}]$, hence any optimal first-stage solution $\boldsymbol{x}^*$ satisfies $\boldsymbol{x}^* \in \{0,1\}^{I_B} = \text{vert}([\boldsymbol{l}, \boldsymbol{u}])$ which, by Corollary 1, proves the result. □

3.3 Enumerative algorithm

We now present an exact method for solving problem (2SRO-P). Motivated by Corollary 1, the main idea of the algorithm is to determine an optimal value of the first-stage variables, and then derive the corresponding optimal values for the second-stage variables. To this aim, we developed a branch-and-bound algorithm in which we relax both the integrality of the x and requirement (10). To ensure feasibility, we perform a spatial branching on the x variables, until each of them attains either its lower or upper bound. The algorithm stores the best feasible solution found (the *incumbent* solution) which is returned when the algorithm stops.

3.3.1 Node solution

Let p denote a generic node of the branch-and-bound tree, associated with bounds $\boldsymbol{l}^p$ and $\boldsymbol{u}^p$ on first-stage variables.

A lower bound on the optimal solution value of node p can be computed solving the following problem:

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{t},\boldsymbol{y},(\boldsymbol{v}^i)_{i\in Q},\boldsymbol{\xi}} \quad & \delta^*(\boldsymbol{\xi}|\Xi) - \sum_{i\in Q} (t_i w_i)_* \left(\boldsymbol{v}^i\right) \\
\text{subject to} \quad & \boldsymbol{x} \in \overline{\mathcal{X}} \cap [\boldsymbol{l}^p, \boldsymbol{u}^p] \\
& (\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}) \in \text{conv}\left(S^p\right) \\
& \sum_{i\in Q} \boldsymbol{v}^i = \boldsymbol{\xi} \\
& \boldsymbol{v}^i \in \mathbb{R}^{|U|} \qquad \forall i \in Q \\
& \boldsymbol{\xi} \in \mathbb{R}^{|U|}
\end{aligned} \tag{LB^p}$$

where $S^p = \{(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}) : \boldsymbol{l}^p \le \boldsymbol{x} \le \boldsymbol{u}^p,\ x_j \in \{0,1\}, \forall j \in I_B,\ (\boldsymbol{t}, \boldsymbol{y}) \in \mathcal{Y}'(\boldsymbol{x})\}$. This problem is exactly the continuous relaxation of problem (P) where the bounds $\boldsymbol{l}$ and $\boldsymbol{u}$ have been replaced by $\boldsymbol{l}^p$ and $\boldsymbol{u}^p$. Note that at the root node we have $\boldsymbol{l}^0 = \boldsymbol{l}$ and $\boldsymbol{u}^0 = \boldsymbol{u}$.

Let $(\boldsymbol{x}^{p*}, \boldsymbol{t}^{p*}, \boldsymbol{y}^{p*}, (\boldsymbol{v}^{i^{p*}})_{i\in Q}, \boldsymbol{\xi}^{p*})$ be an optimal solution of problem LB^p. If $v(\text{LB}^p)$ is greater than or equal to the cost of the incumbent, the node is fathomed by bounding. Otherwise, we distinguish three cases:

- if $\boldsymbol{x}^{p*} \in \text{vert}\,([\boldsymbol{l}^p, \boldsymbol{u}^p])$, by Proposition 2, this solution is optimal for the current node. Hence, the node is fathomed by optimality and the incumbent is updated;
- if $\boldsymbol{x}^{p*} \in \mathcal{X} \setminus \text{vert}\,([\boldsymbol{l}^p, \boldsymbol{u}^p])$, we compute a feasible solution for (2SRO-P) by solving the following model in which the first-stage variables are fixed to $\boldsymbol{x}^{p*}$:

$$
\begin{aligned}
\min_{\boldsymbol{t},\boldsymbol{y},(\boldsymbol{v}^i)_{i\in Q},\boldsymbol{\xi}} \quad & \delta^*(\boldsymbol{\xi}|\Xi) - \sum_{i\in Q}(t_i w_i)_* \left(\boldsymbol{v}^i\right) \\
\text{subject to} \quad & (\boldsymbol{t},\boldsymbol{y}) \in \text{conv}\left(\mathcal{Y}'(\boldsymbol{x}^{p*})\right) \\
& \sum_{i\in Q} \boldsymbol{v}^i = \boldsymbol{\xi} \\
& \boldsymbol{v}^i \in \mathbb{R}^{|U|} \qquad \forall i \in Q \\
& \boldsymbol{\xi} \in \mathbb{R}^{|U|}
\end{aligned}
\tag{UB^p}
$$

  Note that, in this case, $\boldsymbol{x}^{p*}$ corresponds to a feasible first-stage solution; hence, by Assumption 3, problem UB^p is always feasible, and possibly the incumbent is updated. If $v(LB^p) = v(UB^p)$ then node p is solved; otherwise, we perform a branching;
- if $\boldsymbol{x}^{p*} \in \overline{\mathcal{X}} \setminus \mathcal{X}$, we branch.

In the last case, before branching, one can try to round $\boldsymbol{x}^{p*}$; if the resulting point is in $\mathcal{X}$, a feasible solution for (2SRO-P) can be computed. In our experiments, every fractional value for x_j^{p*} with $j \in I_B$ was rounded to the closest integer while variables x_j^{p*} with $j \in I_C$ were not rounded.

3.3.2 Branching

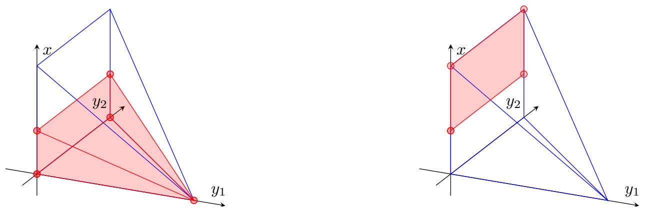
We now describe how to select the branching variable at node p. For each first-stage variable, say with index $j \in I$, we compute the minimum distance of x_j^{p*} from one of its bounds at the node, i.e., we evaluate:

$$\theta_j^p = \min\{x_j^{p*} - l_j^p;\ u_j^p - x_j^{p*}\}.$$

For branching, we give priority to binary variables that do not attain their bound. Otherwise, we resort to spatial branching on continuous variables. In both cases, we select the variable with maximum θ_j^p value, i.e., we select variable $x_{\overline{j}}$ such that,

$$\overline{j} \in \begin{cases} \text{argmax}\{\theta_j^p : j \in I_B\} & \text{if } \exists j \in I_B, \theta_j > 0 \\ \text{argmax}\{\theta_j^p : j \in I_C\} & \text{otherwise.} \end{cases}$$

If $\overline{j} \in I_B$, then a standard binary branching is executed. Otherwise, we resort to spatial branching, and generate two descendant nodes by imposing



(a) Left child ($x \leq \beta$) (b) Right child ($x \geq \beta$)

Fig. 2: Branching on continuous variable x from example 1

$x_{\bar{j}} \leq x_{\bar{j}}^{p*}$ for the left node and $x_{\bar{j}} \geq x_{\bar{j}}^{p*}$ for the right one. We associate to each node the lower bound value of the current node $v(\text{LB}^p)$ and insert them in a list of open nodes. At each iteration, we extract from the list one node with minimum lower bound value, halting the algorithm stops when the list is empty.

Example 5 Figure 2 illustrates the left and right child obtained by spatial branching on $x \leq \beta$ and $x \geq \beta$ from example 1 (here, $\beta = 0.4$). Clearly, the right child allows the same recourse decisions as in $\mathcal{Y}(x)$ for all $x \geq \beta$. The left child, however, still allows second-stage decisions that could end up being infeasible in the original problem. In particular, $(\boldsymbol{x}, \boldsymbol{y}) = (\varepsilon, 1-\varepsilon, 0)$ with $\varepsilon \in (0, \beta]$ is feasible for (LB^p) but not for (2SRO-P).

3.3.3 Single-stage heuristic

We now present a heuristic procedure that can be used at the root node to warm start the branch-and-bound algorithm. This heuristic is based on the definition of a single-stage version of (2SR-P), in which both first- and second-stage variables are simultaneously optimized. The resulting problem can be formulated using the following MINLP:

$$\min \delta^*(\boldsymbol{\xi}|\Xi) - \sum_{i \in Q} (t_i w_i)_* \left(\boldsymbol{v}^i\right) \tag{16}$$

$$\text{subject to } \boldsymbol{x} \in \mathcal{X} \cap [\boldsymbol{l}, \boldsymbol{u}] \tag{17}$$

$$(\boldsymbol{t}, \boldsymbol{y}) \in \mathcal{Y}'(\boldsymbol{x}) \tag{18}$$

$$\sum_{i \in Q} \boldsymbol{v}^i = \boldsymbol{\xi} \tag{19}$$

$$\boldsymbol{v}^i \in \mathbb{R}^{|U|} \qquad \forall i \in Q \tag{20}$$

Note that solving this problem is NP-hard. Let $\boldsymbol{x}^*$ denote its optimal first-stage solution with associated $(\boldsymbol{t}^*, \boldsymbol{y}^*)$ second-stage solution. An improving second stage-solution can be possibly obtained by fixing $\boldsymbol{x} = \boldsymbol{x}^*$ in problem

(2SRO-P), in the spirit of the upper bounding procedure used when solving UB^p.

3.3.4 Convergence

Given a feasible solution of (LB^p), say $(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}, \boldsymbol{V}, \boldsymbol{\xi})$, we introduce the following function

$$F(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{y}, \boldsymbol{V}, \boldsymbol{\xi}) := \delta^*(\boldsymbol{\xi}|\Xi) - \sum_{i \in Q} (t_i w_i)_* \left(\boldsymbol{v}^i\right) = F(\boldsymbol{t}, \boldsymbol{V}, \boldsymbol{\xi}),$$

that returns the solution value in the lower bounding problem.

Proposition 3 (Convergence result) *If $F \in C^0$, our branch-and-bound algorithm either finitely terminates or enters an infinite sequence of nodes for which the optimal solutions of the associated lower bounding problems converge to an optimal solution of 2SRO-P.*

Proof Let us consider the case in which the algorithm enters an infinite sequence P of nodes, indexed by p. We denote by $(\boldsymbol{l}^p, \boldsymbol{u}^p)$ the associated bounds for the $\boldsymbol{x}$ variables, and by $(\boldsymbol{x}^{p*}, \boldsymbol{t}^{p*}, \boldsymbol{y}^{p*}, \boldsymbol{V}^{p*}, \boldsymbol{\xi}^{p*})$ the optimal optimal solutions of the corresponding lower bounding problems. Since branching always reduces the domain of the $\boldsymbol{x}$ variables, then $(\boldsymbol{l}^p, \boldsymbol{u}^p)$ will converge to some values, say $(\boldsymbol{l}^*, \boldsymbol{u}^*)$, and $\boldsymbol{x}^{p*}$ will converge to a solution $\boldsymbol{x}^*$.

We first show that the sequence of optimal solutions of the lower bounding problems converges to an optimal solution of the lower bounding problem defined by bounds $(\boldsymbol{l}^*, \boldsymbol{u}^*)$.

By boundedness of problems (LB^p), there exists $P' \subseteq P$ and $(\boldsymbol{t}^*, \boldsymbol{y}^*, \boldsymbol{V}^*, \boldsymbol{\xi}^*)$, such that $\{(\boldsymbol{x}^{p*}, \boldsymbol{t}^{p*}, \boldsymbol{y}^{p*}, \boldsymbol{V}^{p*}, \boldsymbol{\xi}^{p*})\}_{p \in P'} \to (\boldsymbol{x}^*, \boldsymbol{t}^*, \boldsymbol{y}^*, \boldsymbol{V}^*, \boldsymbol{\xi}^*)$

Since $(\boldsymbol{x}^*, \boldsymbol{t}^*, \boldsymbol{y}^*, \boldsymbol{V}^*, \boldsymbol{\xi}^*)$ is the limit of a sequence of feasible points and the feasible region is closed, $(\boldsymbol{x}^*, \boldsymbol{t}^*, \boldsymbol{y}^*, \boldsymbol{V}^*, \boldsymbol{\xi}^*)$ is feasible for the lower bounding problem LB^* defined by bounds $(\boldsymbol{l}^*, \boldsymbol{u}^*)$, and thus

$$F(\boldsymbol{x}^*, \boldsymbol{t}^*, \boldsymbol{y}^*, \boldsymbol{V}^*, \boldsymbol{\xi}^*) \geq v(\mathrm{LB}^*) := F(\hat{\boldsymbol{x}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{V}}, \hat{\boldsymbol{\xi}}),$$

where $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{V}}, \hat{\boldsymbol{\xi}})$ is an optimal solution for problem LB^*. We now show that $(\boldsymbol{x}^*, \boldsymbol{t}^*, \boldsymbol{y}^*, \boldsymbol{V}^*, \boldsymbol{\xi}^*)$ is as well an optimal solution for this problem, i.e., equality holds. Assume by contradiction that $F(\boldsymbol{x}^*, \boldsymbol{t}^*, \boldsymbol{y}^*, \boldsymbol{V}^*, \boldsymbol{\xi}^*) > F(\hat{\boldsymbol{x}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{V}}, \hat{\boldsymbol{\xi}})$. For all $p \in P'$, since $P' \subseteq P$, we have $[\boldsymbol{l}^*, \boldsymbol{u}^*] \subseteq [\boldsymbol{l}^p, \boldsymbol{u}^p]$ and thus $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{V}}, \hat{\boldsymbol{\xi}})$ is feasible for (LB^p). By continuity of F we now have

$$\{F(\boldsymbol{x}^{p*}, \boldsymbol{t}^{p*}, \boldsymbol{y}^{p*}, \boldsymbol{V}^{p*}, \boldsymbol{\xi}^{p*})\}_{p \in P'} \to F(\boldsymbol{x}^*, \boldsymbol{t}^*, \boldsymbol{y}^*, \boldsymbol{V}^*, \boldsymbol{\xi}^*) > F(\hat{\boldsymbol{x}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{V}}, \hat{\boldsymbol{\xi}}),$$

which contradicts the optimality of $(\boldsymbol{x}^{p*}, \boldsymbol{t}^{p*}, \boldsymbol{y}^{p*}, \boldsymbol{V}^{p*}, \boldsymbol{\xi}^{p*})$ for some p.

We now show that the solution to which the sequence converges is a feasible solution for 2SRO-P. Given the infinite sequence of nodes, there exists at least one variable $j \in I_C$ which is infinitely selected for branching. Thus, we must have $\theta_j \to 0$. Given our branching rule, this implies that all the other continuous variables attain either their lower or upper bounds. Thus $\boldsymbol{x}^* \in \mathrm{vert}\left([\boldsymbol{l}^*, \boldsymbol{u}^*]\right)$, which implies

$(\boldsymbol{t}^*, \boldsymbol{y}^*) \in \text{conv}\left(\mathcal{Y}'(\boldsymbol{x}^*)\right)$ and $F(\boldsymbol{x}^*, \boldsymbol{t}^*, \boldsymbol{y}^*, \boldsymbol{V}^*, \boldsymbol{\xi}^*) = v(\text{LB}^*) \geq v(\text{2SRO-P})$. Since our node selection strategy always picks a node with minimum lower bound, for each node p of the branching sequence we have $v(\text{LB}^p) \leq v(\text{2SRO-P}) \leq v(\text{LB}^*)$. As $v(\text{LB}^p)$ converges to $v(\text{LB}^*)$, we also have $v(\text{LB}^*) = v(\text{2SRO-P})$. □

The previous result applies when considering infinite precision. By introducing a finite tolerance in the algorithm, we can show that an infinite branching cannot occur at any point. More specifically, given an optimal solution $\boldsymbol{x}^{p*}$ of the lower bounding problem at a node p, we introduce the following function

$$G(\boldsymbol{x}^{p*}) := v(\text{UB}^p),$$

that returns the solution value in the upper bounding problem.

Proposition 4 (Convergence in finite precision) *If the branch-and-bound algorithm enters an infinite sequence of nodes, then it must be converging to a point $\boldsymbol{x}^*$ in which function G has a discontinuity.*

Proof The proof of the previous proposition shows that there exists a subsequence $P' \subseteq P$ that converges to a solution $(\boldsymbol{x}^*, \boldsymbol{t}^*, \boldsymbol{y}^*, \boldsymbol{V}^*, \boldsymbol{\xi}^*)$. As $(\boldsymbol{t}^*, \boldsymbol{y}^*) \in \text{conv}\left(\mathcal{Y}'(\boldsymbol{x}^*)\right)$, then $v(LB^*) = G(\boldsymbol{x}^*)$.

Assume now that G is continuous at $\boldsymbol{x}^*$. Then, we have $\{G(\boldsymbol{x}^{p*})\}_{p \in P'} \to G(\boldsymbol{x}^*) = v(\text{LB}^*)$, which allows us to fathom the node by optimality after a finite number of nodes for any positive tolerance. □

We conclude this section by observing that, at each node of the branch-and-bound algorithm, the lower bounding problem can be solved with ε-tolerance in a finite number of operations. Indeed, as shown in Ceria and Soares (1999) and Grossmann and Ruiz (2012), one can reformulate a convex disjunctive program as a compact convex MINLP by introducing an exponential number of auxiliary variables that model the disjunctions. The resulting model can thus be solved in finite number of states by using any algorithm designed for convex optimization.

3.4 A convexification scheme based on column-generation

In this section, we propose a nonlinear column-generation algorithm to be used, at each node p, to solve problem (LB^p) to ε-optimality in a finite number of iterations. According to this scheme, we approximate $\text{conv}(S^p)$ by the convex hull of a finite set of points belonging to S^p.

Restricted Master Problem: To determine this set, we use an iterative approach. At each iteration k, let $K = \{1, \ldots, k\}$ and denote by $H^{pk} = \{(\bar{\mathbf{x}}^{pj}, \bar{\mathbf{t}}^{pj}, \bar{\mathbf{y}}^{pj}) : j \in K\}$ the associated set of points. We clearly have $\text{conv}\left(H^{pk}\right) \subseteq \text{conv}(S^p)$, thus the optimal solution of the problem obtained by substituting $\text{conv}(S^p)$ with $\text{conv}\left(H^{pk}\right)$ in (LB^p) gives an upper bound of (LB^p). The resulting problem, denoted as ($\widehat{\mathrm{LB}}^{pk}$), is called the *Restricted Master*, and is formulated as follows:

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{t},\boldsymbol{y},\boldsymbol{V},\boldsymbol{\xi},\boldsymbol{\alpha}} \; & \delta^*(\boldsymbol{\xi}|\Xi) - \sum_{i\in Q}(t_i w_i)_*\left(\boldsymbol{v}^i\right) && (22)\\
\text{subject to } & \boldsymbol{x} \in \overline{\mathcal{X}} \cap [\boldsymbol{l}^p, \boldsymbol{u}^p] && (23)\\
& \boldsymbol{x} = \sum_{j\in K} \alpha_j \bar{\mathbf{x}}^{pj} && (24)\\
& \boldsymbol{t} = \sum_{j\in K} \alpha_j \bar{\mathbf{t}}^{pj} && (25)\\
& \boldsymbol{y} = \sum_{j\in K} \alpha_j \bar{\mathbf{y}}^{pj} && (26)\\
& \sum_{j\in K} \alpha_j = 1 && (27)\\
& \sum_{i\in Q} \boldsymbol{v}^i = \boldsymbol{\xi} && (28)\\
& \boldsymbol{v}^i \in \mathbb{R}^{|U|} \qquad \forall i \in Q && (29)\\
& \boldsymbol{\xi} \in \mathbb{R}^{|U|} && (30)\\
& \alpha_j \geq 0 \qquad \forall j \in K && (31)
\end{aligned}
$$

$$(\widehat{\mathrm{LB}}^{pk})$$

Following the classical column-generation framework, the current approximation can be improved by means of a so-called *Pricing Problem*, defined as follows.

Pricing Problem: Let $\boldsymbol{\lambda}^{pk*}, \boldsymbol{\mu}^{pk*}, \boldsymbol{\pi}^{pk*}$ and η^{pk*} be the values of the dual variables associated with constraints (24), (25), (26), and (27) in an optimal solution of problem ($\widehat{\mathrm{LB}}^{pk}$).

Pricing asks to solve the following problem

$$(\bar{\mathbf{x}}^{p,k+1}, \bar{\mathbf{t}}^{p,k+1}, \bar{\mathbf{y}}^{p,k+1}) \in \underset{(\boldsymbol{x},\boldsymbol{t},\boldsymbol{y})\in S^p}{\operatorname{argmin}} \; -\boldsymbol{\lambda}^{pk*^T}\boldsymbol{x} - \boldsymbol{\mu}^{pk^T}\boldsymbol{t} - \boldsymbol{\pi}^{pk*^T}\boldsymbol{y} - \eta^{pk*^T} \qquad (\mathrm{PP}^{pk})$$

and generates a new point $(\bar{\mathbf{x}}^{p,k+1}, \bar{\mathbf{t}}^{p,k+1}, \bar{\mathbf{y}}^{p,k+1})$ belonging to S^p. If $v(\mathrm{PP}^{pk}) \geq -\varepsilon$, we have an ε-optimal solution to (LB^p), and hence the algorithm terminates. Otherwise, we set $H^{k+1} = H^k \cup \{(\bar{\mathbf{x}}^{p,k+1}, \bar{\mathbf{t}}^{p,k+1}, \bar{\mathbf{y}}^{p,k+1})\}$, $k = k+1$ and iterate. Note that, at each iteration k, a lower bound on the optimal solution value of (LB^p) is given by $v(\widehat{\mathrm{LB}}^{pk}) - v(\mathrm{PP}^{pk})$. This lower bound, combined with an upper bound, can allow us to early terminate the solution of problem (LB^p).

The convergence of nonlinear column generation has been established in García et al (2003) and implies finite ε-convergence of our method.

4 Computational experiments

In this section, we report computational results of our solution algorithm when applied to two different optimization problems, a facility location problem and a capital budgeting problem, respectively. Both problems are relevant from an application viewpoint and are defined as non-trivial variants of problems already addressed in the literature.

All the experiments were run on an AMD 3960 running at 3.8 GHz, with a time limit equal to 3,600 CPU seconds per run.

4.1 Facility location problem with adjustable capacity and set-up costs

We consider a company which has to decide, among a set V_1 of possible locations, the sites where to open a facility in order to serve a set V_2 of customers, each with an associated with a known demand d_j. The size of each opened facility has to be determined as well, and an upper limit q_i on the capacity that can be installed on each site $i \in V_1$ is given. The objective of the problem is to minimize a cost function that includes both facility-opening costs and the transportation costs to serve the customers. More into details, transportation costs consist of a fixed component h_{ij} to be paid if customer j is assigned to facility i, and a variable component ξ_{ij} to be paid for each unit of good that traverses this connection. We assume that the variable transportation costs are not known precisely, i.e., ξ_{ij} are uncertain parameters.

To model the problem, we introduce, for each location $i \in V_1$, a binary variable x_i taking the value 1 if a facility is opened in location i and 0 otherwise. In addition, we introduce a continuous variable $z_i \in [0,1]$ indicating the fraction of the maximum capacity that is installed. Using these variables, the first-stage feasible space is given by $\mathcal{X} = \{(\boldsymbol{x}, \boldsymbol{z}) \in \{0,1\}^{|V_1|} \times [0,1]^{|V_1|} : \boldsymbol{z} \leq \boldsymbol{x}\}$. To model the second-stage feasible region, we introduce, for every connection between $i \in V_1$ and $j \in V_2$, a binary variable y_{ij} which takes the value 1 if and only if the connection is used, and a continuous variable $w_{ij} \in [0,1]$ representing the fraction of demand d_j served by facility i. We then have:

$$\mathcal{Y}(\boldsymbol{z}) = \left\{ (\boldsymbol{y}, \boldsymbol{w}) : \begin{array}{ll} \boldsymbol{y} \in \{0,1\}^{|V_1| \times |V_2|} & \\ \boldsymbol{w} \in [0,1]^{|V_1| \times |V_2|} & \\ \sum_{i \in V_1} w_{ij} = 1 & \forall j \in V_2 \\ \sum_{j \in V_2} d_j w_{ij} \leq z_i q_i & \forall i \in V_1 \\ w_{ij} \leq y_{ij} & \forall i \in V_1, \forall j \in V_2 \end{array} \right\} \tag{31}$$

Here, the first set of constraint enforces that every customer is served, while the second set of constraint imposes that the installed capacities are not exceeded. Finally, the last set of constraints link the w and the y variables.

For each site $i \in V$, the associated opening cost depends on the activation of the facility and on the installed capacity, and is described by the following convex quadratic function

$$F_i(x_i, z_i) = f_i x_i + \gamma_i z_i^2 + \beta_i z_i$$

Our two-stage robust problem reads:

$$\min_{(\boldsymbol{x},\boldsymbol{z})\in\mathcal{X}} \max_{\boldsymbol{\xi}\in\Xi} \min_{(\boldsymbol{y},\boldsymbol{w})\in\mathcal{Y}(\boldsymbol{z})} \sum_{i\in V_1} \left[f_i x_i + \gamma_i z_i^2 + \beta_i z_i + \sum_{j\in V_2} (h_{ij} y_{ij} + d_j \xi_{ij} w_{ij}) \right] \tag{32}$$

We refer to problem (32) as the *two-stage robust facility location problem with adjustable capacity and set-up transportation costs*. This problem can be cast as (2SRO-P), where the linking constraints between the first- and second-stage variables involve purely continuous variables and exhibits convexity in the first-stage feasible space. Finally, the uncertainty has a linear impact on the second-stage objective function.

We assume that variable transportation costs are described according to a classical ellipsoidal uncertainty set

$$\Xi_\kappa = \left\{ (\xi_{ij})_{i\in V_1, j\in V_2} : (\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})\boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}) \leq \kappa^2 \right\}, \tag{33}$$

where κ is a fixed sensitivity parameter, $\bar{\boldsymbol{\xi}}$ the expected value of ξ_{ij} and $\boldsymbol{\Sigma} = (\sigma_{ij})_{i,j}$ the covariance matrix. For simplicity, we will assume that $\boldsymbol{\Sigma}$ is diagonal (implying that variable costs are independent from each other), although this assumption is not strictly required by our method.

By applying Proposition 1, the following deterministic reformulation results:

$$\min_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{y},\boldsymbol{w}} \sum_{i\in V_1} \left[f_i x_i + \gamma_i z_i^2 + \beta_i z_i + \sum_{j\in V_2} (h_{ij} y_{ij} + d_j \bar{\xi}_{ij} w_{ij}) \right] + \kappa \sqrt{\sum_{i\in V_1} \sum_{j\in V_2} d_j^2 \sigma_{ij}^2 w_{ij}^2} \tag{34}$$

$$\text{subject to } (\boldsymbol{x}, \boldsymbol{z}) \in \mathcal{X} \tag{35}$$

$$(\boldsymbol{y}, \boldsymbol{w}) \in \text{conv}\,(\mathcal{Y}(\boldsymbol{x})) \tag{36}$$

The reader is referred to Appendix B for the derivation of the robust counterpart of ellipsoidal uncertainty sets.

4.1.1 Instance generation

To test the proposed method, a large benchmark of random instances has been generated in the spirit of Cornuejols et al (1991). First, we randomly generate the location of each site $i \in V_1$ and customer $j \in V_2$ using a uniform distribution in the unit square. For each site i and customer j, the fixed component of the transportation cost is obtained by multiplying the euclidean distance between i and j times a random number with uniform distribution in [20, 50]. The expected value of the variable transportation cost is instead obtained by multiplying the euclidean distance times 10. Each covariance parameter is uniformly randomly generated in the interval $[\sigma^-, \sigma^+]$, where σ^- and σ^+ are specified below. For every site $i \in V_1$, the capacity upper limit is generated according to a uniform distribution in $[10, 160]$, and the fixed setup cost is a random number in $[0, 90]$ multiplied by a varying adjustment factor μ. Coefficients γ_i and β_i are computed so that $F_i(1, q_i)$, which represents the cost for activating the full capacity on the site, equals $\sqrt{q_i} \times \alpha$, where α is randomly generated in $[100, 110]$, as in Cornuejols et al (1991). Finally, customers' demands are obtained by defining a random vector $\boldsymbol{d}$ in [0, 1], and scaling its entries so that $\sum_{j \in V_2} q_j / \sum_{i \in V_1} d_i = \nu$, where ν is another parameter.

In all our instances the number of sites and customers are $(4, 8), (6, 12), (8, 16), (9, 18), (10, 20), (11, 22)$ and $(12, 24)$, and the ellipsoidal uncertainty parameter κ belongs to $\{1.0, 1.5, 2.0\}$. Covariance parameters σ^- and σ^+ take values $(0.1, 1), (0.5, 2)$ and $(1, 4)$. Similar to Cornuejols et al (1991), the adjustment factor μ is set to 2.0 for instances with $|V_1||V_2| \leq 500$ and to 1.0 otherwise, and ν takes values in $\{1.5, 2.0, 3.0, 5.0, 10.0\}$. Finally, for each combination of the parameters we generate 4 instances, overall producing 1260 instances.

4.1.2 Numerical results

We now report the computational results of our branch-and-bound algorithm without and with the use, at the root node, of the single-stage heuristic (SSH) of Section 3.3.3; from now on, these versions are denoted as `BB` and `BB_H`, respectively. Table 1 gives the outcome of our experiments for both variants of the algorithm. Each line refers to the 60 instances characterized by the same values of $|V_1|$, $|V_2|$ and κ. Entries of the table give, for each variant, the following information:

- nodes is the average number of branch-and-bound nodes explored (with respect to instances solved to optimality only);
- time is the average computing time (with respect to instances solved to optimality only);
- # opt is the number of optimal solutions.

In addition, for variant `BB_H`, we report in column "% gap$_r$" the average percentage gap at the root node. Letting L_r and U_r be the best lower and upper bound at the root node, the gap is computed as % gap$_r$ $= 100 * \frac{U_r - L_r}{L_r}$. This information is omitted for algorithm `BB`, which never provides a feasible solution at the root node.

The table shows that the complexity of these ACFL instances increases with the size of the underlying network. In addition, for each size of the network, increasing the value of κ (i.e., allowing for more uncertainty in the realization of the profits) makes the instances consistently harder. The basic algorithm `BB`

V_1	V_2	κ	BB nodes	BB time	BB # opt	BB_H % gap$_r$	BB_H nodes	BB_H time	BB_H # opt
4	8	1	4.7	1.3	60	19.6	4.4	1.2	60
		1.5	4.9	1.5	60	19.0	4.5	1.5	60
		2	5.0	2.0	60	18.8	4.8	2.0	60
6	12	1	8.2	15.1	60	19.9	7.6	12.1	60
		1.5	8.9	17.9	60	19.5	8.2	16.0	60
		2	9.1	22.5	60	19.4	8.4	17.7	60
8	16	1	13.8	128.1	60	17.8	12.4	77.3	60
		1.5	13.6	195.2	60	17.4	12.8	113.4	60
		2	13.6	199.2	59	17.1	12.9	191.9	60
9	18	1	17.4	390.3	58	16.5	15.3	296.9	59
		1.5	18.2	441.4	57	17.0	15.4	281.4	56
		2	18.2	392.0	53	17.2	15.9	232.2	54
10	20	1	21.2	824.2	53	18.2	21.3	544.7	57
		1.5	24.0	845.8	50	18.0	21.5	669.2	56
		2	24.6	843.6	48	18.4	21.6	526.3	52
11	22	1	31.1	1369.0	49	20.9	28.6	873.2	55
		1.5	32.0	1229.6	41	20.7	26.9	773.2	51
		2	33.4	1265.3	37	21.2	29.1	1113.0	49
12	24	1	31.9	1325.2	33	19.1	25.2	1113.1	42
		1.5	33.2	1195.4	30	20.4	27.4	1138.3	38
		2	33.5	1208.9	27	20.3	28.0	994.9	34

Table 1: Performance of different variants of the algorithm in solving ACFL problem

solves 85% of the instances, with an average computing time below 10 minutes. Adding SSH heuristic produces a considerable improvement of the results: although the average root node gap is around 20%, the algorithm solves 68 additional instances to optimality (more than 90% in total) and has an average computing time which is reduced by more than 20%.

4.2 Robust Capital Budgeting problem

Our second test-case is a variant of the Robust Capital Budgeting (RCB) problem introduced in Hanasusanto et al (2015) and considered also in Arslan and Detienne (2021).

Consider a company which can allocate a given budget B to a set of projects $i \in \mathcal{N} = \{1, \dots, N\}$; the budget can be increased with loans. Each project $i \in \mathcal{N}$ has a fixed cost c_i and an uncertain profit $\tilde{p}_i(\boldsymbol{\xi})$ which depends on M unknown factors $\boldsymbol{\xi}$ that belong to an uncertainty set Ξ. The company must decide which projects should be activated to maximize the expected profit. To this aim, it may activate some projects after observing the risk factors, though late investments are less effective and are discounted by a factor $f \in [0, 1)$ of their value. In addition, the company has the possibility to request loans both in the first and in the second stage. The maximum amount of a loan is denoted by C_1 and C_2, respectively, and the interest rate is denoted by λ and is increased in the second stage by a factor μ.

The problem can be modeled as the following max-min-max problem.

$$\max_{(\boldsymbol{x},x_0)\in\mathcal{X}} \min_{\boldsymbol{\xi}\in\Xi} \max_{(\boldsymbol{y},y_0)\in\mathcal{Y}(\boldsymbol{x},x_0)} \left[\sum_{i\in\mathcal{N}} \tilde{p}_i(\boldsymbol{\xi})(x_i + f y_i) - (1+\lambda)C_1 x_0 - (1+\lambda\mu)C_2 y_0\right] \tag{37}$$

with $\mathcal{X} = \{(\boldsymbol{x}, x_0) \in \{0,1\}^N \times [0,1] : \sum_{i\in\mathcal{N}} c_i x_i \leq B + C_1 x_0\}$ and $\mathcal{Y}(\boldsymbol{x}, x_0) = \{(\boldsymbol{y}, y_0) \in \{0,1\}^N \times [0,1] : \sum_{i\in\mathcal{N}} c_i(x_i + y_i) \leq B + C_1 x_0 + C_2 y_0, y_i + x_i \leq 1 \ \forall i \in \mathcal{N}\}$. Here, $\boldsymbol{x}$ are binary variables that indicate whether a project has been activated in the first stage or not, while $\boldsymbol{y}$ indicate their activation in the second stage. Variable x_0 (resp. y_0) is a continuous variable indicating the fraction of the loan capacity which is activated in the first (resp. second) stage.

The actual profit associated with each project is given by $\tilde{p}_i(\boldsymbol{\xi}) = \bar{p}_i(1 + \Delta_i(\boldsymbol{\xi}))$ where $\bar{p}_i$ is the nominal profit, $\boldsymbol{\xi}$ denotes the uncertainty belonging to set $\Xi = [-1,1]^M$, and $\Delta_i : [-1,1]^M \to [-0.5, 0.5]$ is a quadratic convex function defined as $\Delta_i(\boldsymbol{\xi}) = \boldsymbol{\xi}^T \boldsymbol{Q}^i \boldsymbol{\xi}/2 + {\boldsymbol{g}^i}^T \boldsymbol{\xi}$.

4.2.1 Convex reformulation

In this section, we show how one can apply Proposition 1 to reformulate (37). By convexifying the inner maximization problem and swapping the inner max and min operators, one obtains

$$\begin{aligned} \max\ & -(1+\lambda)C_1 x_0 - (1+\lambda\mu)C_2 y_0 + \sum_{i\in\mathcal{N}} \bar{p}_i(x_i + f y_i) \\ & + \min_{\boldsymbol{\xi}\in\Xi} \sum_{i\in\mathcal{N}} \Delta_i(\boldsymbol{\xi})(x_i + f y_i)\bar{p}_i && (38) \\ \text{subject to}\ & (\boldsymbol{x}, x_0) \in \mathcal{X} && (39) \\ & (\boldsymbol{y}, y_0) \in \operatorname{conv}\left(\mathcal{Y}(\boldsymbol{x}, x_0)\right). && (40) \end{aligned}$$

We now reformulate the minimization subproblem as follows

$$\begin{aligned} \min_{\boldsymbol{\xi}\in\Xi} \sum_{i\in\mathcal{N}} \Delta_i(\boldsymbol{\xi})(x_i + f y_i)\bar{p}_i &= \min_{\boldsymbol{\xi}\in\mathbb{R}^M} \left[\sum_{i\in\mathcal{N}} \Delta_i(\boldsymbol{\xi})(x_i + f y_i)\bar{p}_i + \delta(\boldsymbol{\xi}|\Xi)\right] && (41) \\ &= \max_{\boldsymbol{\xi}\in\mathbb{R}^M} (-\delta)_*(\boldsymbol{\xi}|\Xi) - \left(\sum_{i\in\mathcal{N}} \Delta_i(\boldsymbol{\xi})(x_i + f y_i)\bar{p}_i\right)^* && (42) \end{aligned}$$

$$
\begin{aligned}
= \qquad \max \quad & (-\delta)_*(\boldsymbol{\xi}|\Xi) - \sum_{i\in\mathcal{N}} \left(t_i\Delta_i(\boldsymbol{s}^i)\right)^* && (43)\\
\text{subject to} \quad & t_i = (x_i + fy_i)\bar{p}_i \qquad \forall i \in \mathcal{N}\\
& \sum_{i\in\mathcal{N}} \boldsymbol{s}^i = \boldsymbol{\xi}\\
& \boldsymbol{\xi} \in \mathbb{R}^M\\
& \boldsymbol{s}^i \in \mathbb{R}^M \qquad \forall i \in \mathcal{N}\\
& \boldsymbol{t} \in \mathbb{R}_+^{|\mathcal{N}|}
\end{aligned}
$$

Note that $(-\delta)_*(\boldsymbol{\xi}|\Xi) = -\delta^*(-\boldsymbol{\xi}|\Xi)$. In addition, by symmetry of Ξ, we have $\delta(-\boldsymbol{\xi}|\Xi) = \delta(\boldsymbol{\xi}|\Xi)$, hence $\delta^*(-\boldsymbol{\xi}|\Xi) = \delta^*(\boldsymbol{\xi}|\Xi) = \max_{\boldsymbol{\zeta}\in\Xi} \boldsymbol{\xi}^T\boldsymbol{\zeta} = ||\boldsymbol{\xi}||_1$, where the last equivalence is based on strong linear duality.

By combining these two results, we have that $(-\delta)_*(\boldsymbol{\xi}|\Xi) = -\,||\boldsymbol{\xi}||_1$.

Moreover

$$
\Delta_i^*(\boldsymbol{\xi}) = \frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{g}^i)^T {\boldsymbol{Q}^i}^{-1}(\boldsymbol{\xi} - \boldsymbol{g}^i) \tag{44}
$$

hence expanding each conjugate that appears in the summation in (43), we get

$$
(t_i\Delta_i)^*(\boldsymbol{s}^i) = \begin{cases} \frac{{\boldsymbol{s}^i}{\boldsymbol{Q}^i}^{-1}\boldsymbol{s}^i}{2t_i} - ({\boldsymbol{g}^i}^T{\boldsymbol{Q}^i}^{-1})^T\boldsymbol{s}^i + ({\boldsymbol{g}^i}^T{\boldsymbol{Q}^i}^{-1}\boldsymbol{g}^i/2)t_i & \text{if } t_i > 0\\ 0 & \text{if } t_i = 0 \text{ and } \boldsymbol{s}^i = \boldsymbol{0}\\ +\infty & \text{otherwise} \end{cases} \tag{45}
$$

As these terms are minimized in the objective function, any optimal solution with $t_i = 0$ must have $\boldsymbol{s}^i = 0$ as well. Note that this can be enforced by introducing, for each project $i \in N$, a non-negative variable r_i and the additional constraint

$$
2t_ir_i \geq {\boldsymbol{s}^i}^T{\boldsymbol{Q}^i}^{-1}\boldsymbol{s}^i.
$$

This constraint is convex since it can be modeled by means of the rotated quadratic cone.

Our final model reads

$$
\begin{aligned}
&\max && (46)\\
&-(1+\lambda)C_1x_0 - (1+\lambda)\mu C_2y_0 + \sum_{i\in\mathcal{N}}\Bigg[\bar{p}_i(x_i + fy_i) - |\xi_i| - r_i\\
&+({\boldsymbol{g}^i}^T{\boldsymbol{Q}^i}^{-1})^T\boldsymbol{s}^i - \left(\frac{{\boldsymbol{g}^i}^T{\boldsymbol{Q}^i}^{-1}\boldsymbol{g}^i}{2}\right)t_i\Bigg] && (47)\\
&\text{subject to} && (48)\\
&(\boldsymbol{x}, x_0) \in \mathcal{X} && (49)\\
&(\boldsymbol{y}, y_0) \in \text{conv}\left(\mathcal{Y}(\boldsymbol{x}, x_0)\right) && (50)\\
&t_i = (x_i + fy_i)\bar{p}_i \qquad \forall i \in \mathcal{N} && (51)\\
&2t_ir_i \geq {\boldsymbol{s}^i}^T{\boldsymbol{Q}^i}^{-1}\boldsymbol{s}^i \qquad \forall i \in \mathcal{N} && (52)\\
&\sum_{i\in\mathcal{N}} \boldsymbol{s}^i = \boldsymbol{\xi} && (53)
\end{aligned}
$$

$$\boldsymbol{\xi} \in \mathbb{R}^{|\mathcal{N}|} \tag{54}$$

$$\boldsymbol{s}^i \in \mathbb{R}^{|\mathcal{N}|} \qquad \forall i \in \mathcal{N} \tag{55}$$

$$\boldsymbol{t} \in \mathbb{R}_+^{|\mathcal{N}|} \tag{56}$$

$$\boldsymbol{r} \in \mathbb{R}_+^{|\mathcal{N}|} \tag{57}$$

4.2.2 Instance generation

The instances in our testbed are generated similar to those in Arslan and Detienne (2021): for each project $i \in \mathcal{N}$, c_i is randomly generated following a uniform distribution between 1 and 100, and the nominal profit $\bar{p}_i$ is defined as $c_i/5$. We set the investment budget $B = H \sum_{i \in \mathcal{N}} c_i$ where H is a given parameter. We assume that postponed investments are discounted by a factor $f = 0.8$.

The loans limits are set to $C_1 = C_2 = 0.2B$ while the interest rate parameters are chosen as $\lambda = 0.025$ and $\mu = 1.2$.

Deviations of the profits are generated as follows. For each project i, we first randomly generate M coefficients $\boldsymbol{g}^i$ taken taken from the M-dimensional unit simplex, and then each such coefficient is multiplied by -0.5 or 0.5 with equal probability. Then, we compute the value of the linear function $\boldsymbol{g}^{iT}\boldsymbol{\xi}$ in all vertices of $[-1, 1]^M$, and compute a convex quadratic function that (i) has the same value as the linear function on each vertex, and (ii) attains its minimum (over $[-1, 1]^M$) in one of these points. (More details about this last step are given in the Appendix).

In our instances, parameter H takes values in $\{0.2, 0.4, 0.6, 0.8\}$, the number of projects N is in $\{30, 40, 50, 60\}$ while the number of risk factors M is either 4 or 8. For each of these settings, we define 10 instances, making a total of 320 instances.

4.2.3 Numerical results

Table 2 gives the same information as Table 1. Since for this problem there are cases in which algorithm `BB` provides a feasible solution at the root node, we also give column "% gap$_r$" (computed as % gap$_r = 100 * \frac{U_r - L_r}{U_r}$) and column "# UB$_r$" reporting the number of instances for which this happens. This latter column is omitted for algorithm `BB_H`, as the SSH heuristic is always able to determine a feasible solution at the root node. Each entry in the table refers to the 40 instances characterized by the same value of N and M, with the exception of those in column % gap^r, which account only the instances for which a valid upper bound U^r is available.

The results show that algorithm `BB` is able to solve almost all the instances in at most 3 minutes. Algorithm `BB_H` has slightly better performances, as it allows to solve to optimality one more instance and reduces the average computing time and number of nodes with respect to `BB`. Overall, `BB_H` is able to solve all instances but 2 to proven optimality, within less than one minute, on average. Finally, observe that `BB_H` may present a larger percentage gap at

		BB					BB_H			
N	M	# UB_r	% gap_r	nodes	time	# opt	% gap_r	nodes	time	# opt
30	4	32	1.7	5.5	17.5	40	3.2	3.1	10.7	40
	8	24	3.3	42.5	131.9	39	1.7	38.7	124.3	39
40	4	28	1.2	5.7	31.1	40	5.4	2.3	15.0	40
	8	25	1.8	23.5	146.6	39	0.8	18.2	117.4	39
50	4	31	0.6	2.7	25.4	40	5.2	1.2	13.5	40
	8	24	1.3	11.9	117.3	38	0.5	15.5	144.0	39
60	4	27	0.5	1.8	23.7	40	5.2	1.0	16.2	40
	8	23	1.2	12.6	180.6	40	0.4	2.2	37.5	40

Table 2: Performance of different variants of the algorithm in solving the RCB problem.

the root node as this figure is computed over the entire set of instances (as opposed to a subset of them for the other algorithm).

4.2.4 Problem variants

We also tested algorithm BB_H on some variants of the problem, obtained combining the following features:

no loans/binary loans/continuous loans: meaning that loans are not available, can be used only at their maximum amount, or can be used at any intermediate value;
independent/dependent risk factors: meaning that risk factors may impact the profit of each project i according to a linear or to a non-linear function. We consider two types of functions $(\Delta_i)_{i\in\mathcal{N}}$: (Q) convex quadratic functions $\Delta_i(\boldsymbol{\xi}) = \boldsymbol{\xi}^T\boldsymbol{Q}^i\boldsymbol{\xi}/2 + {\boldsymbol{g}^i}^T\boldsymbol{\xi}$, and (L) linear functions $\Delta_i(\boldsymbol{\xi}) = {\boldsymbol{g}^i}^T\boldsymbol{\xi}$. In the linear case, the $\boldsymbol{g}^i$ coefficients are generated as in the quadratic case.

Table 3 reports the results of our experiments on the six variants described above. The upper part of the table refers to instances with quadratic risk functions, while the lower part addresses the linear case.

The results show that, in the quadratic case, the "no loans" and "binary loans" variants tend to be slightly easier than the "continuous loans" in terms of number of optimal solutions and average computing time.

As to the linear case, it looks much more challenging that its quadratic counterpart. This counter-intuitive behaviour is due to the way the instances are defined, in particular for what concerns functions $\Delta_i(\boldsymbol{\xi})$. Indeed, the impact of risk factors in the profit reduction is always larger in the linear case than in the quadratic case; since we are considering a robust (i.e., worst-case) setting, this makes the linear case farther from nominal values than the quadratic one. Nevertheless, also in this challenging case, algorithm BB_H is able to solve almost 75% of the instances with an average time below 10 minutes.

	N	M	No loans time	No loans # opt	Binary loans time	Binary loans # opt	Continuous loans time	Continuous loans # opt
Q	30	4	5.5	40	6.0	40	10.7	40
		8	118.4	40	135.8	40	124.3	39
	40	4	8.9	40	9.4	40	15.0	40
		8	169.0	40	97.2	39	117.4	39
	50	4	10.1	40	9.8	40	13.5	40
		8	179.5	40	190.2	40	144.0	39
	60	4	13.8	40	13.2	40	16.2	40
		8	25.1	40	25.7	40	37.5	40
L	30	4	241.1	37	110.7	36	217.3	37
		8	580.0	32	618.9	32	480.9	31
	40	4	223.8	37	195.6	38	193.1	36
		8	447.1	28	276.3	28	247.9	28
	50	4	744.8	22	740.1	24	798.6	25
		8	523.2	39	238.3	40	251.9	40
	60	4	832.9	7	957.4	7	1238.7	8
		8	946.1	32	695.0	33	665.5	32

Table 3: Performance of the full algorithm in solving different variants of the RCB problem.

5 Conclusion

In this work, we studied optimization problems where part of the cost parameters are not known at decision time, and the decision flow is modeled as a two-stage process. In particular, we addressed general problems in which all constraints (including those linking the first and the second stages) are defined by convex functions and involve mixed-integer variables. To the best of our knowledge, this is the first attempt to extend the existing literature to tackle this wide class of problems.

To this aim, we derive a relaxation of the problem which can be formulated as a convex optimization problem, and embed it within a branch-and-bound algorithm where branching occurs on integer and continuous variables. By combining enumeration and on-the-fly generation of the variables, we obtain a branch-and-price scheme, for which we prove convergence to ε-optimality.

In addition to the theoretical analysis, we applied our method to two optimization problems affected by objective uncertainty, namely a variant of the Capacitated Facility Location problem and a capital budgeting problem. Our computational experiments showed that the proposed method is able to solve relevant-size instances for both problems.

Funding

This research was supported by “Mixed-Integer Non Linear Optimisation: Algorithms and Application” consortium, which has received funding from the European Union’s EU Framework Programme for Research and Innovation

Horizon 2020 under the Marie Skłodowska-Curie Actions Grant Agreement No 764759.

Appendix A Recalls of convex and concave conjugate

In this appendix we review some basic results on conjugate functions and Fenchel duality. For a detailed treatment we refer to Rockafellar (1970).

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a given function, its convex conjugate is denoted by $f^* : \mathbb{R}^n \to \mathbb{R}$ and is given by

$$f^*(\boldsymbol{\pi}) = \sup_{\boldsymbol{x} \in \mathrm{dom}(f)} \left\{ \boldsymbol{\pi}^T \boldsymbol{x} - f(\boldsymbol{x}) \right\}$$

Similarly, we denote by g_* the concave conjugate of a given function $g : \mathbb{R}^n \to \mathbb{R}$, given by

$$g_*(\boldsymbol{\pi}) = \inf_{\boldsymbol{x} \in \mathrm{dom}(g)} \left\{ \boldsymbol{\pi}^T \boldsymbol{x} - \boldsymbol{g}(\boldsymbol{x}) \right\}$$

Note that, if f is a proper convex function and g a proper concave function, we have that $f^{**} = f$ and $g_{**} = g$. We now state the following Fenchel duality theorem.

Theorem 1 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a proper convex function and $g : \mathbb{R}^n \to \mathbb{R}$ be a proper concave function, then*

$$\inf_{\boldsymbol{x} \in \mathrm{dom}(f) \cap \mathrm{dom}(g)} \{ f(\boldsymbol{x}) - g(\boldsymbol{x}) \} = \sup_{\boldsymbol{\pi} \in \mathrm{dom}(f^*) \cap \mathrm{dom}(g_*)} \left\{ g_*(\boldsymbol{\pi}) - f^*(\boldsymbol{\pi}) \right\}$$

or equivalently,

$$\sup_{\boldsymbol{x} \in \mathrm{dom}(f) \cap \mathrm{dom}(g)} \{ g(\boldsymbol{x}) - f(\boldsymbol{x}) \} = \inf_{\boldsymbol{\pi} \in \mathrm{dom}(g_*) \cap \mathrm{dom}(f^*)} \left\{ f^*(\boldsymbol{\pi}) - g_*(\boldsymbol{\pi}) \right\}$$

Corollary 3 (Maximizing a concave function over a convex set) *Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a non-empty convex set, $g : \mathbb{R}^n \to \mathbb{R}$ be a proper concave function, then*

$$\sup_{\boldsymbol{x} \in \mathcal{X}} g(\boldsymbol{x}) = \inf_{\boldsymbol{\pi}} \left\{ \delta^*(\boldsymbol{\pi} | \mathcal{X}) - g_*(\boldsymbol{\pi}) \right\}$$

where $\delta(\boldsymbol{x}|\mathcal{X}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in \mathcal{X} \\ +\infty & \text{otherwise.} \end{cases}$

Proof The result holds from the fact that $\sup\{g(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{X}\} = \sup\{g(\boldsymbol{x}) - \delta(\boldsymbol{x}|\mathcal{X})\}$ and by application of Fenchel duality. More precisely, $\delta(\boldsymbol{x}|\mathcal{X})$ is convex and, by non-emptiness of $\mathcal{X}$, is proper. □

Notice that Fenchel duality allows the reformulation of an optimization problem which consists in maximizing a concave function over a convex set as an unconstrained convex problem since $\delta^*(\cdot|\mathcal{X})$ and $(-g_*)(\cdot)$ are convex functions and positively weighted sums preserve convexity.

$h(\boldsymbol{x})$	$h^*(\boldsymbol{\pi})$
Separable sums	
$h(\boldsymbol{x}^1, \boldsymbol{x}^2) = f_1(\boldsymbol{x}^1) + f_2(\boldsymbol{x}^2)$	$h^*(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = f_1^*(\boldsymbol{\pi}^1) + f_2^*(\boldsymbol{\pi}^2)$
Scalar multiplications ($\alpha > 0$)	
$h(\boldsymbol{x}) = \alpha f(\boldsymbol{x})$	$h^*(\boldsymbol{\pi}) = \alpha f^*(\boldsymbol{\pi}/\alpha)$
Affine mapping composition ($\det \boldsymbol{A} \neq 0$)	
$h(\boldsymbol{x}) = f(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$	$h^*(\boldsymbol{\pi}) = f^*(\boldsymbol{A}^{-T}\boldsymbol{\pi}) - \boldsymbol{b}^T\boldsymbol{A}^{-T}\boldsymbol{\pi}$
Sum with affine functions	
$h(\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{a}^T\boldsymbol{x} + \boldsymbol{b}$	$h^*(\boldsymbol{\pi}) = f^*(\boldsymbol{\pi} - \boldsymbol{a}) - \boldsymbol{b}$
Sum of functions	
$h(\boldsymbol{x}) = \sum_{i=1}^{m} f_i(\boldsymbol{x})$	$h^*(\boldsymbol{\pi}) = \inf_{\boldsymbol{v}^i, i=1,\dots,m} \left\{ \sum_{i=1}^{m} f_i^*(\boldsymbol{v}^i) \middle\vert \sum_{i=1}^{m} \boldsymbol{v}^i = \boldsymbol{\pi} \right\}$

Table A1: Some convex conjugate calculus rules

Proposition 5 *Let f be a convex function, we have $(-f)_*(\boldsymbol{\pi}) = -f^*(-\boldsymbol{\pi})$.*

Proof

$$(-f)_*(\boldsymbol{\pi}) = \inf_{\boldsymbol{x}} \left\{ \boldsymbol{\pi}^T\boldsymbol{x} - (-f)(\boldsymbol{x}) \right\} = -\sup_{\boldsymbol{x}} \left\{ -\boldsymbol{\pi}^T\boldsymbol{x} - f(\boldsymbol{x}) \right\} = -f^*(-\boldsymbol{\pi})$$

□

Proposition 6 *Let C be a convex set such that $\boldsymbol{x} \in C \Leftrightarrow -\boldsymbol{x} \in C$, we have $\delta^*(\boldsymbol{\pi}|C) = \delta^*(-\boldsymbol{\pi}|C)$*

Proof We have $\delta^*(\boldsymbol{\pi}|C) = \sup_{\boldsymbol{x}\in C} \boldsymbol{\pi}^T\boldsymbol{x}$. Denoting $\boldsymbol{x}^*$ the optimal primal solution to this optimization problem, there exists $\boldsymbol{u} \in C$ such that $\boldsymbol{u} = -\boldsymbol{x}^*$ and thus $\delta^*(\boldsymbol{\pi}|C) = \sup_{\boldsymbol{u}\in C} -\boldsymbol{\pi}^T\boldsymbol{u} = \delta^*(-\boldsymbol{\pi}|C)$. □

Table A1 reports some calculus rules regarding convex conjugates. The extension to concave conjugates is straightforward.

Appendix B Robust counterpart of conic-representable uncertainty sets

We first start by recalling the following strong duality theorem for conic optimization problems.

Theorem 2 (Conic duality) *If the following conic problem has a strictly feasible solution,*

$$\max \left\{ \boldsymbol{c}^T \boldsymbol{\xi} : \boldsymbol{b} - \boldsymbol{B}\boldsymbol{\xi} \in K \right\}$$

then it is equivalent to

$$\min \left\{ \boldsymbol{b}^T \boldsymbol{\lambda} : \boldsymbol{B}^T \boldsymbol{\lambda} = \boldsymbol{c}, \boldsymbol{\lambda} \in K^* \right\}$$

In the following example, we derive the robust counterpart for general ellipsoidal uncertainty sets.

Example 6 (Ellipsoidal uncertainty set) We consider the following ellipsoidal uncertainty set:

$$\Xi = \left\{ \boldsymbol{\xi} : (\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})^T \boldsymbol{P} (\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}) \le \kappa^2 \right\}$$

where $\boldsymbol{P}$ is a definite positive matrix. We consider the following robust counterpart:

$$\text{maximize } \boldsymbol{c}^T \boldsymbol{\xi} \tag{B1}$$

$$\text{subject to } \boldsymbol{\xi} \in \Xi \tag{B2}$$

Let $\boldsymbol{F}$ be a matrix such that $\boldsymbol{P} = \boldsymbol{F}^T \boldsymbol{F}$, we have

$$(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})^T \boldsymbol{P} (\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}) \le \kappa^2 \Leftrightarrow \left|\left| \boldsymbol{F}(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}) \right|\right|_2 \le \kappa \tag{B3}$$

$$\Leftrightarrow (\kappa, \boldsymbol{F}(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}})) \in \mathcal{Q}^{n+1} \tag{B4}$$

$$\Leftrightarrow \begin{bmatrix} \kappa \\ -\boldsymbol{F}\bar{\boldsymbol{\xi}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{F}\boldsymbol{\xi} \end{bmatrix} \in \mathcal{Q}^{n+1} \tag{B5}$$

Applying the strong duality theorem, we obtain an equivalent minimization problem:

$$\text{minimize } \kappa\mu - (\boldsymbol{F}\bar{\boldsymbol{\xi}})^T \boldsymbol{\lambda} \tag{B6}$$

$$\text{subject to } \; - \boldsymbol{F}^T \boldsymbol{\lambda} = \boldsymbol{c} \tag{B7}$$

$$(\mu, \boldsymbol{\lambda}) \in \mathcal{Q}^{n+1} \tag{B8}$$

By inspection, we have $\boldsymbol{\lambda} = -\boldsymbol{F}^{-T} \boldsymbol{c}$ and thus, one obtains

$$\text{minimize } \kappa\mu + \underbrace{\bar{\boldsymbol{\xi}} \boldsymbol{F}^T \boldsymbol{F}^{-T} \boldsymbol{c}}_{= \boldsymbol{c}^T \bar{\boldsymbol{\xi}}} \tag{B9}$$

$$\text{subject to } (\mu, -\boldsymbol{F}^{-T} \boldsymbol{c}) \in \mathcal{Q}^{n+1} \tag{B10}$$

Note that if $\boldsymbol{P}$ is diagonal, i.e., $\boldsymbol{P} = \text{diag}(p_1^2, \dots, p_n^2)$, we have $\boldsymbol{F} = \text{diag}(p_1, \dots, p_n)$ and thus $\boldsymbol{F}^{-T} = \text{diag}(1/p_1, \dots, 1/p_n)$. The resulting problem is therefore

$$\kappa \sqrt{\sum_{i=1}^{n} \frac{c_i^2}{p_i^2}} + \sum_{i=1}^{n} c_i \bar{\xi}_i \tag{B11}$$

Appendix C Generating quadratic functions for RCB-C-Q

In this appendix, we show how one can generate quadratic functions for the two-stage robust capital budgeting problem. We therefore consider an instance

Ω for the RCB-C-Q. First, we generate linear functions $(\Delta_i^L)_{i\in\mathcal{N}}$ in the same way as what was done for linear instances. Then, the goal becomes the one of finding a quadratic functions interpolating functions $(\Delta_i^L)_{i\in\mathcal{N}}$ at the extreme points of $[-1,1]^M$. Thus, for each project $i \in \mathcal{N}$, we want to find a convex quadratic function Δ_i^Q such that

$$\Delta_i^Q(\bar{\boldsymbol{\xi}}) = \Delta_i^L(\bar{\boldsymbol{\xi}}) \qquad \forall \bar{\boldsymbol{\xi}} \in \text{vert}\left([-1,1]^M\right) \tag{C12}$$

Moreover, we want to ensure that the global minimum of function Δ_i^Q is attained at one of the extreme points of $[-1,1]^M$. To this end, let us introduce $\bar{\boldsymbol{\xi}}^{i,min}$ such that

$$\bar{\boldsymbol{\xi}}^{i,min} \in \text{argmin } \left\{\Delta_i^L(\bar{\boldsymbol{\xi}}) : \bar{\boldsymbol{\xi}} \in \text{vert}\left([-1,1]^M\right)\right\} \tag{C13}$$

We then add the following constraints on Δ_i^Q with $\epsilon \geq 0$ is a small number like 10^{-4}:

$$\Delta_i^Q\left(\bar{\boldsymbol{\xi}}^{i,min} + \epsilon \frac{\bar{\boldsymbol{\xi}} - \bar{\boldsymbol{\xi}}^{i,min}}{\left\|\bar{\boldsymbol{\xi}} - \bar{\boldsymbol{\xi}}^{i,min}\right\|}\right) \geq \Delta_i^L(\bar{\boldsymbol{\xi}}^{i,min}) \qquad \forall \bar{\boldsymbol{\xi}} \in \text{vert}\left([-1,1]^M\right) \tag{C14}$$

Such a function may be generated by means of a semidefinite optimization solver. Finally, function Δ_i^Q is shifted so that $\Delta_i^Q(\mathbf{0}) = 0$.

References

Arslan AN, Detienne B (2021) Decomposition-based approaches for a class of two-stage robust binary optimization problems. INFORMS Journal on Computing (to appear).

Ayoub J, Poss M (2016) Decomposition for adjustable robust linear optimization subject to uncertainty polytope. Computational Management Science 13: 219–239.

Ben-Tal A, Goryashko A, Guslitzer E, et al (2004) Adjustable robust solutions of uncertain linear programs. Mathematical Programming 99(2):351–376.

Ben-Tal A, Ghaoui LE, Nemirovski A (2009) Robust Optimization. Princeton University Press.

Bertsimas D, Caramanis C (2010) Finite adaptability in multistage linear optimization. IEEE Transactions on Automatic Control 55(12):2751–2766.

Bertsimas D, Brown D, Caramanis C (2010) Theory and applications of robust optimization. SIAM Review 53: 464–501.

Bertsimas D, Litvinov E, Sun X, et al (2013) Adaptive robust optimization for the security constrained unit commitment problem. IEEE Transactions on Power Systems 28:52–63.

Ceria S, Soares J (1999) Convex programming for disjunctive convex optimization. Mathematical Programming 86:595–614.

Cornuejols G, Sridharan R, Thizy J (1991) A comparison of heuristics and relaxations for the capacitated plant location problem. European Journal of Operational Research 50(3):280–297.

Gabrel V, Lacroix M, Murat C, Remli N (2011) Robust location transportation problems under uncertain demands. Discrete Applied Mathematics 164: 100–111.

García R, Marín A, Patriksson M (2003) Column generation algorithms for nonlinear optimization, I: Convergence analysis. Optimization 52(2):171–200.

Grossmann IE, Ruiz JP (2012) Generalized disjunctive programming: A framework for formulation and alternative algorithms for MINLP optimization. In: Lee J, Leyffer S (Eds) Mixed Integer Nonlinear Programming. Springer, New York, NY, pp 93–115.

Hanasusanto GA, Kuhn D, Wiesemann W (2015) K-adaptability in two-stage robust binary programming. Operations Research 63(4):877–891.

Hassene A, Bazgan C, Vanderpooten D (2009) Min-max and min-max regret versions of combinatorial optimization problems: A survey. European Journal of Operational Research 197:427–438.

Jiang R, Zhang M, Li G, Guan Y (2014) Two-stage network constrained robust unit commitment problem. European Journal of Operational Research 234:751–762.

Kämmerling N, Kurtz J (2020) Oracle-based algorithms for binary two-stage robust optimization. Computational Optimization and Applications 77(2):539–569.

Leyffer S, Menickelly M, Munson T, Vanaret C, Wild SM (2020) A survey of nonlinear robust optimization. INFOR: Information Systems and Operational Research 58(2):342–373.

Li C, Grossmann IE (2019) A finite ε-convergence algorithm for two-stage stochastic convex nonlinear programs with mixed-binary first and second-stage variables. Journal of Global Optimization 75(4):921–947.

Perchet V, Vigeral G (2015) A minmax theorem for concave-convex mappings with no regularity assumptions. Journal of Convex Analysis 22:537–540.

Rockafellar R (1970) Convex Analysis. Princeton University Press.

Sherali HD, Fraticelli B (2002) A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. Journal of Global Optimization 22:319–342.

Sherali HD, Zhu X (2006) On solving discrete two-stage stochastic programs having mixed-integer first- and second-stage variables. Mathematical Programming 108(2):597–616.

Soyster AL (1973) Technical note–convex programming with set-inclusive constraints and applications to inexact linear programming. Operations Research 21(5):1154–1157.

Subramanyam A, Gounaris CE, Wiesemann W (2019) K-adaptability in two-stage mixed-integer robust optimization. Mathematical Programming Computation 12(2):193–224.

Terry T, Epelman M, Thiele A (2009) Robust linear optimization with recourse. Technical Report, Lehigh University

Yanıkoğlu İ, Gorissen BL, den Hertog D (2019) A survey of adjustable robust optimization. European Journal of Operational Research 277(3):799–813.

Zeng B, Zhao L (2013) Solving two-stage robust optimization problems by a column-and-constraint generation method. Operations Research Letters 41:457–461.