EXACT APPROACHES FOR CONVEX ADJUSTABLE ROBUST OPTIMIZATION

HENRI LEFEBVRE, ENRICO MALAGUTI, AND MICHELE MONACI

ABSTRACT. Adjustable Robust Optimization (ARO) is a paradigm for facing uncertainty in a decision problem, in case some recourse actions are allowed after the actual value of all input parameters is revealed. While several approaches have been introduced for the linear case, little is known regarding exact methods for the convex case. In this work, we introduce a new general framework for attacking ARO problems involving convex functions in the recourse problem. We first recall a semi-infinite reformulation of the problem and, provided that one can solve a non-convex separation problem, show how to solve it either by a generalized Benders decomposition or by a column-and-constraint generation approach. For the relevant case in which the uncertainty set has an affine mapping to a 0-1 polytope, we show that the separation problem can be reformulated as a convex Mixed-Integer NonLinear Problem, thus allowing us to derive computationally sound exact methods. Finally, we apply the resulting algorithms to two different applications, namely a nonlinear facility location problem and a nonlinear resource allocation problem, to numerically assess their computational performance.

1. INTRODUCTION

Adjustable Robust Optimization (ARO) is a paradigm used to face uncertainty in case some recourse actions are allowed after the actual value of all input parameters is revealed. An ARO problem can be formulated as

$$\inf_{\boldsymbol{x}\in X} \sup_{\boldsymbol{\xi}\in\Xi} \inf_{\boldsymbol{y}\in Y(\boldsymbol{x},\boldsymbol{\xi})} g_0(\boldsymbol{x},\boldsymbol{y}),$$
(ARO)

in which X denotes the feasible set of decisions to be taken here and now (first-stage decisions), Ξ is the uncertainty set, and $Y(x, \xi)$ is the set of all feasible recourse actions (second-stage decisions) for a given $x \in X$ and $\xi \in \Xi$. In this paper we consider a broad class of ARO problems, characterised by convex objective function g_0 and second stage feasible set. For this class of problems, we devise solution approaches based on separation of first-stage decisions via a cutting planes approach. In the very relevant case in which Ξ can be affinely mapped to a 0-1 polytope (see e.g., Bertsimas and Sim 2004), we show that the separation problem can be formulated as a convex Mixed-Integer NonLinear Problem (MINLP).

Notations. Matrices and vectors are written in bold case, e.g., $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, while scalars are written in normal font, e.g., a_{ij} and b_i . We use \leq to compare vectors of agreeable size, component-wise and let **0** be the zero-vector of appropriate size (which will be clear from the context). We denote by \mathbb{R} the

Date: November 27, 2023.

²⁰²⁰ Mathematics Subject Classification. 90C11, 90C17, 90C30.

Key words and phrases. Adjustable Robust Optimization, Convex Optimization, Mixed-Integer NonLinear Programming, Fenchel Duality.

extended real line $\mathbb{R} \cup \{-\infty, \infty\}$. For a given function $f : \mathbb{R}^n \to \overline{\mathbb{R}}, f^*$ denotes its convex conjugate defined as

$$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x} \in \operatorname{dom}(f)} \left\{ \boldsymbol{y}^\top \boldsymbol{x} - f(\boldsymbol{x}) \right\}$$
(1)

with dom $(f) = \{ \boldsymbol{x} \in \mathbb{R}^n : f(\boldsymbol{x}) < +\infty \}$ its domain. We let f_* denote its concave conjugate. Assuming that f is real-valued, its associated perspective function $h : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ is defined as h(x,t) := tf(x/t) if t > 0, and $h(x,0) = \liminf_{(v',t')\to(v,0)} t'f(v'/t')$; see Rockafellar (1996), p. 67. For ease of exposition, we use tf(x/t) to denote the perspective function h(x,t) in the rest of this paper. For a given convex set S, we let vert (S) denote the set of extreme points of S, and we let relint(S) denote its relative interior. We recall that a 0-1-polytope is a polytope whose extreme points are binary vectors.

1.1. Literature review.

1.1.1. Linear case. ARO problems are known to be intractable even in the simple case in which only linear functions appear in the constraints defining $Y(x, \boldsymbol{\xi})$; see Ben-Tal et al. (2004). For this setting, several exact approaches have been introduced in the literature. A first line of research is based on an adaptation of Benders' decomposition algorithm, in which the second-stage problem is dualized and an epigraph reformulation of the remaining maximization problem is used. Then, "Benders' cuts" are dynamically generated in a dynamic way; see, e.g., Terry et al. (2009), Bertsimas et al. (2013), Jiang et al. (2014) and Gabrel et al. (2014). An alternative approach is the column-and-constraint generation algorithm proposed by Zeng and Zhao (2013). In this scheme, a restricted master problem is iteratively solved and augmented by introducing second-stage variables and constraints associated with harmful scenarios. The identification of such scenarios requires the solution of a bilevel problem. Later, Ayoub and Poss (2016) solves this bilevel problem by means of a MILP obtained by exploiting a description of the uncertainty set in terms of its extreme points. Finally, the Fourier-Motzkin elimination technique is used in Zhen et al. (2018) to remove the second-stage decisions from the definition of (ARO) and solve the problem to optimality.

Given the complexity results for linear ARO problems, many approximation methods have been presented in the literature for this class of problems. For instance, Bertsimas and Caramanis (2010) introduce the finite adaptability approach (also known as K-adaptability), in which a fixed number of second-stage decisions must be taken in the first stage so that each of them is used to address a subset of the uncertainty set, the union of such subsets being the whole original uncertainty set. An MILP formulation was introduced by Hanasusanto et al. (2015) for problems with binary second-stage decisions and objective uncertainty only. For the general (still linear) case, a scenario-based branch-and-bound algorithm was later proposed by Subramanyam et al. (2019). The Affine Decision Rule (ADR) approach was introduced by Ben-Tal et al. (2004) in which second-stage decisions are replaced by affine functions of the uncertainty. Though this restriction may seem arbitrary and restrictive, experimental evidences show that the obtained approximation is typically of good quality. Moreover, in Bertsimas and Goyal (2011), the authors show that an optimal affine policy always exists when the uncertainty set is a simplex and the uncertain parameters only appear in the right-hand-side of the second-stage linear problem. The work in Bertsimas and Bidkhori (2014) provides, for the same class of problems, a priori approximation bounds between the solution of (ARO) and its ADR approximation based on geometrical properties of the uncertainty set. In addition, the authors show that this gap is zero when the uncertainty set is the intersection of an ℓ_2 -ball and the non-negative orthant. In Bertsimas and

Ruiter (2016), the authors derive a dualized formulation of linear ARO problems and show that its ADR approximation is typically faster to solve than its primal ADR counterpart. They also show that an optimal primal ADR can be derived from an optimal dual ADR.

1.1.2. Convex case. Regarding the convex case, occurring when $Y(\boldsymbol{x}, \boldsymbol{\xi})$ is defined by convex functions of the first- and second- stage variables, the techniques introduced for the linear case are not directly applicable; the scientific literature is much more sparse and existing approaches focus on specific settings.

Takeda et al. (2007) considers ARO problems in which the uncertainty set is expressed as the convex hull of a finite set of points, and report conditions under which such problems can be reduced to a single-stage problem. In Boni and Ben-Tal (2008), the authors consider ARO problems with ellipsoidal uncertainty set and conic quadratic second-stage constraints. They show that an optimal ADR can be obtained by means of a semidefinite problem. In Ruiter et al. (2022), the authors extend the approach proposed by Bertsimas and Ruiter (2016) and derive a dualized problem for a class of convex ARO problems. Linearity of the dualized problem with respect to the uncertain parameters allows then to obtain a tight approximation by using ADRs. Moreover, the authors show that a primal *feasible* ADR can be derived from an optimal dual ADR. This is in contrast with the result of Bertsimas and Ruiter (2016) in which an *optimal* primal ADR could be derived from the dual. Moreover, we enlight that their approach considers a smaller class of problems compared to that studied in this work for they require g_i to be separable in \boldsymbol{x} and \boldsymbol{y} for all i = 1, ..., m (i.e., there exists g_i^X and g_i^Y such that $g_i(\boldsymbol{x}, \boldsymbol{y}) = g_i^X(\boldsymbol{x}) + g_i^Y(\boldsymbol{y})$).

1.2. Main assumptions. Let n_X, n_{Ξ}, n_Y and m be given natural numbers, we make the following assumptions.

Assumption 1. The first-stage feasible set $X \subset \mathbb{R}^{n_X}$ is bounded.

Assumption 2. The uncertainty set $\Xi \subset \mathbb{R}^{n_{\Xi}}$ is a bounded polyhedron.

Assumption 3. For each i = 1, ..., m and $j = 1, ..., n_{\Xi}$, we let $f_{ij} : \mathbb{R}^{n_X} \to \mathbb{R}$ be given real-valued convex functions; for a given $\mathbf{x} \in \mathbb{R}^{n_X}$, we denote by $\mathbf{F}(\mathbf{x})$ the $m \times n_{\Xi}$ matrix whose generic element is $f_{ij}(\mathbf{x})$. Similarly, for each i = 1, ..., m, we let $g_i : \mathbb{R}^{n_X+n_Y} \to \mathbb{R}$ be a real-valued convex function; for a given $\mathbf{x} \in \mathbb{R}^{n_X}$ and $\mathbf{y} \in \mathbb{R}^{n_Y}$, we denote by $\mathbf{g}(\mathbf{x}, \mathbf{y})$ the m-dimensional vector whose generic element is $g_i(\mathbf{x}, \mathbf{y})$. For given $\mathbf{x} \in X$ and $\boldsymbol{\xi} \in \Xi$, the second-stage feasible space $Y(\mathbf{x}, \boldsymbol{\xi})$ is defined by

$$Y(\boldsymbol{x},\boldsymbol{\xi}) = \{\boldsymbol{y} \in \mathbb{R}^{n_Y} : \boldsymbol{F}(\boldsymbol{x})\boldsymbol{\xi} + \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) \le \boldsymbol{0}\}.$$
(2)

Moreover, we assume that the objective function $g_0 : \mathbb{R}^{n_X + n_Y} \to \mathbb{R}$ is a real-valued convex function and that relint $(\text{dom}(g_i)) \neq \emptyset$ for all i = 0, ..., m.

Assumptions 1-3 simply state that (ARO) is an ARO problem with polyhedral uncertainty set and convex second-stage feasible space. Additionally, we make the following technical assumption used to prove our main theorem.

Assumption 4. For any $(x_0, \mathbf{x}) \in \mathbb{R} \times X$ and any $\boldsymbol{\xi} \in \Xi$, the set

$$Z(x_0, \boldsymbol{x}, \boldsymbol{\xi}) = \left\{ \begin{pmatrix} \beta_0 \\ \boldsymbol{\beta} \end{pmatrix} \in \mathbb{R}^{m+1} : \exists \boldsymbol{y} \in \mathbb{R}^{n_Y}, \begin{array}{c} g_0(\boldsymbol{x}, \boldsymbol{y}) - x_0 &\leq \beta_0 \\ \boldsymbol{F}(\boldsymbol{x}) \boldsymbol{\xi} + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}) &\leq \boldsymbol{\beta} \end{array} \right\}$$
(3)

is closed.

We now present an example illustrating Assumption 4.

Example 1. To ease our example, let us assume that $n_Y = 1$ and $\mathbf{F} = \mathbf{0}$, i.e., the set $Y(\mathbf{x}, \boldsymbol{\xi})$ does not depend on \mathbf{x} and $\boldsymbol{\xi}$. We consider a case defined by $g_0(\mathbf{x}, y) = g_0(y) = 1/y$ and, for some $l \in \mathbb{R}$ and $u \in \mathbb{R}$,

$$\boldsymbol{g}(\boldsymbol{x}, y) = \boldsymbol{g}(y) = \begin{pmatrix} y - u \\ l - y \end{pmatrix}.$$
(4)

Then, Assumption 4 requires that

$$Z(x_0, \boldsymbol{x}, \boldsymbol{\xi}) = Z(x_0) = \left\{ (\beta_0, \beta_1, \beta_2) \in \mathbb{R}^3 : \exists y \in \mathbb{R}, \begin{array}{l} 1/y - x_0 \leq \beta_0 \\ y - u \leq \beta_1 \\ l - y \leq \beta_2 \end{array} \right\}$$
(5)

is closed. We treat two cases.

Case $l = -\infty$ and $u = +\infty$: Then $Z(x_0)$ reduces to $\{\beta_0 \in \mathbb{R} : \exists y \in \mathbb{R}, 1/y - x_0 \leq \beta\}$. For $x_0 = 0$, we have $Z(0) = (0, \infty)$ which is not closed. Thus, Assumption 4 is not satisfied.

Case $l, u \in \mathbb{R} \setminus \{0\}$: Then, $Z(x_0)$ reduces to

$$\{(\beta_0, \beta_1, \beta_3) \in \mathbb{R}^3 : \exists y \in [l - \beta_2, u + \beta_1] : 1/y - x_0 \le \beta_0\},$$
(6)
which is always closed.

Finally, the next lemma, which directly derives from Geoffrion (1972), gives a sufficient condition for Assumption 4 to hold.

Lemma 1. Assume that, for any $x \in X$ and any $\xi \in \Xi$, the set $Y(x, \xi)$ is compact and that g_0 is continuous. Then, Assumption 4 holds.

1.3. Contributions. As discussed in the literature review, few exact methods for convex ARO have been proposed so far, mostly relying on strong assumptions. In this work, we first recall a quite general reformulation for ARO problems, involving an exponential number of constraints. Then we start filling the literature gap by showing that the separation of those constraints can be performed via the solution of a non-convex program. This result, obtained through the use of convex conjugates and Fenchel duality, can be applied to any convex ARO, including cases in which the second stage is an SOCP, an SDP, or a (conic) LP. We show that in a very relevant case, arising when the uncertainty set can be affinely mapped to a 0-1 polytope, the resulting separation problem admits a MINLP reformulation. This includes budgeted uncertainty sets and uncertainty sets expressed as the convex hull of a finite discrete set. In this case, the full power of modern MINLP algorithms and machinery can be exploited for deriving effective solution approaches.

In particular, we consider a generalized Benders decomposition (GBD) algorithm (Geoffrion 1972) and a Column-and-Constraint-Generation (CCG) method (Zeng and Zhao 2013), for which we show finite convergence under mild hypothesis.

Finally, we give the computational evidence of the applicability of our solution methods to two applications arising from practical fields. The first one is a nonlinear version of the Facility Location Problem, involving both binary and continuous decisions, while the second one is a nonlinear variant of a Resource Allocation Problem from the literature.

2. Theoretical development

2.1. A non-convex separation problem. Problem (ARO) can be reformulated as (see e.g., Takeda et al. (2007))

$$x_0$$
 (7a)

 $\mathbf{5}$

s.t.
$$\boldsymbol{x} \in X, x_0 \in \mathbb{R},$$
 (7b)

$$\forall \boldsymbol{\xi} \in \Xi, \exists \boldsymbol{y} \in Y(\boldsymbol{x}, \boldsymbol{\xi}), x_0 \ge g_0(\boldsymbol{x}, \boldsymbol{y}). \tag{7c}$$

Since explicitly adding all constraints (7c) to the formulation is not viable in practice, we follow a separation approach in which, given pair (x_0, \boldsymbol{x}) , we check whether a violated constraint exists. Solving the *separation problem* asks to answer the following question:

"Given $(x_0, \boldsymbol{x}) \in \mathbb{R} \times X$, can we show that for any $\boldsymbol{\xi} \in \Xi$ there exists a $\hat{\boldsymbol{y}} \in Y(\boldsymbol{x}, \boldsymbol{\xi})$ such that $x_0 \ge g_0(\boldsymbol{x}, \hat{\boldsymbol{y}})$? If not, can we identify $\hat{\boldsymbol{\xi}} \in \Xi$ such that either $Y(\boldsymbol{x}, \hat{\boldsymbol{\xi}}) = \emptyset$ or $\forall \boldsymbol{y} \in Y(\boldsymbol{x}, \hat{\boldsymbol{\xi}}), x_0 < g_0(\boldsymbol{x}, \boldsymbol{y})$?".

In the following Lemma, we give a sufficient and necessary condition for answering an easier question: given $(x_0, \boldsymbol{x}) \in \mathbb{R} \times X$ and $\boldsymbol{\xi} \in \Xi$, is there a feasible second-stage decision $\hat{\boldsymbol{y}} \in Y(\boldsymbol{x}, \boldsymbol{\xi})$ such that $x_0 \geq g_0(\boldsymbol{x}, \hat{\boldsymbol{y}})$?

Lemma 2. Let $(x_0, \boldsymbol{x}) \in \mathbb{R} \times X$ and $\boldsymbol{\xi} \in \Xi$. Then, if Assumptions 3-4 hold, there exists $\hat{\boldsymbol{y}} \in Y(\boldsymbol{x}, \boldsymbol{\xi})$ such that $x_0 \geq g_0(\boldsymbol{x}, \hat{\boldsymbol{y}})$ if, and only if, the following condition holds

$$\forall (\lambda_0, \boldsymbol{\lambda}) \in \mathbb{R}_{\geq 0}^{m_Y + 1}, \quad \inf_{\boldsymbol{y} \in \mathbb{R}^{n_Y}} \left\{ \boldsymbol{\lambda}^\top (\boldsymbol{F}(\boldsymbol{x})\boldsymbol{\xi} + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y})) + \lambda_0 (g_0(\boldsymbol{x}, \boldsymbol{y}) - x_0) \right\} \leq 0.$$
(8)

Proof. First, it is straightforward to verify that $\hat{y} \in Y(x, \xi)$ implies (8). Assume now that condition (8) holds, in which case we have

$$\sup_{(\lambda_0,\boldsymbol{\lambda})\in\mathbb{R}_{>0}^{m_Y+1}}\inf_{\boldsymbol{y}\in\mathbb{R}^{n_Y}}\left\{\boldsymbol{\lambda}^{\top}(\boldsymbol{F}(\boldsymbol{x})\boldsymbol{\xi}+\boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}))+\lambda_0(g_0(\boldsymbol{x},\boldsymbol{y})-x_0)\right\}\leq 0.$$
 (9)

Since (0,0) is a possible choice for (λ_0, λ) in (9), we have that

$$\sup_{(\lambda_0,\boldsymbol{\lambda})\in\mathbb{R}_{>0}^{m_Y+1}}\inf_{\boldsymbol{y}\in\mathbb{R}^{n_Y}}\left\{\boldsymbol{\lambda}^{\top}(\boldsymbol{F}(\boldsymbol{x})\boldsymbol{\xi}+\boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}))+\lambda_0(g_0(\boldsymbol{x},\boldsymbol{y})-x_0)\right\}=0.$$
 (10)

As the left-hand side of (10) is the dual of

 $\inf_{x_0, \boldsymbol{x}}$

$$\inf_{\boldsymbol{y}\in\mathbb{R}^{n_Y}}\left\{0: x_0 \ge g_0(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{y}\in Y(\boldsymbol{x}, \boldsymbol{\xi})\right\}$$
(11)

and has finite value, by Assumption 4 and Theorem 5.1 in Geoffrion (1972), (11) must be feasible, i.e., there indeed exists $\boldsymbol{y} \in Y(\boldsymbol{x}, \boldsymbol{\xi})$ such that $x_0 \geq g_0(\boldsymbol{x}, \boldsymbol{y})$. \Box

Remark 1. The condition (8) from Lemma 2 remains valid when adding the restriction $||(\lambda_0, \boldsymbol{\lambda})|| \leq 1$, where $|| \cdot ||$ is any norm of \mathbb{R}^{m+1} . Indeed, scaling does not impact the sign of the optimization problem in (9).

Thanks to Lemma 2, we now introduce a non-convex optimization problem which solves the separation problem.

Theorem 1. Let $(x_0, x) \in \mathbb{R} \times X$. Then, if Assumptions 3-4 hold, the following propositions are equivalent:

(1) $\forall \boldsymbol{\xi} \in \Xi, \exists \hat{\boldsymbol{y}} \in Y(\boldsymbol{x}, \boldsymbol{\xi}), x_0 \geq g_0(\boldsymbol{x}, \hat{\boldsymbol{y}});$

(2) The following non-convex optimization problem has an optimal objective value which is non-positive

$$\sup_{\boldsymbol{\xi},\lambda_0,\boldsymbol{\lambda},\boldsymbol{u}^0,\dots,\boldsymbol{u}^m} -\sum_{i=0}^m \lambda_i g_i \Big|_{\boldsymbol{x}}^* \left(\frac{\boldsymbol{u}^i}{\lambda_i}\right) + \boldsymbol{\lambda}^\top \boldsymbol{F}(\boldsymbol{x}) \boldsymbol{\xi} - \lambda_0 x_0$$
(12a)

s.t.
$$\sum_{i=0}^{m} \boldsymbol{u}^{i} = \boldsymbol{0},$$
 (12b)

$$(\lambda_0, \boldsymbol{\lambda}) \in \Lambda, \tag{12c}$$

$$\boldsymbol{\xi} \in \boldsymbol{\Xi}, \tag{12d}$$

$$\boldsymbol{u}^i \in \mathbb{R}^{n_Y} \qquad i = 0, 1, \dots, m,$$
 (12e)

with
$$g_i|_{\boldsymbol{x}}(\bullet) = g_i(\boldsymbol{x}, \bullet)$$
, and $\Lambda = \{(\lambda_0, \boldsymbol{\lambda}) \in \mathbb{R}^m_+ \times \mathbb{R}_+ : ||(\lambda_0, \boldsymbol{\lambda})|| \le 1\}.$

Proof. Let $(x_0, \boldsymbol{x}) \in \mathbb{R} \times X$. By Lemma 2, for any $\boldsymbol{\xi} \in \Xi$, there exists $\hat{\boldsymbol{y}} \in Y(\boldsymbol{x}, \boldsymbol{\xi})$ such that $x_0 \geq g_0(\boldsymbol{x}, \hat{\boldsymbol{y}})$ if, and only if, condition (8) is satisfied. Let $(\lambda_0, \boldsymbol{\lambda})$ be any element of Λ . We start by re-arranging the terms of (8) for $(\lambda_0, \boldsymbol{\lambda})$ as follows

$$\inf_{\boldsymbol{y}\in\mathbb{R}^{n_Y}}\left\{\boldsymbol{\lambda}^{\top}\boldsymbol{g}|_{\boldsymbol{x}}(\boldsymbol{y}) + \lambda_0 g_0|_{\boldsymbol{x}}(\boldsymbol{y})\right\} + \boldsymbol{\lambda}^{\top}\boldsymbol{F}(\boldsymbol{x})\boldsymbol{\xi} - \lambda_0 x_0 \le 0,$$
(13)

in which terms which do not depend on \boldsymbol{y} are moved out from the optimization problem. Now, letting $\phi(\boldsymbol{y}) = \boldsymbol{\lambda}^{\top} \boldsymbol{g}|_{\boldsymbol{x}}(\boldsymbol{y}) + \lambda_0 g_0|_{\boldsymbol{x}}(\boldsymbol{y})$, by definition, the inf problem in (13) is $(-\phi)_*(\mathbf{0})$. By exploiting the fact that $(-\phi)_*(\boldsymbol{z}) = -\phi^*(-\boldsymbol{z})$ for any \boldsymbol{z} (see Rockafellar (1996), p. 308), we have that (13) is equivalent to

$$-\phi^*(\mathbf{0}) + \boldsymbol{\lambda}^\top \boldsymbol{F}(\boldsymbol{x})\boldsymbol{\xi} - \lambda_0 x_0 \le 0.$$
(14)

Using standard conjugate rules (see Rockafellar (1996), p. 145), one obtains the following expression of ϕ^*

$$\phi^*(\boldsymbol{z}) = \inf_{\lambda_0, \boldsymbol{\lambda}, \boldsymbol{u}^0, \dots, \boldsymbol{u}^m} \sum_{i=1}^m (\lambda_i g_i|_{\boldsymbol{x}})^* (\boldsymbol{u}^i) + (\lambda_0 g_0|_{\boldsymbol{x}})^* (\boldsymbol{u}^0)$$
(15a)

$$\sum_{i=0}^{m} \boldsymbol{u}^{i} = \boldsymbol{z},\tag{15b}$$

$$\boldsymbol{u}^i \in \mathbb{R}^{n_Y} \qquad i = 0, 1, ..., m.$$
 (15c)

Then, we have $(\lambda_i g_i|_{\boldsymbol{x}})^*(\boldsymbol{u}^i) = \lambda_i g_i|_{\boldsymbol{x}}^*(\boldsymbol{u}^i/\lambda_i)$ (see Rockafellar (1996), p. 140). The proof is achieved by requiring that (13) be enforced for all $\boldsymbol{\xi} \in \Xi$.

The result from Theorem 1 addresses generic convex functions. In the following two examples, we apply this general result to two prominent special cases. In particular, we show that Theorem 1 reduces to the results of Ayoub and Poss (2016) if all involved functions are linear. In the second example, we take interest in problems with convex functions defined using ℓ_p -norms.

Example 2 (Linear case). Assume that, for each i = 0, ..., m, it holds $g_i(\boldsymbol{x}, \boldsymbol{y}) = t^i \boldsymbol{x} + \boldsymbol{w}^i \boldsymbol{y} - b_i$ for given $t^i \in \mathbb{R}^{n_X}$, $\boldsymbol{w}^i \in \mathbb{R}^{n_Y}$ and $b_i \in \mathbb{R}$, and define $r_i(\boldsymbol{x}) = t^i \boldsymbol{x} - b_i$. By observing that

$$g_i \Big|_{\boldsymbol{x}}^* \left(\frac{\boldsymbol{u}^i}{\lambda_i} \right) = \begin{cases} -r_i(\boldsymbol{x}) & \text{if } \frac{\boldsymbol{u}^i}{\lambda_i} = \boldsymbol{w}^i \\ +\infty & \text{otherwise} \end{cases}$$

6

we conclude that the first case must be enforced and Theorem 1 yields the following separation problem:

$$\max_{\boldsymbol{\xi},\lambda_0,\boldsymbol{\lambda}} (\boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{F}(\boldsymbol{x})\boldsymbol{\xi})^\top \boldsymbol{\lambda} + (r^0(\boldsymbol{x}) - x_0)\lambda_0$$
(16a)

s.t.
$$\boldsymbol{W}^{\top}\boldsymbol{\lambda} + \boldsymbol{w}_{(0)}^{\top}\lambda_0 = \boldsymbol{0},$$
 (16b)

$$(\lambda_0, \boldsymbol{\lambda}) \in \Lambda, \tag{16c}$$

$$\boldsymbol{\xi} \in \boldsymbol{\Xi}.$$
 (16d)

where $\mathbf{r}(\mathbf{x})$ denotes the vector with components $r_i(\mathbf{x})$ for i = 1, ..., m.

We enlight that the derivation requires linearity with respect to \boldsymbol{y} only, and hence the result remains valid when replacing $\mathbf{t}^i \boldsymbol{x}$ with $t^i(\boldsymbol{x})$ for a generic function $t^i : \mathbb{R}^{n_X} \to \mathbb{R}$. Finally, our result includes the specific case, addressed in Theorem 1 in Ayoub and Poss (2016), in which \boldsymbol{F} is affine in $\boldsymbol{x}, \, \boldsymbol{t}^i = 0$ for all i = 1, ..., m and $\boldsymbol{w}^0 = \mathbf{0}$.

Example 3 (ℓ_p -norm objective and constraints). Assume that, for each i = 0, 1, ..., m, it holds $g_i(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{K}_X^i \boldsymbol{x} + \boldsymbol{K}_Y^i \boldsymbol{y} + \boldsymbol{\chi}^i||_{p_i} + t^i \boldsymbol{x} + \boldsymbol{w}^i \boldsymbol{y} - b_i$ for given matrices \boldsymbol{K}_X^i , \boldsymbol{K}_Y^i , $\boldsymbol{\chi}^i$, vectors t^i , \boldsymbol{w}^i and scalar b_i . Let us denote, for each $i = 0, \ldots, m, a^i(\boldsymbol{x}) = \boldsymbol{K}_X^i \boldsymbol{x} + \boldsymbol{\chi}^i$ and $r_i(\boldsymbol{x}) = t^i \boldsymbol{x} - b_i$. Finally, let \boldsymbol{W} be the matrix composed by vectors \boldsymbol{w}^i ($i = 1, \ldots, m$), and $\boldsymbol{r}(\boldsymbol{x})$ be the vector with components $r_i(\boldsymbol{x})$ ($i = 1, \ldots, m$). Then, the separation problem from Theorem 1 reads¹

$$\sup_{\boldsymbol{\xi},\lambda_0,\boldsymbol{\lambda},\boldsymbol{z}^0,\dots,\boldsymbol{z}^m} \sum_{i=0}^m \boldsymbol{a}^i(\boldsymbol{x})^\top \boldsymbol{z}^i + (\boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{F}(\boldsymbol{x})\boldsymbol{\xi})^\top \boldsymbol{\lambda} + (r_0(\boldsymbol{x}) - x_0)\lambda_0$$
(17a)

s.t.
$$\sum_{i=0}^{m} \boldsymbol{K}_{Y}^{i^{\top}} \boldsymbol{z}^{i} + \boldsymbol{W}^{\top} \boldsymbol{\lambda} + \boldsymbol{w}^{0^{\top}} \lambda_{0} = \boldsymbol{0}, \qquad (17b)$$

$$\left|\left|\boldsymbol{z}^{i}\right|\right|_{p_{i}^{\prime}} \leq \lambda_{i} \qquad i = 0, 1, \dots, m, \tag{17c}$$

$$\boldsymbol{z}^{i} \in \mathbb{R}^{n_{Y}} \qquad i = 0, 1, ..., m, \tag{17d}$$

$$(\lambda_0, \boldsymbol{\lambda}) \in \Lambda, \tag{17e}$$

$$\boldsymbol{\xi} \in \boldsymbol{\Xi}.$$
 (17f)

2.2. Generalized Benders decomposition. In this section, we introduce a new GBD algorithm able to solve (ARO) by means of successive separation of infeasible (x_0, \boldsymbol{x}) points.

For notational convenience, we denote by s a generic tuple $(\boldsymbol{\xi}, \lambda_0, \boldsymbol{\lambda}, \boldsymbol{u}^0, ..., \boldsymbol{u}^m)$, and by S the set of all such tuples satisfying constraints (12b)-(12e). In addition, we introduce function σ defined for each $x_0 \in \mathbb{R}$, $\boldsymbol{x} \in X$ and $\boldsymbol{s} \in S$ as the objective function (12a), i.e.,

$$\sigma(x_0, \boldsymbol{x}; \boldsymbol{s}) := -\sum_{i=0}^m \lambda_i g_i \big|_{\boldsymbol{x}}^* \left(\frac{\boldsymbol{u}}{\lambda_i} \right) + \boldsymbol{\lambda}^\top \boldsymbol{F}(\boldsymbol{x}) \boldsymbol{\xi} - \lambda_0 x_0.$$

In the following theorem, we use the result from Theorem 1 to introduce an alternative projected formulation of (ARO).

Theorem 2. If Assumption 3-4 hold, Problem (ARO) is equivalently solved by the following infinite-dimensional problem

$$\inf_{x_0, \boldsymbol{x}} x_0 \tag{18a}$$

s.t.
$$\boldsymbol{x} \in X, x_0 \in \mathbb{R}$$
 (18b)

$$\sigma(x_0, \boldsymbol{x}; \boldsymbol{s}) \le 0 \qquad \forall \boldsymbol{s} \in S.$$
(18c)

If, moreover, X is a convex MINLP set, then this problem is a convex MINLP.

¹Full details of the derivation can be found in Appendix A.

Proof. The reformulation holds by Theorem 1. Assume that X is a convex MINLP set. To show that the continuous relaxation of (18) is convex, we have to show that, for any $\mathbf{s} \in S$, function $(x_0, \mathbf{x}) \mapsto \sigma(x_0, \mathbf{x}; \mathbf{s})$ is convex. Note that since $\lambda, \boldsymbol{\xi} \geq \mathbf{0}$ are fixed and, for each $i = 1, \ldots, m$ and $j = 1, \ldots, n_{\Xi}$, function f_{ij} is convex, we have that $\mathbf{x} \mapsto \lambda^{\top} \mathbf{F}(\mathbf{x}) \boldsymbol{\xi} - \lambda_0 x_0$ is a non-negative sum of convex functions. Thus, it is convex. We therefore focus on the remaining part and show that $\mathbf{x} \mapsto g_i|_{\mathbf{x}}^*(\boldsymbol{\pi})$ is a concave function for any fixed $\boldsymbol{\pi} \in \mathbb{R}^{n_Y}$. To this end, let $\boldsymbol{\pi} \in \mathbb{R}^{n_Y}$ be fixed with i = 0, ..., m. By definition, we have

$$g_i|_{\boldsymbol{x}}^*(\boldsymbol{\pi}) = \sup_{\boldsymbol{y} \in \operatorname{dom}(g_i|_{\boldsymbol{x}})} \{ \boldsymbol{\pi}^\top \boldsymbol{y} - g_i|_{\boldsymbol{x}}(\boldsymbol{y}) \} = \sup_{\boldsymbol{y} \in \operatorname{dom}(g_i|_{\boldsymbol{x}})} \{ \boldsymbol{\pi}^\top \boldsymbol{y} - g_i(\boldsymbol{x}, \boldsymbol{y}) \}.$$
(19)

Let us introduce new variables $\boldsymbol{z} \in \mathbb{R}^{n_X}$ such that $\boldsymbol{z} = \boldsymbol{x}$. Then, the following holds by Lagrangian duality (note that we have relint $(\text{dom}(g_i)) \neq \emptyset$; see Assumption 3):

$$g_i|_{\boldsymbol{x}}^*(\boldsymbol{\pi}) = \sup_{(\boldsymbol{z}, \boldsymbol{y}) \in \operatorname{dom}(g_i), \boldsymbol{z} = \boldsymbol{x}} \{ \boldsymbol{\pi}^\top \boldsymbol{y} - g_i(\boldsymbol{z}, \boldsymbol{y}) \}$$
(20)

$$= \inf_{\boldsymbol{\lambda} \in \mathbb{R}^{n_X}} \sup_{(\boldsymbol{z}, \boldsymbol{y}) \in \operatorname{dom}(g_i)} \{ \boldsymbol{\lambda}^\top (\boldsymbol{z} - \boldsymbol{x}) + \boldsymbol{\pi}^\top \boldsymbol{y} - g_i(\boldsymbol{z}, \boldsymbol{y}) \}$$
(21)

$$= \inf_{\boldsymbol{\lambda} \in \mathbb{R}^{n_X}} \sup_{(\boldsymbol{z}, \boldsymbol{y}) \in \operatorname{dom}(g_i)} \{ -\boldsymbol{\lambda}^\top \boldsymbol{x} + \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\pi} \end{pmatrix}^\top \begin{pmatrix} \boldsymbol{z} \\ \boldsymbol{y} \end{pmatrix} - g_i(\boldsymbol{z}, \boldsymbol{y}) \}$$
(22)

$$= \inf_{\boldsymbol{\lambda} \in \mathbb{R}^{n_X}} \{ -\boldsymbol{\lambda}^\top \boldsymbol{x} + g_i^*(\boldsymbol{\lambda}, \boldsymbol{\pi}) \}.$$
(23)

Thus, $g_i|_{\boldsymbol{x}}^*(\boldsymbol{\pi})$ can be expressed as the infimum of infinitely many affine functions of \boldsymbol{x} . As a result, it is concave in \boldsymbol{x} .

Algorithm 1 Generalized Benders decomposition

- 1: **Given** an instance of Problem (ARO), a tolerance $\varepsilon > 0$ and an initial set $S^0 \subseteq S$ such that (MP_t) is bounded with t = 0.
- 2: Let $t \leftarrow 0$ be an iteration counter.
- 3: repeat

5:

4: Solve

$$\inf\{x_0: (x_0, \boldsymbol{x}) \in \mathbb{R} \times X, \quad \sigma(x_0, \boldsymbol{x}; \boldsymbol{s}) \le 0 \ \forall \boldsymbol{s} \in S^t\}$$
(MP_t)

with a feasibility tolerance ε .

if (MP_t) is infeasible then Problem (ARO) is infeasible, stop. end if.

6: Let $(x_0^t, \boldsymbol{x}^t)$ be an optimal point of (MP_t) .

7: Solve the separation problem (12) for (x_0^t, x^t) and let $s^t = (\boldsymbol{\xi}^t, \lambda_0^t, \boldsymbol{\lambda}^t, \boldsymbol{u}^{0^t}, ..., \boldsymbol{u}^{m^t})$ denote an optimal point. 8: Let $S^{t+1} \leftarrow S^t \cup \{\boldsymbol{s}^t\}$ and $t \leftarrow t+1$.

9: **until** $\sigma(x_0^t, \boldsymbol{x}^t; \boldsymbol{s}^t) \leq \varepsilon$

Based on Theorem 2, we can derive a cutting-plane algorithm in which cuts (18c) are dynamically generated. The complete procedure is reported in Algorithm 1, with a tolerance $\varepsilon > 0$ used for checking violation of constraints (18c). We now give sufficient conditions for Algorithm 1 to terminate after a finite number of iterations. In the following theorem, we show that assuming Lipschitz continuity of σ on variables (x_0, \mathbf{x}) is sufficient for finite termination. Later, we show that finite termination is ensured as well in case set X is discrete.

Theorem 3 (Finite termination - Lipschitz). Let Assumptions 1-4 hold and assume that $(x_0, \mathbf{x}) \mapsto \sigma(x_0, \mathbf{x}, \hat{\mathbf{s}})$ is Lipschitz continuous for any $\hat{\mathbf{s}} \in S$. Then, Algorithm 1 finitely terminates.

Proof. Assume that the algorithm does not finitely terminate. It must be that (MP_t) is feasible for all $t \ge 0$, otherwise the algorithm would have stopped at Line 5. Moreover, we must have

$$\sigma(x_0^t, \boldsymbol{x}^t; \boldsymbol{s}^t) > \varepsilon \quad \forall t > 0,$$
(24)

since, otherwise, the algorithm would have stopped at Line 9. Now, consider any iteration $k \in \mathbb{N}$: it must be that

$$\sigma(x_0^t, \boldsymbol{x}^t; \boldsymbol{s}^k) \le 0 \quad \forall t > k, \tag{25}$$

since constraints " $\sigma(x_0, \boldsymbol{x}; \boldsymbol{s}^k) \leq 0$ " is then part of (MP_t) for t > k. Combining (24) and (25), we have

$$\varepsilon < \sigma(x_0^k, \boldsymbol{x}^k; \boldsymbol{s}^k) - \sigma(x_0^t, \boldsymbol{x}^t; \boldsymbol{s}^k) \quad \forall t > k.$$
(26)

By Lipschitz continuity of σ , there exists K > 0 such that

$$\varepsilon < K \| (x_0^k, \boldsymbol{x}^k) - (x_0^t, \boldsymbol{x}^t) \| \quad \forall t > k.$$
(27)

Thus, at each iteration t, a ball B^t of center (x_0^t, x^t) with radius ε/K is prevented from being reached in any future iteration. However, by Assumption 1, X is bounded and, thus, the total volume of all the balls that are cut must be bounded, which contradicts that $t \to \infty$ and that the algorithm does not terminate.

We now show two cases in which the Lipschitz continuity assumption is verified.

Example 4 (Separable functions). Let us assume that, for each i = 0, ..., m, there exists functions g_i^X and g_i^Y such that $g_i(\boldsymbol{x}, \boldsymbol{y}) = g_i^X(\boldsymbol{x}) + g_i^Y(\boldsymbol{y})$ and that g_i^X are Lipschitz continuous functions and that f_{ij} is Lipschitz continuous for any i = 1, ..., m and $j = 1, ..., n_{\Xi}$. Then, $(x_0, \boldsymbol{x}) \mapsto \sigma(x_0, \boldsymbol{x}, \hat{\boldsymbol{s}})$ is Lipschitz continuous for all $\hat{\boldsymbol{s}} \in S$.

Proof. We first show that $-g_i|_{\boldsymbol{x}}^*$ is Lipschitz continuous in \boldsymbol{x} . Let $\boldsymbol{\pi} \in \mathbb{R}^{n_Y}$ be fixed. By definition, it holds

$$-g_i|_{\boldsymbol{x}}^*(\boldsymbol{\pi}) = -\sup_{\boldsymbol{y}\in\operatorname{dom}(-g_i|_{\boldsymbol{x}}^*)} \{\boldsymbol{\pi}^\top \boldsymbol{y} - g_i(\boldsymbol{x}, \boldsymbol{y})\}$$
(28)

$$= -\sup_{\boldsymbol{y}\in\mathrm{dom}(-g_i|_{\boldsymbol{x}}^*)} \{\boldsymbol{\pi}^\top \boldsymbol{y} - g_i^X(\boldsymbol{x}) - g_i^Y(\boldsymbol{y})\}$$
(29)

$$= g_i^X(\boldsymbol{x}) - g_i^{Y^*}(\boldsymbol{\pi}).$$
(30)

Thus, $\mathbf{x} \mapsto -g_i|_{\mathbf{x}}^*(\pi)$ is Lipschitz continuous for any π . The rest follows by nonnegative sums of Lipschitz continuous functions and scaling.

Example 5 (ℓ_p -norms objective and constraints). Consider the setting of Example 3 in which all constraints as well as the objective function are defined using ℓ_p -norms. In this case, σ is given by

$$\sigma(x_0, \boldsymbol{x}, \boldsymbol{s}) = \sum_{i=0}^{m} \boldsymbol{a}^i(\boldsymbol{x})^\top \boldsymbol{z}^i + (\boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{F}(\boldsymbol{x})\boldsymbol{\xi})^\top \boldsymbol{\lambda} + (r^0(\boldsymbol{x}) - x_0)\lambda_0, \quad (31)$$

which is Lipschitz continuous in (x_0, \mathbf{x}) if \mathbf{F} have all its components Lipschitz continuous.

In the next theorem we show that, when X is a finite discrete set, finite termination is obtained without further assumptions on function σ .

Theorem 4 (Finite termination - discrete). Let Assumptions 1-4 hold and assume that X is a discrete set. Then, Algorithm 1 finitely terminates.

Proof. Assume that the algorithm does not terminate. Then (\mathbf{MP}_t) must be feasible for all $t \ge 0$ and (24) must hold; see proof of Theorem 3. Because X is finite, there must exist a point (\hat{x}_0, \hat{x}) that is repeated during the algorithm, i.e., there exist natural numbers i and j with i < j such that $(\hat{x}_0, \hat{x}) = (x_0^i, \mathbf{x}^i) = (x_0^j, \mathbf{x}^j)$. However, we show that this is impossible. By Equation (24), the following holds

$$\sigma(x_0^i, \boldsymbol{x}^i, \boldsymbol{s}^i) = \sigma(\hat{x}_0, \hat{\boldsymbol{x}}, \boldsymbol{s}^i) > \varepsilon.$$
(32)

Yet, by Equation (25) and since i < j, we have

$$\sigma(x_0^j, \boldsymbol{x}^j, \boldsymbol{s}^i) = \sigma(\hat{x}_0, \hat{\boldsymbol{x}}, \boldsymbol{s}^i) \le 0.$$
(33)

We therefore get a contradiction showing that the algorithm must terminate after a finite number of iterations. $\hfill \Box$

The previous results show that Algorithm 1 terminates in a finite number of iterations. We now discuss the approximation of the solution returned by the algorithm with respect to the tolerance ε . For the sake of generality, we consider the case in which the separation problem is solved by means of an oracle having accuracy $\delta \geq 0$, i.e., a procedure which, for a given pair (x_0, \boldsymbol{x}) , returns an $\overline{\boldsymbol{s}} \in S$ such that

$$\sup_{\boldsymbol{s}\in S} \sigma(x_0, \boldsymbol{x}; \boldsymbol{s}) - \sigma(x_0, \boldsymbol{x}, \overline{\boldsymbol{s}}) \le \delta.$$
(34)

In addition, for a given $\alpha \geq 0$, we introduce the set-valued map $Y^{\alpha}(\boldsymbol{x}, \boldsymbol{\xi})$ which, for any $\boldsymbol{x} \in X$ and any $\boldsymbol{\xi} \in \Xi$, is a super-set of $Y(\boldsymbol{x}, \boldsymbol{\xi})$ in which all constraints are relaxed by a term α , i.e.,

$$Y^{\alpha}(\boldsymbol{x},\boldsymbol{\xi}) = \left\{ \boldsymbol{y} \in \mathbb{R}^{n_{Y}} : \boldsymbol{F}(\boldsymbol{x})\boldsymbol{\xi} + \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) \le \alpha \boldsymbol{e} \right\}.$$
(35)

Theorem 5 (Correctness). Let Assumptions 1-4 hold and assume that the separation problem is solved by means of an oracle having accuracy $\delta \ge 0$. If Algorithm (1) terminates, it correctly identifies Problem ARO as infeasible, or returns a solution $(x_0, \mathbf{x}) \in \mathbb{R} \times X$ such that $\forall \boldsymbol{\xi} \in \Xi, \exists \mathbf{y} \in Y^{\varepsilon + \delta}(\mathbf{x}, \boldsymbol{\xi}), x_0 \ge g_0(\mathbf{x}, \mathbf{y}) - \varepsilon - \delta$.

Proof. Assume that the algorithm terminates in Line 9, i.e., at some iteration T, the separation problem returns a solution s^T such that $\sigma(x_0^T, \boldsymbol{x}^T; \boldsymbol{s}^T) \leq \varepsilon$. Since the separation problem is solved with a precision up to δ , it holds

$$\sup_{\boldsymbol{s}\in S} \sigma(x_0^T, \boldsymbol{x}^T; \boldsymbol{s}) - \sigma(x_0^T, \boldsymbol{x}^T, \boldsymbol{s}^T) \le \delta.$$
(36)

Thus, it must be that $\sup_{s \in S} \sigma(x_0^T, \boldsymbol{x}^T; \boldsymbol{s}) \leq \varepsilon + \delta$. In turn, this implies

$$\sup_{\boldsymbol{s}\in S} \sigma(\boldsymbol{x}_0^T, \boldsymbol{x}^T; \boldsymbol{s}) \tag{37}$$

$$= \sup_{\boldsymbol{\xi} \in \Xi, (\lambda_0, \boldsymbol{\lambda}) \in \Lambda} \inf_{\boldsymbol{y} \in \mathbb{R}^{n_Y}} \left\{ \boldsymbol{\lambda}^\top (\boldsymbol{F}(\boldsymbol{x}^T) \boldsymbol{\xi} + \boldsymbol{g}(\boldsymbol{x}^T, \boldsymbol{y})) + \lambda_0 (g_0(\boldsymbol{x}^T, \boldsymbol{y}) - \boldsymbol{x}_0^T) \right\}$$
(38)

$$= \sup_{\boldsymbol{\xi} \in \Xi} \inf_{\boldsymbol{y} \in \mathbb{R}^{n_Y}} \sup_{(\lambda_0, \boldsymbol{\lambda}) \in \Lambda} \left\{ \boldsymbol{\lambda}^\top (\boldsymbol{F}(\boldsymbol{x}^T) \boldsymbol{\xi} + \boldsymbol{g}(\boldsymbol{x}^T, \boldsymbol{y})) + \lambda_0 (g_0(\boldsymbol{x}^T, \boldsymbol{y}) - \boldsymbol{x}_0^T) \right\}$$
(39)

$$= \sup_{\boldsymbol{\xi} \in \Xi} \inf_{\boldsymbol{y} \in \mathbb{R}^{n_{Y}}} \max\left\{ \max_{i=1,\dots,m} \left\{ \boldsymbol{f}_{(i)}(\boldsymbol{x}^{T})\boldsymbol{\xi} + g_{i}(\boldsymbol{x}^{T},\boldsymbol{y}) \right\}; g_{0}(\boldsymbol{x}^{T},\boldsymbol{y}) - x_{0}^{T}; 0 \right\}$$
(40)

$$\leq \varepsilon + \delta. \tag{41}$$

Here, the minimax theorem from Perchet and Vigeral (2015) was used to swap the sup and inf operators. This shows that $(x_0^T, \boldsymbol{x}^T)$ is such that $\forall \boldsymbol{\xi} \in \Xi, \exists \boldsymbol{y} \in Y^{\varepsilon+\delta}(\boldsymbol{x}, \boldsymbol{\xi}), x_0 \geq g_0(\boldsymbol{x}, \boldsymbol{y}) - \varepsilon - \delta$.

Otherwise, assume that the algorithm stops at Line 5, i.e., at some iteration Problem (MP_t) is infeasible. As Problem (MP_t) is a relaxation of (18), this implies infeasibility of (ARO).

2.3. Column-and-constraint generation. As a second solution approach, we propose a CCG algorithm for our convex setting. We refer to Zeng and Zhao (2013) for an introduction to this class of algorithms in the linear case.

Lemma 3. Problem (ARO) is equivalently solved by the following finite-dimensional problem

$$\inf_{x_0, \boldsymbol{x}, \boldsymbol{y}_{\boldsymbol{\xi}}} x_0 \tag{42a}$$

s.t.
$$\boldsymbol{x} \in X$$
, (42b)

$$x_0 \ge g_0(\boldsymbol{x}, \boldsymbol{y}_{\boldsymbol{\xi}}) \qquad \forall \boldsymbol{\xi} \in \operatorname{vert}(\boldsymbol{\Xi}), \qquad (42c)$$

$$\boldsymbol{y}_{\boldsymbol{\xi}} \in Y(\boldsymbol{x}, \boldsymbol{\xi}) \qquad \forall \boldsymbol{\xi} \in \operatorname{vert}(\boldsymbol{\Xi}).$$
 (42d)

Proof. Consider the inf-sup-inf formulation of (ARO) and let, for a fixed $\bar{x} \in X$, $\zeta_{\bar{x}}$ be defined as $\zeta_{\bar{x}}(\boldsymbol{\xi}) = \inf_{\boldsymbol{y} \in Y(\bar{x}, \boldsymbol{\xi})} g_0(\bar{x}, \boldsymbol{y})$. From Fiacco and Kyparisis (1986), it holds that $\zeta_{\bar{x}}$ is a convex function. Thus, we have that

$$\forall \bar{\boldsymbol{x}} \in X, \qquad \sup_{\boldsymbol{\xi} \in \Xi} \zeta_{\bar{\boldsymbol{x}}}(\boldsymbol{\xi}) = \sup_{\boldsymbol{\xi} \in \operatorname{vert}(\Xi)} \zeta_{\bar{\boldsymbol{x}}}(\boldsymbol{\xi}) \tag{43}$$

The rest follows; see e.g. Takeda et al. (2007).

The core idea of CCG (see, Zhen et al. (2018)) is to solve model (42) initially with a (nonempty) subset of scenarios $\hat{\Xi} \subseteq$ vert (Ξ) in Constraints (42c)-(42d). Then, given an optimal solution (x_0^*, x^*) to this relaxed problem, the separation problem is solved to check the feasibility of (x_0^*, x^*) . If it is feasible, then it is also optimal for (ARO). Otherwise, there exists a value for the uncertain parameters, say $\hat{\xi}$, which disproves the feasibility of (x_0^*, x^*) . Thus, constraints of type (42c)-(42d) are added to the relaxation, introducing new variables $y_{\hat{\xi}}$. This step is repeated until no such $\hat{\xi}$ can be identified by solving the separation problem. A complete description of the algorithm is given in Algorithm 2.

Algorithm 2 Column-and-Constraint Generation

- 1: Given an instance of (ARO) and an initial set $\Xi^0 \subseteq \Xi$ such that $(\widetilde{\mathrm{MP}}_t)$ is bounded with t = 0.
- 2: Let $t \leftarrow 0$ be an iteration counter.
- 3: repeat
- 4: Solve

 $\inf\{x_0: (x_0, \boldsymbol{x}) \in \mathbb{R} \times X, \quad \boldsymbol{y}_{\hat{\boldsymbol{\xi}}} \in Y(\boldsymbol{x}, \hat{\boldsymbol{\xi}}) \wedge x_0 \ge g_0(\boldsymbol{x}, \boldsymbol{y}_{\hat{\boldsymbol{\xi}}}) \; \forall \hat{\boldsymbol{\xi}} \in \Xi^t\}. \qquad (\widetilde{\mathrm{MP}}_t)$

- 5: if (MP_t) is infeasible then Problem (ARO) is infeasible, stop. end if.
- 6: Let $(x_0^t, \boldsymbol{x}^t)$ be an optimal point of (MP_t) .

7: Solve the separation problem (12) for (x_0^t, x^t) and let $s^t = (\boldsymbol{\xi}^t, \lambda_0^t, \boldsymbol{\lambda}^t, \boldsymbol{u}^{0^t}, ..., \boldsymbol{u}^{mt})$ denote an optimal point. 8: Let $\Xi^{t+1} \leftarrow \Xi^t \sqcup \{\boldsymbol{\xi}^t\}$ and $t \leftarrow t+1$

8: Let
$$\Xi^{e+2} \leftarrow \Xi^e \cup \{\boldsymbol{\xi}^e\}$$
 and $t \leftarrow t+1$.
9: **until** $\sigma(x_0^t, \boldsymbol{x}^t; \boldsymbol{s}^t) \leq \varepsilon$

We now show, without introducing further assumptions, finite convergence of Algorithm 2.

Theorem 6 (Finite termination). Algorithm 2 terminates after a finite number of iterations.

Proof. To prove the result it is enough to observe that the number of constraints (42c)-(42d) is bounded by the number of vertices of Ξ , and that one different vertex is identified at each iteration.

We conclude this section by comparing the cuts generated by the two algorithms. Let $\hat{\boldsymbol{\xi}}$ be a given scenario and consider the case in which Algorithm 2 imposes feasibility of a given first-stage solution, say $(x_0, \boldsymbol{x}) \in \mathbb{R} \times X$, with respect to that scenario. To this aim, Algorithm 2 would add a pair of constraints (42c)– (42d) which ensure that $\exists \boldsymbol{y} \in Y(\boldsymbol{x}, \hat{\boldsymbol{\xi}})$ so that $x_0 \geq g_0(\boldsymbol{x}, \boldsymbol{y})$. According to Theorem 1, this is true if and only if

$$\sup_{(\hat{\boldsymbol{\xi}},\lambda_0,\boldsymbol{\lambda},\boldsymbol{u}^0,\ldots,\boldsymbol{u}^m)\in S}\sigma(x_0,\boldsymbol{x};(\hat{\boldsymbol{\xi}},\lambda_0,\boldsymbol{\lambda},\boldsymbol{u}^0,\ldots,\boldsymbol{u}^m))\leq 0$$
(44)

Let us consider any point $s = (\boldsymbol{\xi}, \lambda_0, \boldsymbol{\lambda}, \boldsymbol{u}^0, \dots, \boldsymbol{u}^m) \in S$. Thus, (44) implies that

$$(x_0, \boldsymbol{x}; s) \leq 0 \quad \forall s = (\boldsymbol{\xi}, \lambda_0, \boldsymbol{\lambda}, \boldsymbol{u}^0, \dots, \boldsymbol{u}^m) \text{ such that } \boldsymbol{\xi} = \boldsymbol{\xi},$$

i.e., a whole family of constraints that would be generated by Algorithm 1.

2.4. Uncertainty sets with an affine mapping to a 0-1-polytope. Algorithms 1 and 2 give general schemes for solving problem (ARO). However, their practical use depends on the possibility to solve the separation problem. In this section, we present a convex MINLP formulation of the latter in the relevant case in which Ξ has an affine mapping of a 0-1 polytope. Note that, given any polytope, there always exists an affine mapping to a 0-1 polytope. To see this, one can always express Ξ as the set of convex combinations of its extreme points. In the following, we will denote by $\Omega \subseteq \mathbb{R}^{n_{\Omega}}$ the 0-1 polytope associated with Ξ , and assume its dimension n_{Ω} be manageable. Clearly, if we assume that Ξ is given by

$$\Xi = \{ \boldsymbol{\xi} \in [0, 1]^{n_{\Xi}} : \boldsymbol{U}\boldsymbol{\xi} \le \boldsymbol{d} \}$$

$$\tag{45}$$

in which U is a totally unimodular matrix and d is integral, the identity mapping can be used and $n_{\Omega} = n_{\Xi}$. This is notably the case for the budgeted uncertainty set (Bertsimas and Sim 2004) with an integer parameter. For the case with fractional budget parameter, Ayoub and Poss (2016) shows that an affine mapping with a 0-1 polytope of size $2n_{\Xi}$ exists. We now state our theorem.

Theorem 7. Let $\Omega \subseteq \mathbb{R}^{n_{\Omega}}$ be a given 0-1 polytope and let $\rho^{0}, \rho^{1}, ..., \rho^{n_{\Omega}} \in \mathbb{R}^{n_{\Xi}}$ be some vectors. Assume that $\Xi = \tilde{\rho}(\Omega)$ where $\tilde{\rho} : \boldsymbol{\omega} \mapsto \rho^{0} + \sum_{k=1}^{n_{\Omega}} \rho^{k} \omega_{k}$. Then, given a pair $(x_{0}, \boldsymbol{x}) \in \mathbb{R} \times X$, the separation model introduced in Theorem 1 can be reformulated as the following convex MINLP.

$$\sup_{\substack{\boldsymbol{\omega},\lambda_{0},\boldsymbol{\lambda},\boldsymbol{u}^{0},\ldots,\boldsymbol{u}^{n},\\\boldsymbol{\theta}^{1},\ldots,\boldsymbol{\theta}^{n_{\Omega}}}} -\sum_{i=0}^{m} \lambda_{i}g_{i}\big|_{\boldsymbol{x}}^{*}\left(\frac{\boldsymbol{u}}{\lambda_{i}}\right) + \boldsymbol{\lambda}^{\top}\boldsymbol{F}(\boldsymbol{x})\boldsymbol{\rho}^{0} + \sum_{k=1}^{n_{\Omega}} \boldsymbol{\theta}^{k^{\top}}\boldsymbol{F}(\boldsymbol{x})\boldsymbol{\rho}^{k} - \lambda_{0}x_{0} \quad (46a)$$

s.t.
$$\sum_{i=0}^{m} \boldsymbol{u}^{i} = \boldsymbol{0},$$
 (46b)

$$(\lambda_0, \boldsymbol{\lambda}) \in \Lambda, \tag{46c}$$

$$\boldsymbol{\theta}^k \leq \boldsymbol{\lambda} \qquad k = 1, ..., n_{\Omega},$$
(46d)

$$\boldsymbol{\theta}^k \le \omega_k \boldsymbol{e} \qquad k = 1, \dots, n_{\Omega}, \tag{46e}$$

$$\boldsymbol{\theta}^k \ge \boldsymbol{\lambda} + \omega_k \boldsymbol{e} - \boldsymbol{e} \qquad k = 1, ..., n_{\Omega},$$
(46f)

$$\boldsymbol{\theta}^k \in \mathbb{R}^m_{\geq 0} \qquad k = 1, ..., n_\Omega, \tag{46g}$$

$$\boldsymbol{\omega} \in \Omega \cap \{0, 1\}^{n_{\Omega}},\tag{46h}$$

 $\boldsymbol{u}^i \in \mathbb{R}^{n_Y} \qquad i = 0, 1, ..., m. \tag{46i}$

 σ

Proof. By replacing each $\boldsymbol{\xi} \in \Xi$ by an $\boldsymbol{\omega} \in \Omega$ such that $\boldsymbol{\xi} = \boldsymbol{\rho}^0 + \sum_{k=1}^{n_\Omega} \boldsymbol{\rho}^k \omega_k$, objective function (12a) can be rewritten as

$$-\sum_{i=0}^{m} \lambda_{i} g_{i} |_{\boldsymbol{x}}^{*} \left(\frac{\boldsymbol{u}^{i}}{\lambda_{i}} \right) + \boldsymbol{\lambda}^{\top} \boldsymbol{F}(\boldsymbol{x}) \boldsymbol{\rho}^{0} + \sum_{k=1}^{n_{\Omega}} \boldsymbol{\lambda}^{\top} \boldsymbol{F}(\boldsymbol{x}) \boldsymbol{\rho}^{k} w_{k} - \lambda_{0} x_{0}$$
(47)

Noting that this function includes bilinear terms between $\boldsymbol{\xi}$ and $\boldsymbol{\omega}$ and that these variables are not linked in the feasible space, we can restrict our attention to $\boldsymbol{\omega} \in \operatorname{vert}(\Omega) \subseteq \{0,1\}^{n_{\Omega}}$. By introducing variables $\theta_i^k = \lambda_i \omega_k$ $(i = 1, ..., m \text{ and } k = 1, ..., n_{\Omega})$, the bilinear term can be linearized as follows

$$\sum_{k=1}^{n_{\Omega}} \boldsymbol{\lambda}^{\top} \boldsymbol{F}(\boldsymbol{x}) \boldsymbol{\rho}^{k} w_{k} = \sum_{k=1}^{n_{\Omega}} \sum_{i=1}^{m} \sum_{j=1}^{n_{\Xi}} f_{ij}(\boldsymbol{x}) \boldsymbol{\rho}_{j}^{k} \underbrace{\lambda_{i} \omega_{k}}_{=\boldsymbol{\theta}_{i}^{k}} = \sum_{k=1}^{n_{\Omega}} \boldsymbol{\theta}^{k^{\top}} \boldsymbol{F}(\boldsymbol{x}) \boldsymbol{\rho}^{k}.$$
(48)

The result follows as (46d)–(46f) are linearization constraints and $0 \le \lambda_i \le 1$ by assumption.

3. Applications

We tested our methods on variants of two relevant problems arising from logistic and planning applications, namely the Facility Location Problem (FLP) and a Resource Allocation Problem (RAP). In both cases, robustness with respect to uncertain input parameters is taken into account.

Our algorithms are implemented in C++17 using Mosek 10.0 to solve the underlying optimization sub-problems and the open-source library *idol*, see Lefebvre (2023). All experiments were executed on an Intel Xeon Gold 6126 at 2.6 GHz, with a time limit equal to 7,200 CPU seconds per run. Code and instances are freely available at https://github.com/hlefebvr/AD-convex-adjustable-robust-optimization.

3.1. Facility location problem.

3.1.1. Problem description. In the standard capacitated FLP, we are given a set V_1 of potential facilities that can be opened and a set V_2 of customers to be served. Each facility $i \in V_1$ has a capacity q_i and an opening cost f_i , while each customer $j \in V_2$ is associated with a demand d_j . In addition, for each facility $i \in V_1$ and customer $j \in V_2$, a unitary transportation cost $t_{ij} > 0$ is given. The problem asks to decide which facilities to open so as to serve all customers while minimizing the sum of opening and transportation costs. We assume that the demand of a customer can be split among multiple facilities.

In the congested version of the problem, introduced by Desrochers et al. (1995), the amount of good, say v_i , leaving a facility $i \in V_1$ determines congestion at the facility, which is reflected by an additional cost defined as

$$F_{a,b}(v_i) = av_i + bv_i^2,\tag{49}$$

in which a and b are given parameters. The same congestion effect was also considered by Fischetti et al. 2016.

By introducing, for each facility $i \in V_1$, a decision variable x_i taking value one if and only if facility i is activated and, for each connection $(i, j) \in V_1 \times V_2$, a non-negative variable y_{ij} representing the amount of good transported from i to j,

13

the deterministic version of the problem can be formulated as follows.

$$\min \sum_{i \in V_1} \left(f_i x_i + F_{a,b}(v_i) + \sum_{j \in V_2} t_{ij} y_{ij} \right)$$

$$(50)$$

s.t.
$$\sum_{j \in V_2} y_{ij} = v_i \qquad \qquad \forall i \in V_1 \qquad (51)$$

$$\sum_{i \in V_1} y_{ij} = d_j \qquad \qquad \forall j \in V_2 \qquad (52)$$

$$v_i \le q_i x_i \qquad \qquad \forall i \in V_1 \qquad (53)$$

$$y_{ij} \ge 0 \qquad \qquad \forall (i,j) \in V_1 \times V_2 \qquad (54)$$

$$x_i \in \{0, 1\} \qquad \qquad \forall i \in V_1 \tag{55}$$

$$v_i \ge 0 \qquad \qquad \forall i \in V_1 \qquad (56)$$

The objective function (50) minimizes the sum of opening, congestion and transportation costs. Constraints (51) define the amount of good leaving each facility, whereas constraints (52) ensure that every demand is satisfied. Finally, constraints (53) enforce capacity constraints of each facility.

We consider here a robust version of this problem in which demands are uncertain: while opening decisions are taken here and now, transportation decisions are taken in a second stage, when the uncertain demands can be observed. More formally, we assume that, for each customer j, the actual demand is $d_j = \bar{d}_j + \tilde{d}_j\xi_j$, where \bar{d}_j and \tilde{d}_j denote the minimum demand and maximum demand increase, respectively. Vector $\boldsymbol{\xi}$ modelling the overall uncertainty of the problem belongs to the following budgeted uncertainty set (see Bertsimas and Sim (2004))

$$\Xi = \left\{ \boldsymbol{\xi} \in [0,1]^{|V_2|} : \sum_{j \in V_2} \xi_j \le \Gamma \right\}.$$
 (57)

Here, $\Gamma > 0$ is a parameter used to control the conservatism of the obtained solution.

Thus, in this application, the set X is defined as $X = \{0, 1\}^{|V_1|}$ while, for given $\boldsymbol{x} \in X$ and $\boldsymbol{\xi} \in \Xi$, set $Y(\boldsymbol{x}, \boldsymbol{\xi})$ includes all pairs $(\boldsymbol{y}, \boldsymbol{v})$, such that $\boldsymbol{y} \in \mathbb{R}_{\geq 0}^{|V_1| \times |V_2|}$, $\boldsymbol{v} \in \mathbb{R}_{\geq 0}^{|V_1|}$, and fulfilling constraints (52)-(56) with $d_j = \bar{d}_j + \tilde{d}_j \xi_j$. Thus, our model reads

$$\min_{\boldsymbol{x}\in X} \left\{ \sum_{i\in V_1} f_i x_i + \max_{\boldsymbol{\xi}\in\Xi} \min_{(\boldsymbol{y},\boldsymbol{v})\in Y(\boldsymbol{x},\boldsymbol{\xi})} \sum_{i\in V_1} \left(F_{a,b}(v_i) + \sum_{j\in V_2} t_{ij} y_{ij} \right) \right\}.$$
(58)

3.1.2. Instance generation. We consider a benchmark of instances that are randomly generated according to Cornuejols et al. (1991). First, potential facilities and customers are randomly placed in a unit square and transportation cost between a facility i and a customer j is defined as their Euclidean distance multiplied by 10. The capacity of each facility $i \in V_1$ is $q_i = \mathcal{U}(10, 160)$, while its activation cost is $f_i = \mathcal{U}(0, 90) + \mathcal{U}(100, 110)\sqrt{q_i}$, with $\mathcal{U}(a, b)$ a uniformly generated random number between a and b. Minimum demands of customers are randomly generated in the interval [0, 1] and scaled so that $\sum_{i \in V_1} q_i / \sum_{j \in V_2} \bar{d}_j = \mu$, with $\mu > 1$ a parameter. Finally, coefficients a and b are fixed to 0.75, as in Fischetti et al. (2016).

The test set includes instances obtained by varying sizes and parameters. In particular, $(|V_1|, |V_2|)$ takes values (10, 15), (10, 20), (15, 30); μ takes values 1.5, 2.0 and 5.0; As to uncertainty, the ratio \tilde{d}_j/\bar{d}_j between maximum demand increase and minimum demand is set to 0.25 and 0.50. The budget uncertainty Γ is set to $\lfloor p|V_2 \rfloor$ with $p \in \{0.10, 0.20, 0.30\}$, i.e., up to a fraction p of the customers maximally

15

change their demands. For each combination of these parameters, 10 instances are generated, thus producing a total of 540 instances.

3.1.3. Results. Table 1 reports the outcome of our experiments for algorithms implementing generalized Benders decomposition (GBD) and column-and-constraint generation (CCG) on FLP instances. For each algorithm we report the number of instances solved to optimality (out of 30) and the average values of the computing time (t_{TOT}) , which is then split in the time spent for solving the master and the separation problems (t_M and t_S , respectively). In addition, we report the average number of iterations before convergence. All figures refer to instances that are solved to proven optimality only.

					Algo	GBD	Algorithm CCG						
$ V_1 $	$ V_2 $	p	\tilde{d}/\bar{d}	# opt	t_{TOT}	t_M	t_S	# Iter	# opt	t_{TOT}	t_M	t_S	# Iter
10	15	0.10	0.25	30	15.80	3.34	12.36	64.27	30	1.20	0.66	0.50	2.53
10	15	0.10	0.50	30	13.92	2.57	11.26	58.30	30	1.17	0.58	0.51	2.53
10	15	0.20	0.25	30	115.05	2.15	112.80	53.83	30	8.09	0.61	7.43	2.53
10	15	0.20	0.50	30	106.38	1.55	104.75	47.03	30	8.08	0.57	7.45	2.53
10	15	0.30	0.25	30	291.95	2.35	289.52	56.43	30	22.85	0.60	22.20	2.47
10	15	0.30	0.50	30	238.38	1.37	236.95	43.80	30	23.27	0.58	22.64	2.50
10	20	0.10	0.25	30	78.50	2.17	76.22	57.67	30	4.35	0.61	3.68	2.53
10	20	0.10	0.50	30	75.19	1.90	73.21	55.03	30	4.21	0.57	3.58	2.50
10	20	0.20	0.25	30	1031.98	1.86	1030.03	53.40	30	87.36	0.66	86.63	2.67
10	20	0.20	0.50	30	1003.10	1.60	1001.41	49.97	30	88.24	0.65	87.52	2.67
10	20	0.30	0.25	24	3205.64	1.17	3204.37	41.33	30	803.40	0.67	802.66	2.70
10	20	0.30	0.50	27	3521.86	0.73	3521.06	35.48	30	812.86	0.66	812.13	2.73
15	30	0.10	0.25	20	1816.64	1.24	1815.25	50.55	30	130.35	2.39	127.84	2.80
15	30	0.10	0.50	20	1654.60	0.83	1653.66	45.10	30	126.98	2.14	124.71	2.77
15	30	0.20	0.25	0	-	-	-	-	0	-	-	-	-
15	30	0.20	0.50	0	-	-	-	-	0	-	-	-	-
15	30	0.30	0.25	0	-	-	-	-	0	-	-	-	-
15	30	0.30	0.50	0	-	-	-	-	0	-	-	-	-

TABLE 1. Results on FLP instances

The results show that instances with $(|V_1|, |V_2|)$ equal to (10, 15) can be consistently solved to optimality by both methods. Among instances with $(|V_1|, |V_2|)$ equal to (10, 20), GBD fails in 9 cases for p = 0.3, while CCG still solves all instances in less than 15 minutes, on average. Finally, instances with $(|V_1|, |V_2|)$ equal to (15, 30) turned out to be very challenging. GBD solves only 40 instances, all with p = 0.10, while it fails for all instances with larger values of p. Algorithm CCG solves all the 60 instances with p = 0.10 and fails as well on the remaining ones. Overall, the results indicate that CCG outperforms GBD in terms of number of instances solved (420 vs 391). Concerning the number of iterations, for those rows in which both methods solve all the instances, the figure for CCG is always one order of magnitude smaller than the one for GBD. This confirms that the theoretical observation that a single iteration of the former method yields a set of variables and constraints that is equivalent to a whole family of cuts generated by a number of iterations by GBD (see Section 2.3). As the computational effort for generating a single cut is comparable in the two methods, CCG turns out to be one order of magnitude faster than GBD in solving those instances and results in the best solution approach for this application.

3.2. Resource allocation problem.

3.2.1. Problem description. We now consider a resource allocation problem introduced by Luedtke (2010). In this problem, we are given a set I of resources that can be acquired to serve a set J of customers. Each resource $i \in I$ is associated to a unitary cost c_i , and each customer $j \in J$ has a demand d_j . We denote by μ_{ij} the service rate of resource i for customer j, i.e., how many units of the customer's demand can be served by the resource. The problem is to determine the amount of each resource to be acquired, and how to allocate resources to customers, so as to satisfy all demands at minimum cost. The deterministic version of this problem is as follows,

$$\min \sum_{i \in I} c_i x_i \tag{59}$$

s.t.
$$\sum_{i \in J} y_{ij} \le x_i$$
 $\forall i \in I$ (60)

$$\sum_{i \in I} \mu_{ij} y_{ij} \ge d_j \qquad \qquad \forall j \in J \qquad (61)$$

$$x_i \ge 0 \qquad \qquad \forall i \in I \tag{62}$$

$$y_{ij} \ge 0 \qquad \qquad \forall (i,j) \in I \times J \tag{63}$$

in which each variable x_i $(i \in I)$ represents the acquired amount of resource i while y_{ij} $((i, j) \in I \times J)$ denotes the amount of resource i allocated to customer j. Constraints (60) impose that allocated resources do not exceed the acquired amount, whereas Constraints (61) enforce that all demands are met.

A more realistic variant of this problem is obtained by considering that congestion affects the efficiency of resources. This may happen for example in the case of server allocation, where an increased allocation of customer demands to one server may induce delays; thus, in order to ensure that the same amount of the resource can be allocated to each customer, a larger amount of resource has to be acquired with respect to the uncongested setting.

We model congestion by means of equation (49), with $v_i = \sum_{j \in J} y_{ij}$ the total amount of resource allocated to the customers. We assume that a = 1, i.e., we implicitly scale all coefficients accordingly. Thus, the *i*-th constraint (60) can be replaced by

$$\sum_{j \in J} y_{ij} + b_i \left(\sum_{j \in J} y_{ij}\right)^2 \le x_i.$$
(64)

We observe that when $b_i = 0$ the problem reduces to its uncongested version. A very similar expression for congestion in resource allocation has been considered in Lodi et al. (2022) in the context of chance-constraint optimization.

In an uncertain setting, the demands of the customers are not known in advance. In particular, we model uncertainty by introducing a vector $\boldsymbol{\xi}$ of random parameters such that each customer $j \in J$ has demand $d_j = \bar{d}_j + \xi_j \tilde{d}_j$. As in the previous application, we assume vector $\boldsymbol{\xi}$ to belong to a budgeted uncertainty set defined by (57).

While decisions related to the amount of resources to be acquired have to be taken here and now, the assignment of these resources to customers can be defined later in time, after the actual demands materialize. Accordingly, this application can be modelled as an ARO problem, with $X = \mathbb{R}_{\geq 0}^{|I|}$ and, for each $(\boldsymbol{x}, \boldsymbol{\xi}) \in X \times \Xi$, $Y(\boldsymbol{x}, \boldsymbol{\xi})$ is defined as the set of all vectors $\boldsymbol{y} \in \mathbb{R}_{\geq 0}^{|I| \times |J|}$ fulfilling constraints (61) and (64) with $d_j = \bar{d}_j + \tilde{d}_j \xi_j$.

3.2.2. *Instance generation*. We evaluate the performances of our algorithms on a large benchmark of random instances. The service rate values are uniformly

generated in the interval [0, 1]. For each resource $i \in I$, the unitary cost c_i is set to $\mathcal{U}(8, 10) \sum_{j \in J} \mu_{ij}/|J|$ and the congestion coefficient b_i is randomly generated between 0 and 1. Finally, for each customer $j \in J$, the demand d_j is uniformly drawn between 1 and 50.

We generate 10 instances for each pair (|I|, |J|) equal to (10, 20), (10, 30), (15, 30), (15, 40), (20, 40) and (20, 50). As in the previous application, uncertainty on demands is modelled by defining the ratio \tilde{d}_j/\bar{d}_j equal to 0.25 and 0.50, and by considering different values of the budget uncertainty Γ , which is set to $\lfloor p | J \rfloor \rfloor$ with $p \in \{0.10, 0.20, 0.30\}$. Overall, we thus have a benchmark composed by 360 instances.

3.2.3. *Results.* Table 2 has the same structure as Table 1, and every row refers to a subset of 10 instances with the same features.

				Algorithm GBD					Algorithm CCG				
I	J	p	\tilde{d}/\bar{d}	# opt	t_{TOT}	t_M	t_S	# Iter	# opt	t_{TOT}	t_M	t_S	# Iter
10	20	0.10	0.25	10	208.39	0.62	207.54	194.10	10	3.21	0.12	3.03	2.80
10	20	0.10	0.50	10	218.04	0.63	217.17	205.30	10	3.31	0.10	3.14	3.00
10	20	0.20	0.25	10	1541.07	0.59	1540.22	193.10	10	22.72	0.16	22.49	2.90
10	20	0.20	0.50	10	1580.57	0.66	1579.64	200.70	10	21.38	0.10	21.21	2.80
10	20	0.30	0.25	9	4720.62	0.64	4719.68	204.67	10	72.05	0.08	71.90	3.00
10	20	0.30	0.50	7	4168.99	0.68	4167.97	211.71	9	80.03	0.12	79.84	3.22
10	30	0.10	0.25	10	4079.36	0.73	4078.26	218.70	8	45.50	0.06	45.36	2.62
10	30	0.10	0.50	10	4127.34	0.77	4126.20	222.00	10	54.33	0.10	54.15	3.00
10	30	0.20	0.25	-	-	-	-	-	6	1764.69	0.15	1764.45	2.83
10	30	0.20	0.50	-	-	-	-	-	8	1921.74	0.17	1921.48	3.38
10	30	0.30	0.25	-	-	-	-	-	4	3683.68	0.05	3683.55	2.50
10	30	0.30	0.50	-	-	-	-	-	5	3931.56	0.11	3931.38	2.40
15	20	0.10	0.25	10	517.66	1.58	515.57	375.00	10	4.00	0.12	3.81	2.70
15	20	0.10	0.50	10	552.79	1.75	550.51	386.70	10	3.99	0.18	3.73	2.60
15	20	0.20	0.25	9	4334.03	1.73	4331.67	385.33	10	36.13	0.20	35.85	2.80
15	20	0.20	0.50	9	4434.38	1.76	4432.00	393.44	10	32.31	0.08	32.15	2.70
15	20	0.30	0.25	1	4197.12	1.61	4194.91	392.00	10	143.92	0.09	143.75	3.00
15	20	0.30	0.50	1	4478.98	1.69	4476.66	389.00	10	133.22	0.18	132.95	2.80
15	30	0.10	0.25	2	5132.34	1.68	5129.85	380.50	9	54.34	0.16	54.07	2.89
15	30	0.10	0.50	2	5465.35	1.96	5462.65	397.00	10	65.39	0.14	65.08	3.10
15	30	0.20	0.25	-	-	-	-	-	10	2243.79	0.17	2243.49	3.50
15	30	0.20	0.50	-	-	-	-	-	10	2109.74	0.14	2109.48	3.40
15	30	0.30	0.25	-	-	-	-	-	6	4361.71	0.13	4361.46	2.83
15	30	0.30	0.50	-	-	-	-	-	6	4045.70	0.15	4045.44	3.00
20	20	0.10	0.25	10	990.15	3.97	985.22	579.10	10	4.70	0.10	4.50	2.80
20	20	0.10	0.50	10	968.40	3.94	963.50	571.90	10	4.56	0.09	4.37	2.70
20	20	0.20	0.25	4	4817.30	3.57	4812.77	536.00	10	38.68	0.12	38.46	2.80
20	20	0.20	0.50	4	4548.84	3.22	4544.66	505.50	10	44.88	0.18	44.59	3.20
20	20	0.30	0.25	-	-	-	-	-	10	120.76	0.14	120.52	2.80
20	20	0.30	0.50	-	-	-	-	-	10	121.90	0.13	121.66	2.90
20	30	0.10	0.25	-	-	-	-	-	10	76.22	0.24	75.85	2.70
20	30	0.10	0.50	-	-	-	-	-	10	76.55	0.16	76.25	2.70
20	30	0.20	0.25	-	-	-	-	-	10	2446.86	0.13	2446.59	2.70
20	30	0.20	0.50	-	-	-	-	-	10	2497.04	0.15	2496.75	2.80
20	30	0.30	0.25	-	-	-	-	-	4	5915.26	0.13	5914.99	3.00
20	30	0.30	0.50	-	-	-	-	-	3	5249.05	0.12	5248.78	3.00

TABLE 2. Results on RAP instances

The results show that, in this application as well, algorithm CCG outperforms GBD. Overall, the former solves 318 instances, whereas the latter solves only 120 instances. We observe that uncertainty plays a crucial role in determining the hardness of the instances, the larger |J| and p, the most challenging the instance.

					Se	olved inst	Unsolved instances				
I	J	p	\tilde{d}/d	Count	Total	Master	Sepatation	# Iter	Count	# Iter	# Errors
10	20	0.10	0.25	10	3.21	0.12	3.03	2.80	-	-	-
10	20	0.10	0.50	10	3.31	0.10	3.14	3.00	-	-	-
10	20	0.20	0.25	10	22.72	0.16	22.49	2.90	-	-	-
10	20	0.20	0.50	10	21.38	0.10	21.21	2.80	-	-	-
10	20	0.30	0.25	10	72.05	0.08	71.90	3.00	-	-	-
10	20	0.30	0.50	9	80.03	0.12	79.84	3.22	1	2.00	1
10	30	0.10	0.25	8	45.50	0.06	45.36	2.62	2	2.50	2
10	30	0.10	0.50	10	54.33	0.10	54.15	3.00	-	-	-
10	30	0.20	0.25	6	1764.69	0.15	1764.45	2.83	4	2.25	4
10	30	0.20	0.50	8	1921.74	0.17	1921.48	3.38	2	1.50	2
10	30	0.30	0.25	4	3683.68	0.05	3683.55	2.50	6	1.67	2
10	30	0.30	0.50	5	3931.56	0.11	3931.38	2.40	5	1.60	1
15	20	0.10	0.25	10	4.00	0.12	3.81	2.70	-	-	-
15	20	0.10	0.50	10	3.99	0.18	3.73	2.60	-	-	-
15	20	0.20	0.25	10	36.13	0.20	35.85	2.80	-	-	-
15	20	0.20	0.50	10	32.31	0.08	32.15	2.70	-	-	-
15	20	0.30	0.25	10	143.92	0.09	143.75	3.00	-	-	-
15	20	0.30	0.50	10	133.22	0.18	132.95	2.80	-	-	-
15	30	0.10	0.25	9	54.34	0.16	54.07	2.89	1	1.00	1
15	30	0.10	0.50	10	65.39	0.14	65.08	3.10	-	-	-
15	30	0.20	0.25	10	2243.79	0.17	2243.49	3.50	-	-	-
15	30	0.20	0.50	10	2109.74	0.14	2109.48	3.40	-	-	-
15	30	0.30	0.25	6	4361.71	0.13	4361.46	2.83	4	1.75	-
15	30	0.30	0.50	6	4045.70	0.15	4045.44	3.00	4	1.75	-
20	20	0.10	0.25	10	4.70	0.10	4.50	2.80	-	-	-
20	20	0.10	0.50	10	4.56	0.09	4.37	2.70	-	-	-
20	20	0.20	0.25	10	38.68	0.12	38.46	2.80	-	-	-
20	20	0.20	0.50	10	44.88	0.18	44.59	3.20	-	-	-
20	20	0.30	0.25	10	120.76	0.14	120.52	2.80	-	-	-
20	20	0.30	0.50	10	121.90	0.13	121.66	2.90	-	-	-
20	30	0.10	0.25	10	76.22	0.24	75.85	2.70	-	-	-
20	30	0.10	0.50	10	76.55	0.16	76.25	2.70	-	-	-
20	30	0.20	0.25	10	2446.86	0.13	2446.59	2.70	-	-	-
20	30	0.20	0.50	10	2497.04	0.15	2496.75	2.80	-	-	-
20	30	0.30	0.25	4	5915.26	0.13	5914.99	3.00	6	1.33	-
20	30	0.30	0.50	3	5249.05	0.12	5248.78	3.00	7	1.71	-

TABLE 3. Results on RAP with CCG

Indeed, among the 120 instances with |J| = 30 and $p \ge 0.20$, GBD always fails whereas CCG proves optimality in 82 cases. As in the previous application, the average number of iterations required by CCG is consistently smaller than that of GBD, typically by two orders of magnitude, the average computing time per iteration being comparable. As may be expected, both algorithms spend most of the computing time in performing separation, requiring up to a few thousands seconds for the most challenging instances.

4. Conclusion

In this paper, we studied general Adjustable Robust Optimization problems in which the second-stage feasible set is defined by means of convex constraints. These problems can be recast into a formulation with infinitely many constraints, to be handled via a separation approach. By means of Fenchel duality, we are able to express the separation problem as a non-convex problem, allowing the derivation of solution schemes based on either generalized Benders decomposition or columnand-constraint generation. Finally, we show that, for the relevant case in which the

uncertainty set can be mapped into a 0-1 polytope, the separation problem can be expressed as a convex MINLP formulation, allowing us to embed state-of-the-art MINLPs algorithms into an effective solution approach. Computational experiments on two different applications compare the alternative solution schemes, and provide insights on their relative performances when solving this class of problems.

References

- Ayoub, J. and M. Poss (2016). "Decomposition for adjustable robust linear optimization subject to uncertainty polytope." In: *Computational Management Science* 13.2, pp. 219–239.
- Ben-Tal, A., A. Goryashko, E. Guslitzer, and A. Nemirovski (2004). "Adjustable robust solutions of uncertain linear programs." In: *Mathematical Programming* 99.2, pp. 351–376.
- Ben-Tal, A., D. den Hertog, and J.-P. Vial (2014). "Deriving robust counterparts of nonlinear uncertain inequalities." In: *Mathematical Programming* 149.1-2, pp. 265– 299.
- Bertsimas, D. and H. Bidkhori (2014). "On the performance of affine policies for two-stage adaptive optimization: a geometric perspective." In: *Mathematical Programming* 153.2, pp. 577–594.
- Bertsimas, D. and C. Caramanis (2010). "Finite Adaptability in Multistage Linear Optimization." In: *IEEE Transactions on Automatic Control* 55.12, pp. 2751– 2766.
- Bertsimas, D. and V. Goyal (2011). "On the power and limitations of affine policies in two-stage adaptive optimization." In: *Mathematical Programming* 134.2, pp. 491– 531.
- Bertsimas, D., E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng (2013). "Adaptive Robust Optimization for the Security Constrained Unit Commitment Problem." In: *IEEE Transactions on Power Systems* 28.1, pp. 52–63.
- Bertsimas, D. and F. J. C. T. de Ruiter (2016). "Duality in Two-Stage Adaptive Linear Optimization: Faster Computation and Stronger Bounds." In: *INFORMS Journal on Computing* 28.3, pp. 500–511.
- Bertsimas, D. and M. Sim (2004). "The Price of Robustness." In: Operations Research 52.1, pp. 35–53.
- Boni, O. and A. Ben-Tal (2008). "Adjustable robust counterpart of conic quadratic problems." In: Mathematical Methods of Operations Research 68.2, pp. 211–233.
- Cornuejols, G., R. Sridharan, and J. Thizy (1991). "A comparison of heuristics and relaxations for the capacitated plant location problem." In: *European Journal of Operational Research* 50.3, pp. 280–297.
- Desrochers, M., P. Marcotte, and M. Stan (1995). "The congested facility location problem." In: *Location Science* 3.1, pp. 9–23.
- Fiacco, A. V. and J. Kyparisis (1986). "Convexity and concavity properties of the optimal value function in parametric nonlinear programming." In: *Journal of Optimization Theory and Applications* 48.1, pp. 95–126.
- Fischetti, M., I. Ljubić, and M. Sinnl (2016). "Benders decomposition without separability: A computational study for capacitated facility location problems." In: European Journal of Operational Research 253.3, pp. 557–569.
- Gabrel, V., M. Lacroix, C. Murat, and N. Remli (2014). "Robust location transportation problems under uncertain demands." In: *Discrete Applied Mathematics* 164, pp. 100–111.
- Geoffrion, A. M. (1972). "Generalized Benders decomposition." In: Journal of Optimization Theory and Applications 10.4, pp. 237–260.

- Hanasusanto, G. A., D. Kuhn, and W. Wiesemann (2015). "K-Adaptability in Two-Stage Robust Binary Programming." In: Operations Research 63.4, pp. 877– 891.
- Jiang, R., M. Zhang, G. Li, and Y. Guan (2014). "Two-stage network constrained robust unit commitment problem." In: *European Journal of Operational Research* 234.3, pp. 751–762.
- Lefebvre, H. (2023). idol: Generic decomposition methods for mathematical programming. publicly available online. URL: https://hlefebvr.github.io/idol/ (visited on 09/18/2023).
- Lodi, A., E. Malaguti, M. Monaci, G. Nannicini, and P. Paronuzzi (2022). A solution algorithm for chance-constrained problems with integer second-stage recourse decisions. URL: http://www.optimization-online.org/DB_FILE/2021/07/ 8514.pdf.
- Luedtke, J. (2010). "An Integer Programming and Decomposition Approach to General Chance-Constrained Mathematical Programs." In: Integer Programming and Combinatorial Optimization. Springer Berlin Heidelberg, pp. 271–284.
- Perchet, V. and G. Vigeral (2015). "A Minmax Theorem for Concave-Convex Mappings with no Regularity Assumptions." In: *Journal of Convex Analysis* 22.
- Rockafellar, R. T. (1996). *Convex analysis*. en. Princeton Landmarks in Mathematics and Physics. Princeton, NJ: Princeton University Press.
- Ruiter, F. J. C. T. de, J. Zhen, and D. den Hertog (2022). "Dual Approach for Two-Stage Robust Nonlinear Optimization." In: *Operations Research*.
- Subramanyam, A., C. E. Gounaris, and W. Wiesemann (2019). "K-adaptability in two-stage mixed-integer robust optimization." In: *Mathematical Programming Computation* 12.2, pp. 193–224.
- Takeda, A., S. Taguchi, and R. H. Tütüncü (2007). "Adjustable Robust Optimization Models for a Nonlinear Two-Period System." In: *Journal of Optimization Theory* and Applications 136.2, pp. 275–295.
- Terry, T., M. Epelman, and A. Thiele (2009). "Robust linear optimization with recourse." In: *technical report, Lehigh University*.
- Zeng, B. and L. Zhao (2013). "Solving two-stage robust optimization problems using a column-and-constraint generation method." In: Operations Research Letters 41.5, pp. 457–461.
- Zhen, J., D. den Hertog, and M. Sim (2018). "Adjustable Robust Optimization via Fourier–Motzkin Elimination." In: Operations Research 66.4, pp. 1086–1100.

APPENDIX A. ADDITIONAL PROOFS

In this appendix, we give additional proofs of results used throughout this paper. We refer to Ben-Tal et al. (2014) for useful convex conjugate calculus rules.

A.1. **Proof of Example 3.** As stated in the example, assume that each function g_i (i = 0, ..., m) be generically defined by means of ℓ_p -norms, i.e., we have

$$g_i(\boldsymbol{x}, \boldsymbol{y}) = \left| \left| \boldsymbol{K}_X^i \boldsymbol{x} + \boldsymbol{K}_Y^i \boldsymbol{y} + \boldsymbol{\chi}^i \right| \right|_{p_i} + \boldsymbol{t}^i \boldsymbol{x} + \boldsymbol{w}^i \boldsymbol{y} + b_i.$$
(65)

A.1.1. Computing convex conjugates. Before applying Theorem 1, we first compute the convex conjugate of a generic functions g_i for some fixed $i \in \{0, \ldots, m\}$. To this end, let us rewrite $g_i|_{\boldsymbol{x}}$ as

$$g_i|_{\boldsymbol{x}}(\boldsymbol{y}) = h_1(\boldsymbol{y}) + \boldsymbol{t}^i \boldsymbol{x} + \boldsymbol{w}^i \boldsymbol{y} - b_i,$$
(66)

with $h_1(\boldsymbol{y}) = \left| \left| \boldsymbol{K}_X^i \boldsymbol{x} + \boldsymbol{K}_Y^i \boldsymbol{y} + \boldsymbol{\chi}^i \right| \right|_{p_i}$. By addition to an affine function, it holds

$$g_i|_{\boldsymbol{x}}^*(\boldsymbol{\pi}) = h_1^*(\boldsymbol{\pi} - \boldsymbol{w}^i) - \boldsymbol{t}^i \boldsymbol{x} + b_i.$$
(67)

Now, we may write h_1 as

$$h_1(\boldsymbol{y}) = h_2(\boldsymbol{K}_Y^i \boldsymbol{y}) \tag{68}$$

with $h_2(\boldsymbol{y}) = ||\boldsymbol{y} + \boldsymbol{K}_X^i \boldsymbol{x} + \chi^i||_{p_i}$. By composition with a linear mapping and since dom $(h_2) = \mathbb{R}^{n_Y}$, we have

$$h_1^*(\boldsymbol{\pi}) = \inf_{\boldsymbol{\alpha}} \{ h_2^*(\boldsymbol{\alpha}) : \boldsymbol{K}_Y^{i^T} \boldsymbol{\alpha} = \boldsymbol{\pi} \}.$$
(69)

In turn, together with (67), we have

$$g_i|_{\boldsymbol{x}}^*(\boldsymbol{\pi}) = \inf_{\boldsymbol{\alpha}} \{h_2^*(\boldsymbol{\alpha}) : \boldsymbol{K}_Y^{i^T} \boldsymbol{\alpha} = \boldsymbol{\pi} - \boldsymbol{w}^i\} - \boldsymbol{t}^i \boldsymbol{x} + b_i.$$
(70)

Then, let us rewrite h_2 as

$$h_2(\boldsymbol{y}) = h_3(\boldsymbol{y} + \boldsymbol{K}_X^i \boldsymbol{x} + \boldsymbol{\chi}^i)$$
(71)

with $h_3(\boldsymbol{y}) = ||\boldsymbol{y}||_{p_i}$. Thus, by translation of argument, we have

$$h_2^*(\pi) = h_3^*(\pi) - (K_X^i x + \chi^i)^T \pi.$$
 (72)

Now, h_3 being a norm, its convex conjugate is the indicator of the unit ball for the dual norm, i.e.,

$$h_3^*(\pi) = \delta(\pi | B_{p'_i}(\mathbf{0}, 1))$$
(73)

with $1/p_i + 1/p'_i = 1$. Together with (70) and (72), we have

$$g_i|_{\boldsymbol{x}}^*(\boldsymbol{\pi}) = \inf_{\boldsymbol{\alpha}} \left\{ \delta(\boldsymbol{\alpha}|B_{p'_i}(\boldsymbol{0},1)) - (\boldsymbol{K}_X^i \boldsymbol{x} + \boldsymbol{\chi}^i)^T \boldsymbol{\alpha} : \boldsymbol{K}_Y^i{}^T \boldsymbol{\alpha} = \boldsymbol{\pi} - \boldsymbol{w}^i \right\} - \boldsymbol{t}^i \boldsymbol{x} + b_i.$$
(74)

By optimality in (74), we get

$$g_i|_{\boldsymbol{x}}^*(\boldsymbol{\pi}) = \inf_{\boldsymbol{\alpha}} \left\{ -(\boldsymbol{K}_X^i \boldsymbol{x} + \boldsymbol{\chi}^i)^T \boldsymbol{\alpha} - \boldsymbol{t}^i \boldsymbol{x} + b_i \right\}$$
(75a)

s.t.
$$\mathbf{K}_Y^{i T} \boldsymbol{\alpha} = \boldsymbol{\pi} - \boldsymbol{w}^i$$
 (75b)

$$\|\boldsymbol{\alpha}\|_{p'_i} \le 1 \tag{75c}$$

$$\boldsymbol{\alpha} \in \mathbb{R}^{n_Y}.$$
 (75d)

A.1.2. Applying Theorem 1. We now rewrite (75) and replace π with u^i/λ_i and α with α^i . By scalar multiplication with λ_i , we get

$$\lambda_i g_i |_{\boldsymbol{x}}^* (\boldsymbol{u}^i / \lambda_i) = \inf_{\boldsymbol{\alpha}^i} \lambda_i \left(-(\boldsymbol{K}_X^i \boldsymbol{x} + \boldsymbol{\chi}^i)^T \boldsymbol{\alpha}^i - \boldsymbol{t}^i \boldsymbol{x} + b_i \right)$$
(76a)

s.t.
$$\boldsymbol{K}_Y^i \boldsymbol{\alpha}^i = \boldsymbol{u}^i / \lambda_i - \boldsymbol{w}^i$$
 (76b)

$$\|\boldsymbol{\alpha}\|_{p_i'} \le 1 \tag{76c}$$

$$\boldsymbol{\alpha}^i \in \mathbb{R}^{n_Y}.$$
 (76d)

Then, by introducing $\boldsymbol{z}^i = \lambda_i \boldsymbol{\alpha}^i$, we have

$$\lambda_i g_i |_{\boldsymbol{x}}^* (\boldsymbol{u}^i / \lambda_i) = \inf_{\boldsymbol{z}^i} \left\{ -(\boldsymbol{K}_X^i \boldsymbol{x} + \boldsymbol{\chi}^i)^T \boldsymbol{z}^i + \lambda_i (b_i - \boldsymbol{t}^i \boldsymbol{x}) \right\}$$
(77a)

s.t.
$$\boldsymbol{K}_Y^i^T \boldsymbol{z}^i = \boldsymbol{u}^i - \lambda_i \boldsymbol{w}^i$$
 (77b)

$$\left|\left|\boldsymbol{z}^{i}\right|\right|_{p_{i}^{\prime}} \leq \lambda_{i} \tag{77c}$$

$$\boldsymbol{z}^i \in \mathbb{R}^{n_Y}.$$
 (77d)

By substitution into Theorem 1, we obtain the following model:

$$\sup \sum_{i=0}^{m} \left((\boldsymbol{K}_{X}^{i} \boldsymbol{x} + \boldsymbol{\chi}^{i})^{T} \boldsymbol{z}^{i} + \lambda_{i} (\boldsymbol{t}^{i} \boldsymbol{x} - b_{i}) \right) + \boldsymbol{\lambda}^{T} \boldsymbol{F}(\boldsymbol{x}) \boldsymbol{\xi} - \lambda_{0} \boldsymbol{x}_{0}$$
(78a)

s.t.
$$\sum_{i=0}^{m} \left(\boldsymbol{K}_{Y}^{i}{}^{T}\boldsymbol{z}^{i} + \lambda_{i}\boldsymbol{w}^{i} \right) = \boldsymbol{0}$$
(78b)

$$\left|\left|\boldsymbol{z}^{i}\right|\right|_{p_{i}^{\prime}} \leq \lambda_{i} \qquad i = 0, 1, \dots, m \tag{78c}$$

$$\boldsymbol{z}^{i} \in \mathbb{R}^{n_{Y}} \qquad i = 0, 1, ..., m \tag{78d}$$

$$(\lambda_0, \boldsymbol{\lambda}) \in \Lambda \tag{78e}$$

$$\boldsymbol{\xi} \in \boldsymbol{\Xi}.$$
 (78f)

(H. Lefebvre,) Trier University, Department of Mathematics, Universitätsring 15, 54296 Trier, Germany

(E. Malaguti, M. Monaci) Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione "Guglielmo Marconi", Università di Bologna, Viale del Risorgimento, 2, 40136 Bologna, Italy.

Email address: lefebvre@uni-trier.de Email address: enrico.malaguti@unibo.it Email address: michele.monaci@unibo.it