ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Lemont, IL 60439

Nonlinear Robust Optimization

Sven Leyffer, Matt Menickelly, Todd Munson, Charlie Vanaret, and Stefan M. Wild

Mathematics and Computer Science Division

Preprint ANL/MCS-P9040-0218

January 31, 2018

This material was based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357.

Contents

Nonlinear Robust Optimization

Sven Leyffer[a], Matt Menickelly[a], Todd Munson[a], Charlie Vanaret[a], and Stefan M. Wild[a]

[a]Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Ave., Argonne, IL 60439, USA

ARTICLE HISTORY

Compiled February 7, 2018

ABSTRACT

Robust optimization (RO) has attracted much attention from the optimization community over the past decade. RO is dedicated to solving optimization problems subject to uncertainty: design constraints must be satisfied for all the values of the uncertain parameters within a given uncertainty set. Uncertainty sets may be modeled as deterministic sets (boxes, polyhedra, ellipsoids), in which case the RO problem may be reformulated via worst-case analysis, or as families of distributions. The challenge of RO is to reformulate or approximate robust constraints so that the uncertain optimization problem is transformed into a tractable deterministic optimization problem. Most reformulation methods assume linearity of the robust constraints or uncertainty sets of favorable shape, which represents only a fraction of real-world applications. This survey addresses nonlinear RO and includes problem formulations and applications, solution approaches, and available software with code samples.

Keywords: Nonlinear robust optimization.

AMS-MSC2000: 90C30, 62K25.

1. Introduction and Notation

Over the past decade, robust optimization has attracted much attention. A number of excellent surveys and monographs exist [10,17,22,38,44,60], which deal mainly with linear and conic cases. Related papers on the general class of semi-infinite optimization can also be found, for example, in [47,66–68,70]. This survey focuses on nonlinear robust optimization (NRO), which is becoming more important in real-world applications. The NRO problem is

$$\begin{array}{ll} \underset{x\in\mathcal{X}}{\text{minimize}} & f(x) \\ \text{subject to} & c(x;u) \leq 0, \quad \forall u \in \mathcal{U}(x), \end{array} \tag{1.1}$$

where $x \in \mathbb{R}^n$ are the decision variables, $\mathcal{X} \subseteq \mathbb{R}^n$ is the certain feasible set, $u \in \mathbb{R}^p$ are the uncertain parameters, $\mathcal{U}(x)$ is the uncertainty set, and the constraints $c(x;u)$ model the impact of the

Preprint ANL/MCS-P9040-0218

CONTACT: Sven Leyffer. Email: leyffer@anl.gov

uncertainty on the design. Formally, we define $\mathcal{U}(x) : \mathbb{R}^n \to \mathbb{R}^p$ as a set-valued mapping representing the uncertainty set with $c : \mathbb{R}^{n+p} \to \mathbb{R}^q$ the robustness criteria. Whenever the uncertainty set $\mathcal{U}(x)$ is of infinite cardinality, the problem (1.1) is a semi-infinite nonlinear optimization problem. We assume throughout that all functions are smooth on the appropriate sets. We can relax this assumption for some algorithms, provided we are prepared to deal with subgradients. We also assume that $\mathcal{U}(x)$ is a nonempty, compact set for all $x \in \mathcal{X}$. If $\mathcal{U}(x)$ were empty for all $x \in \mathcal{X}$, then the uncertainty constraint could be removed. The compactness assumption ensures that the uncertainty is bounded (unbounded uncertainty is not typically encountered in real-world applications). If the uncertainty set $u \in \mathcal{U}$ is independent of the decision variables x and the constraints are separable in x and u, $c(x; u) = g(x) + d(u) \leq 0$, then we can replace the uncertain constraint by a deterministic constraint, provided that we can (globally) solve

$$d^* := \max_{u \in \mathcal{U}} d(u),$$

where the $\max$ is computed separately for each component of $d(u)$. In this case, the uncertainty constraint becomes $c(x; u) = g(x) + d^* \leq 0$.

We can accommodate minimax problems in robust optimization, where the objective function depends on uncertain parameters, in the formulation (1.1) by introducing an additional variable for the epigraph of the uncertain objective, thereby moving the uncertain objective to $c(x; u)$. We note that, without loss of generality, we can assume that $\mathcal{U}(x) = \mathcal{U}_1(x) \times \cdots \times \mathcal{U}_q(x)$ and that the uncertain constraints can be written as $c_j(x; u) \leq 0, \ \forall u \in \mathcal{U}_j(x)$; see [17]. Thus, we can treat each robustness constraint individually. For ease of presentation, however, we limit our discussion here to the single constraint case ($q = 1$), although the methods target problems with multiple constraints.

In the remainder of this paper, we provide background for robust optimization and different formulations in Section 2, applications and example problems in Section 3, an overview of solution approaches in Section 4, a description of available software in Section 5, a discussion of nonconvexity and global optimization in Section 6, and conclusions in Section 7.

Notation. Throughout the survey, we use the convention that finite sets are denoted by roman letters and infinite sets by calligraphic letters. The decision variables are denoted by $x \in \mathbb{R}^n$, and the uncertain parameters are denoted by $u \in \mathbb{R}^p$. The (deterministic) set of feasible points is denoted by $\mathcal{X}$, and the set of uncertain parameters is denoted by $\mathcal{U}$. We denote the nominal value of the uncertain parameters by $\hat{u} \in \mathcal{U}$. In many applications, the nominal value, $\hat{u}$, is the value of the parameters that the system would take in the absence on uncertainty.

2. Nonlinear Robust Optimization Formulations and Theory

Here we discuss a number of important special cases of nonlinear robust optimization. We start with a description of the nominal optimization problem and its properties and provide stationarity conditions for the standard robust optimization formulation. We then consider minimax problems that arise when we need to handle implementation errors, before discussing a special form of robust optimization, called distributionally robust optimization.

2.1. The Nominal Problem

In general, we can derive a relaxation of the nonlinear robust optimization problems, (1.1), by enforcing the robust constraints over a subset of uncertain parameters. Of particular interest is a finite subset, $U \subset \mathcal{U}$, which results in a nonlinear optimization relaxation. The (global) solution to any such relaxation yields a lower bound. The nominal problem is a particular relaxation obtained by choosing $U = \{\hat{u}\}$ and is defined as

$$\begin{aligned} \underset{x \in \mathcal{X}}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x; \hat{u}) \le 0, \end{aligned} \tag{2.2}$$

which is a finite-dimensional nonlinear problem. We make the following observations:

(1) It follows that the nominal problem is a relaxation of (1.1), which implies that its (global) solution provides a lower bound on the solution of (1.1).
(2) Clearly, if the feasible set $\{x \mid c(x; \hat{u}) \le 0\}$, is empty, then it follows that (2.2), and hence (1.1), has no solution.
(3) However, if (2.2) has no solution, then it does not follow that the robust problem (1.1) has no solution, as illustrated by the following example showing that robustness can immunize a problem against unbounded solutions.

    Consider the two-dimensional problem

$$\underset{x \in \mathbb{R}^2}{\text{minimize}} \; x_1 + x_2 \quad \text{subject to } u_1 x_1 + u_2 x_2 - u_1^2 - u_2^2 \le 0, \; \forall u \in [-1, 1]^2,$$

which corresponds to minimizing $x_1 + x_2$ inside the unit ball. Setting the nominal value of the uncertain parameters as $\hat{u} = (\frac{\sqrt{3}}{2}, \frac{1}{2})$, it follows that the nominal problem is unbounded (Figure 1a), while the robust problem has an optimal solution at $x = (-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$ (Figure 1b).

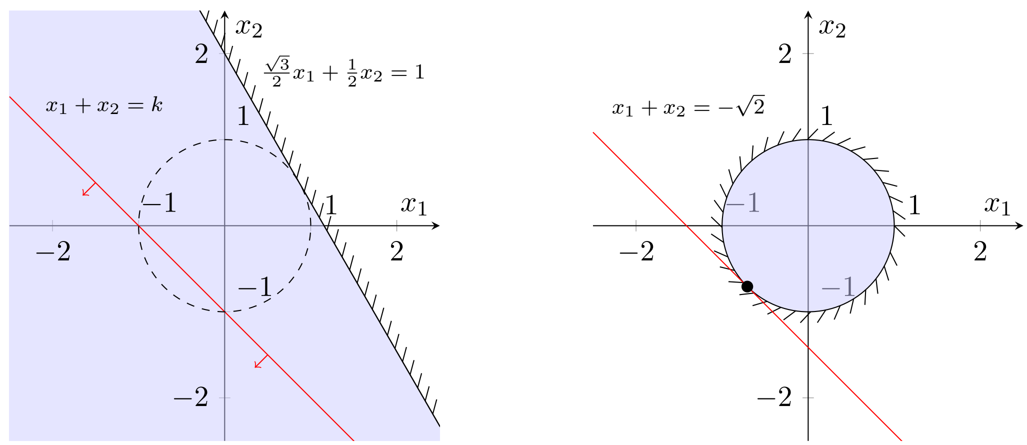
2.2. Stationarity Conditions for Nonlinear Robust Optimization

First-order optimality conditions for (1.1) have been derived by John; see, for example, [47, 68]. We start by defining the active index set for the active constraints,

$$\mathcal{U}_0(x^*) := \{u \in \mathcal{U} \mid c(x^*; u) = 0\}.$$

Theorem 2.1 (Stationarity Conditions for Robust Optimization [47, 68]). *Let x^* be a local minimizer of* (1.1) *(and $\mathcal{X} = \mathbb{R}^n$). Then there exist a finite subset $U_0'(x^*) \subset \mathcal{U}_0(x^*)$ and multipliers $\lambda_i \ge 0$ for each $u^{(i)} \in U_0'(x^*)$ such that*

$$\lambda_0 \nabla f(x^*) + \sum_{u^{(i)} \in U_0'(x^*)} \lambda_i \nabla_x c(x^*; u^{(i)}) = 0, \quad \lambda_0 + \sum_{u^{(i)} \in U_0'(x^*)} \lambda_i = 2. \tag{2.3}$$



(a) Unbounded nominal problem (b) Bounded NRO

Figure 1.: Unbounded nominal problem and bounded NRO.

We note, that the constant "2" in the multiplier sum can be replaced by any positive number, but our choice simplifies the derivation of stationarity conditions for problems with implementation errors in Section 2.3.

We say that x^* is a stationary point of (1.1) if x^* satisfies (2.3). We note that for $\lambda_0 = 0$, these conditions are related to the stationarity conditions of the robust nonlinear feasibility problem

$$\underset{x}{\text{minimize}} \max_{u \in \mathcal{U}(x)} \tfrac{1}{2}\left\|c(x;u)\right\|_2^2.$$

Moreover, if x^* satisfies an extended constraint qualification, we can replace the Fritz-John condition in Theorem 2.1 with the Karush-Kuhn-Tucker condition [47]

$$\nabla f(x^*) + \sum_{u^{(i)} \in U_0'(x^*)} \lambda_i \nabla_x c(x^*; u^{(i)}) = 0$$

with $\lambda_i \geq 0$ for each $u^{(i)} \in U_0'(x^*)$.

The stationarity condition (2.3) is not as useful as standard Fritz-John conditions in nonlinear optimization because the index set $\mathcal{U}_0(x^*)$ can contain an infinite number of points and has no closed-form characterization. Moreover, even given $\mathcal{U}_0(x^*)$, one still needs to find $u^{(i)}$ and λ_i simultaneously to satisfy (2.3), which in general is a set of nonlinear equations.

We can show that x^* is a stationary point of (1.1) if and only if $d_x = 0$ solves a linearized robust optimization problem:

Theorem 2.2. *A robust feasible point, x^*, is a stationary point of* (1.1) *if, and only if, $d_x = 0$ solves the following linearized problem:*

$$\begin{array}{ll} \underset{d_x}{\textit{minimize}} & \nabla f(x^*)^T d_x \\ \textit{subject to} & c(x^*;u) + \nabla_x c(x^*;u)^T d_x \leq 0, \quad \forall u \in \mathcal{U}(x) \\ & x^* + d_x \in \mathcal{X}. \end{array} \tag{2.4}$$

Proof. We start by stating the stationarity conditions of (2.4) at $d_x = 0$. If $d_x = 0$ solves (2.4), then it follows again by Theorem 2.1 that there exist a finite subset $U'' \subset \mathcal{U}(x^*)$ and multipliers $\nu_i \geq 0$ for all $u^{(i)} \in U''$ such that

$$\nu_0 \nabla f(x^*) + \sum_{u^{(i)} \in U''} \nu_i \nabla_x c(x^*; u^{(i)}) = 0, \quad \nu_0 + \sum_{u^{(i)} \in U_0'} \nu_i = 2, \tag{2.5}$$

holds.

Next, we show that if x^* is a stationary point of (1.1), then $d_x = 0$ solves (2.4). If x^* is stationary, then it follows that $c(x^*; u) \leq 0,\ \forall u \in \mathcal{U}(x)$, which implies that $d_x = 0$ is feasible in (2.4). To see that d_x is also a stationary point, we simply compare the stationarity conditions (2.5) and (2.3).

Finally, we show that if d_x solves (2.4), then x^* is a stationary point of (1.1). The equivalence of the first-order conditions can be seen by comparing (2.5) and (2.3), and feasibility of x^* follows from the feasibility of $d_x = 0$:

$$c(x^*; u) + \nabla_x c(x^*; u)^T d_x \leq 0,\ \forall u \in \mathcal{U}(x) \quad \Rightarrow \quad c(x^*; u) \leq 0,\ \forall u \in \mathcal{U}(x),$$

which concludes the proof. $\square$

It may seem surprising that Theorem 2.2 does not require any constraint qualification or conditions on $\mathcal{U}(x)$ to hold. However, these conditions are implicitly assumed in the stationarity of x^*, which implies the existence of multipliers.

To the best of our knowledge, the stationarity condition of Theorem 2.2 is new. However, we note that unless $\mathcal{U}(x)$, $c(x^*; u)$, and $\nabla_x c(x^*; u)$ have special structure, the linearized robust optimization problem (2.4) is not necessarily easier to solve than (1.1), because it is still a nonlinear problem in u. We observe that if $\mathcal{U}(x) = \mathcal{U}$ is independent of x and polyhedral or conic, then (2.4) is a tractable linear or conic optimization problem; see Section 4.2. A simpler stationarity condition is obtained if we consider the active constraints, replacing $u \in \mathcal{U}$ by $u \in \mathcal{U}_0(x^*)$ or even $u \in U_0'(x^*)$ in (2.4).

2.3. Stationarity Conditions for Problem with Implementation Errors

In many applications, we are interested in decision variables x that are robust to manufacturing errors or general implementation errors. Problems of this kind can be expressed as a minimax optimization problem

$$\underset{x \in \mathcal{X}}{\text{minimize}} \ \max_{u \in \mathcal{U}(x)} \ f(x+u), \tag{2.6}$$

which could be expressed equivalently in the form of (1.1) as

$$\begin{aligned} \underset{x \in \mathcal{X}, t \in \mathbb{R}}{\text{minimize}} \quad & t \\ \text{subject to} \quad & f(x+u) - t \leq 0, \quad \forall u \in \mathcal{U}(x). \end{aligned} \tag{2.7}$$

We note that $\mathcal{U} = \mathcal{U}(x)$ may depend on the decision variables x; but in the existing literature that explicitly considers (2.6), this is not the case.

A characterization of robust local minima, as well as descent directions at a point x, for the minimax function given in (2.6) can be found in [20]. The work in [20] considers the special case where $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{U} = \{u \mid \|u\|_2 \leq \Delta\}$ for some $\Delta > 0$, because this particular uncertainty set allows for easily stated sufficient conditions for a point in $\mathbb{R}^n$ to be a robust local minimum of (2.6). For general $\mathcal{U}(x)$, we can describe necessary conditions for a point x being a robust local minimum by using the conditions of Theorem 2.1 applied to (2.7); these conditions in this case are equivalent to the existence of $\lambda_i \geq 0$ such that

$$\begin{aligned} \sum_{u^{(i)} \in U_0'(x^*)} \lambda_i \nabla f(x^* + u^{(i)}) &= 0 \\ \sum_{u^{(i)} \in U_0'(x^*)} \lambda_i &= 1. \end{aligned} \tag{2.8}$$

One can show that these necessary conditions are in fact equivalent to the sufficient conditions derived in [20] given their particular choice of $\mathcal{U}$.

Observing that in (2.8), the set $U_0'(x^*)$ is equivalent to the set

$$\operatorname*{argmax}_{u \in \mathcal{U}(x^*)} f(x^* + u),$$

there is a natural geometric interpretation of these conditions, which we illustrate in Figure 2 given $\mathcal{X} = \mathbb{R}^2$ and $\mathcal{U} = \{u \mid \|u\|_2 \leq \Delta\}$. In Figure 2b, we have $|U_0'(x^*)| = 3$, and the corresponding gradients $\nabla f(x^* + u^{(i)})$ positively span $\mathbb{R}^2$. Hence, the necessary (and in this case, sufficient) conditions in (2.8) are satisfied. In Figure 2c, we have $|U_0'(x^*)| = 1$ (and $u^{(1)}$ occurs in the interior of $\mathcal{U}(x^*)$); because $\nabla f(x^* + u^{(1)}) = 0$, the conditions in (2.8) are also satisfied in this case. The example in Figure 2c also illustrates how for nonconvex f, the concept of a robust local mimimum can be practically dissatisfying, as there is an open neighborhood about x^* so that every point in the neighborhood is also a robust local minimum for (2.6).

In Figure 2a, we observe that given $\nabla f(x+u^{(1)})$ and $\nabla f(x+u^{(2)})$, there do not exist nonnegative multipliers such that (2.8) can be satisfied, implying that x cannot be a robust local minimum. Although we will not go into the algebraic details here, Figure 2a also illustrates the related concept of a cone of descent directions for (2.6) at a point x. It is geometrically intuitive that the shaded area of Figure 2a is a cone of descent, since for any small perturbation s such that $x + s$ is in the cone, neither $x + u^{(1)}$ nor $x + u^{(2)}$ will be in $\mathcal{U}(x + s)$. Thus, the maximum value of the inner problem of (2.6) given $\mathcal{U}(x + s)$ is strictly bounded above by the maximum value of the inner problem of (2.6) given $\mathcal{U}(x)$, implying that s is a descent direction.

2.4. Distributionally Robust Optimization

A topic of increasing interest in the past decade has been distributionally robust stochastic optimization (DRSO). The typical problem of stochastic optimization, without any reformulations,

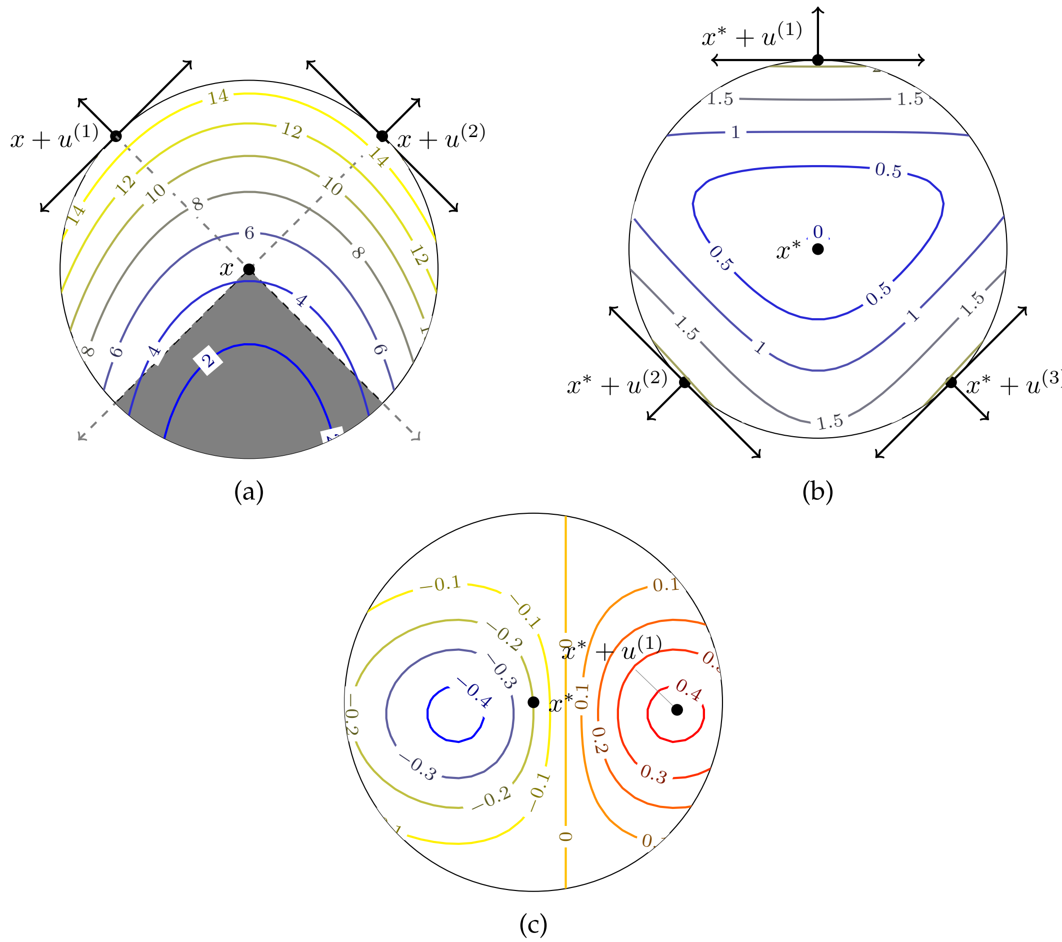


Figure 2.: Geometric intuition for (a) descent directions of (2.6) at a point x, (b),(c) x being a robust local minimum of (2.6)

can be cast as an unconstrained problem

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad \mathbb{E}_\pi \Big[f(x, \pi(\omega)) \Big], \tag{2.9}$$

where $\mathbb{E}$ is the expectation operator, $\pi : \Omega \to \mathbb{R}^d$ is a measurable probability distribution, Ω is a sample space, and the objective function is a mapping $f : \mathcal{X} \times \mathbb{R}^d \to \mathbb{R}$. Generally, however, a modeler does not have access to a closed-form distribution π, and it is thus desirable to consider a robust version of (2.9)

$$\underset{x \in \mathcal{X}}{\text{minimize}} \ \max_{\pi \in \mathcal{P}} \quad \mathbb{E}_\pi \Big[f(x, \pi(\omega)) \Big], \tag{2.10}$$

introducing an uncertainty set $\mathcal{P}$ of possible distributions. As in (2.7), the minimax problem in (2.10) can be reformulated as a problem of the form (1.1).

Much of the existing research in DRSO focuses on developing uncertainty sets $\mathcal{P}$ so that the solution of (2.10) is tractable. A dominant strategy [16, 28] considers first- and second-order moments μ and Σ of empirically observed realizations of $\pi(\omega)$ and $\mathcal{P}$ is defined as a set of probability distributions having first- and second-order moments "close" to μ and Σ. A more recent, but promising, strategy [32, 65] assumes that some reference nominal distribution π is known, and then defines $\mathcal{P}$ as the set of distributions "close" to π in a measure-theoretic sense, the so-called Wasserstein distance. We will not discuss either of these strategies in any further detail, but we point to the cited papers and the references therein.

We remark on a direction of research that is also referred to as distributionally robust optimization but is notably different from the one presented in (2.10), which led to the development of the ROME software package [40, 41]. That body of work is concerned with nominal linear optimization problems and the tractable reformulations of robustified linear constraints. In [40, 41], uncertainty sets may be constructed to leverage knowledge of distributional properties of the uncertainty such as bounds on moments, bounds on the distributional support, and directional deviations. The work in [40, 41] is also extended to compute nonanticipative (but relatively simple) decision rules for multistage problems while maintaining tractability.

3. Applications and Illustrative Examples

Here we provide examples of nonlinear robust optimization problems considered in the literature. We emphasize cases with infinite-cardinality uncertainty sets, but we note that finite-cardinality uncertainty set examples are prevalent; see, for example, the minimax regret and test problems in [33,45,46]. The illustrative examples in Section 3.2 are used in later sections to demonstrate the solution techniques and available software.

3.1. Applications of Nonlinear Robust Optimization

Robust convex quadratically constrained optimization problems have been solved for applications including financial portfolio selection problems [43], equalization of time-invariant communication channels [62], and statistical learning problems [42,49].

Designing truss and frame structures under uncertain loads [8] and design of antenna arrays [14] have also been addressed from a nonlinear optimization perspective.

A number of control problems involving uncertain initial and state conditions result in nonlinear robust optimization problems. Spacecraft attitude control was addressed through a minimax approach in [25] and the scheduling of industrial processes was the subject of [54]. The chemical engineering problem of batch distillation was considered by using an elliptic uncertainty set in [29,30].

Robust optimization has also been considered in situations where the objective function or uncertain constraints are available only through the output of a simulation. Examples of these include the design of a DC–DC converter [26] and electromagnetic matching for nanophotonic engineering [19,20].

3.2. Illustrative Examples

We use the following examples throughout the paper to illustrate our approaches. These problems all have objective functions that are smooth on the appropriate sets and have nonempty, compact uncertainty sets. In addition, they satisfy the following conditions:

C1 The certain feasible set $\mathcal{X}$ is convex and $f(x)$ is a convex function.
C2 The uncertainty set $\mathcal{U}(x)$ is convex and compact for all x.
C3 The robust constraints $c(x;u) \leq 0$ are convex in x and concave in u.

We discuss the implications if some of the convexity assumptions are relaxed in Section 6.

Our first example is a simple two-dimensional problem in which the uncertainty set is independent of the variables, x.

Example 3.1. *Consider the following robust optimization problem illustrated in Figure 3.*

$$\begin{cases} \underset{x \geq 0}{\text{minimize}} & (x_1-4)^2 + (x_2-1)^2 \\ \text{subject to} & x_1\sqrt{u} - x_2 u \leq 2, \qquad \forall u \in \left[\frac{1}{4}, 2\right] \end{cases} \tag{3.11}$$

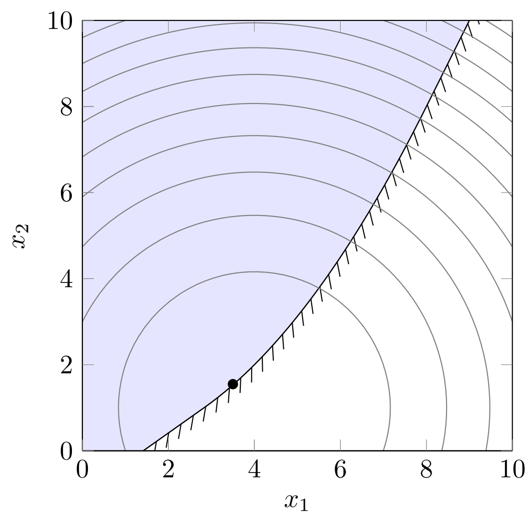


Figure 3.: The two-dimensional robust optimization problem from Example 3.1

Our second example is built from a 3-SAT problem, and exemplifies the situation when the uncertainty set also depends on th variables, x.

Example 3.2. *Consider the following special case of a robust 3-SAT problem [61],*

$$\begin{cases} \underset{x \in \mathcal{X}}{\text{minimize}} & -x_1 \\ \text{subject to} & x_1 - u_1 x_5 - u_2 x_6 \leq 0, \quad \forall u \in \mathcal{U}(x), \end{cases} \tag{3.12}$$

where the uncertainty set depends linearly on the certain variables (x_2, x_3, x_4)*,*

$$\mathcal{U}(x) = \left\{ u \;\middle|\; u_1 \geq x_2, u_1 \geq x_3, u_1 \geq 1 - x_4, u_1 \leq 1, u_2 \geq x_2, u_2 \geq 1 - x_3, u_2 \geq 1 - x_4, u_2 \leq 1 \right\},$$

and $\mathcal{X} = \{x : 0 \leq x_1 \leq 2;\ 0 \leq x_2, x_3, x_4, x_5 \leq 1\}$.

4. Solution Approaches

Several approaches for solving nonlinear robust optimization problems have been studied in the literature. We start by formulating the robust optimization problem as a bilevel optimization problem and then applying a reformulation that results in a mathematical program with equilibrium constraints (MPEC). Next, we discuss a series of reformulations of robust constraints that employ duality and result in equivalent "tractable" constraints. Then, we provide a method based on outer approximations, and, finally, we discuss methods for robust optimization problems with implementation errors.

4.1. Bilevel Approach to Robust Optimization

Following the approach in [67] to semi-infinite optimization, we show how to formulate robust counterparts for (1.1) as bilevel optimization problems. We start with the following form of (1.1):

$$\begin{array}{ll} \underset{x \in \mathcal{X}}{\text{minimize}} & f(x) \\ \text{subject to} & c(x; u) \leq 0, \quad \forall u \in \mathcal{U}(x) := \{u \mid g(x; u) \leq 0\}, \end{array} \tag{4.13}$$

which is equivalent to requiring that the maximum of $c(x; u)$ over $\mathcal{U}(x)$ be nonpositive. We can write (4.13) equivalently as

$$\begin{array}{ll} \underset{x \in \mathcal{X}}{\text{minimize}} & f(x) \\ \text{subject to} & \max\limits_{u} \left\{ c(x; u) \mid g(x; u) \leq 0 \right\} \leq 0, \end{array} \tag{4.14}$$

which is a bilevel optimization problem. Since the functions are smooth and the uncertainty set is nonempty and compact for each $x \in \mathcal{X}$ by assumption, the lower-level optimization problem has a solution, and the objective has a finite value. We note, that if we do not assume that the uncertainty set is nonempty and bounded, then the lower-level problem can be infeasible or have an unbounded objective function value and/or the norm of u at the "solution" may be infinite.

Assuming further that for all $x \in \mathcal{X}$, $c(x; u)$ is concave in u and $g(x; u)$ is convex in u (the lower-level problem is a convex optimization problem) and $g(x; u)$ satisfies a constraint qualification for all $x \in \mathcal{X}$, then we can use the first-order conditions of the lower-level problem in (4.14) to obtain the equivalent optimization problem with equilibrium constraints,

$$\begin{array}{ll} \underset{x \in \mathcal{X}, u, \lambda}{\text{minimize}} & f(x) \\ \text{subject to} & c(x; u) \leq 0 \\ & \nabla_u \mathcal{L}(x; u, \lambda) = 0 \\ & 0 \leq \lambda \ \perp \ -g(x; u) \geq 0, \end{array} \tag{4.15}$$

where the Lagrangian of the lower-level problem is

$$\mathcal{L}(x; u, \lambda) := c(x; u) - \lambda^T g(x; u)$$

and $\perp$ indicates the complementarity slackness condition (componentwise, either the left or right inequality is active). Under the assumptions made on the lower-level problem, any global solution to (4.15) is a solution to (1.1), while all other feasible points provide upper bounds on the true robust solution. Global infeasibility of (4.15) implies that (1.1) is infeasible, while local infeasibility yields no information.

Approaches to solving (4.15) involve replacing the complementarity condition with an equivalent inequality [7, 15, 35–37, 51–53, 55, 64] or exploring the possibly exponential number of subproblems that resolve the complementarity slackness conditions. Smooth reformulations of the complementarity condition result in nonconvex nonlinear optimization problems that do not satisfy traditional constraint qualifications. Nevertheless, nonlinear programming solvers can compute local solutions. Global solutions to the resulting nonconvex problems using methods based on convex relaxations of the constraints require compact feasible regions, necessitating further assumptions on the constraint qualification to ensure that the Lagrange multipliers in the lower-level problem are bounded.

We illustrate this approach using Example 3.1:

Example 4.1. *Consider the robust optimization problem in Example 3.1. Then it follows that the MPEC formulation is obtained as:*

$$\left\{ \begin{array}{ll} \underset{x \geq 0, u, l^+, l^-}{\textit{minimize}} & (x_1 - 4)^2 + (x_2 - 1)^2 \\ \textit{subject to} & x_1 \sqrt{u} - x_2 u \leq 2 \\ & \frac{x_1}{2\sqrt{u}} - x_2 + l^+ - l^- = 0 \\ & 0 \leq l^+ \ \perp \ u - \frac{1}{4} \geq 0 \\ & 0 \leq l^- \ \perp \ 2 - u \geq 0. \end{array} \right. \tag{4.16}$$

4.2. Reformulations of Classes of Robust Optimization Problems

Many classes of uncertain constraints can be reformulated equivalently as finite-dimensional deterministic optimization problems by using duality. These derivations were first proposed in a series of papers by Ben-Tal and Nemirovski [9–14] who refer to these reformulations as "tractable robust constraints". Tractability refers to the fact that the reformulated problem can be solved in polynomial time provided that all other problem functions allow polynomial-time algorithms.

As in Section 4.1, the derivation starts from (4.14), and we assume that for all $x \in \mathcal{X}$, $c(x;u)$ is concave in u and $g(x;u)$ is convex in u and that $g(x;u)$ satisfies a constraint qualification for all $x \in \mathcal{X}$. Rather than writing the first-order conditions of the lower-level problem, we instead form its dual, such as the Lagrangian or Fenchel dual. In particular, robust counterparts of nonlinear uncertain constraints for general convex functions can be found by exploiting the support function δ^* and the concave conjugate function c_* [9]. The authors show that

$$c(x;u) \le 0, \quad \forall u \in \mathcal{U} := \{u \mid Du + q \ge 0\}$$

if and only if

$$x, v \text{ satisfy } \delta^*(v \mid \mathcal{U}) - c_*(x;v) \le 0.$$

This result allows general convex sets. In general, however, no closed-form expressions exist either for the conjugate of a convex function or for the support function. [9, Table 3] lists expressions for conjugates of some simple functions. We illustrate this approach with the Wolfe dual, and arrive at the problem

$$\begin{aligned} &\underset{x\in\mathcal{X}}{\text{minimize}} && f(x) \\ &\text{subject to} && \min_{u,\lambda} \left\{ \mathcal{L}(x;u,\lambda) \mid \nabla_u \mathcal{L}(x;u,\lambda) = 0,\ \lambda \ge 0 \right\} \le 0, \end{aligned} \tag{4.17}$$

where the Lagrangian of the lower-level problem is

$$\mathcal{L}(x;u,\lambda) := c(x;u) - \lambda^T g(x;u)$$

and $\lambda \ge 0$ are the Lagrange multipliers of the lower-level constraints. We can now omit the inner minimization because if we find any (u,λ) such that $\mathcal{L}(x;u,\lambda) \le 0$, then it follows that the minimum is nonpositive. Thus, we arrive at the single-level problem

$$\begin{aligned} &\underset{x\in\mathcal{X},u,\lambda}{\text{minimize}} && f(x) \\ &\text{subject to} && \mathcal{L}(x;u,\lambda) \le 0 \\ &&& \nabla_u \mathcal{L}(x;u,\lambda) = 0 \\ &&& \lambda \ge 0. \end{aligned} \tag{4.18}$$

For nonlinear functions, the resulting problem is typically a nonconvex optimization problem. Under the assumptions made on the lower-level problem, any global solution to (4.18) is a solution to (1.1), while all other feasible points provide upper bounds on the true robust solution. If (4.18) is globally infeasible and there is no duality gap, then (1.1) is also infeasible. If either there is a duality gap or (4.18) is only locally infeasible, then we cannot draw any conclusions. We note that for some nonconvex problems, one can also show that the duality gap is zero, see, e.g. [24].

A connection exists between (4.18) and (4.15). We observe that both have the condition that $\nabla_u \mathcal{L} = 0$ and $\lambda \ge 0$. Adding $\lambda^T g(x;u)$, which equals zero from the complementarity slackness condition, to $c(x;u)$ shows that (4.18) is a relaxation of (4.15). The difference, however, is the conclusions that can be drawn when these two problems are globally infeasible.

We illustrate the Wolfe-dual approach using Example 3.1.

Example 4.2. *Consider the robust optimization problem in Example 3.1. Then it follows that the Wolfe-dual formulation is given by*

$$\begin{cases} \underset{x\geq 0,u,l^+\geq 0,l^-\geq 0}{\text{minimize}} & (x_1-4)^2+(x_2-1)^2 \\ \text{subject to} & x_1\sqrt{u}-x_2u-2+l^+(u-\frac{1}{4})+l^-(2-u)\leq 0 \\ & \frac{x_1}{2\sqrt{u}}-x_2+l^+-l^-=0, \end{cases} \tag{4.19}$$

Our general form (4.18) recovers the robust counterpart of a linear robust constraint.

Example 4.3. *Consider the following problem:*

$$\begin{array}{ll} \underset{x\in\mathcal{X}}{\text{minimize}} & f(x) \\ \text{subject to} & (a+Pu)^T x \leq b, \quad \forall u\in\mathcal{U}:=\{u \mid Du+q\geq 0\}, \end{array} \tag{4.20}$$

where P, D are matrices of suitable dimensions such that $\mathcal{U}$ is compact. We see that (4.20) *becomes*

$$\begin{array}{ll} \underset{x\in\mathcal{X},u}{\text{minimize}} & f(x) \\ \text{subject to} & \max\limits_{u}\left\{(a+Pu)^T x-b \mid Du+q\geq 0\right\}\leq 0, \end{array}$$

which is equivalent to the Wolfe-dual problem

$$\begin{array}{ll} \underset{x\in\mathcal{X},u,\lambda}{\text{minimize}} & f(x) \\ \text{subject to} & \min\limits_{u,\lambda}\left\{(a+Pu)^Tx-b+\lambda^T(Du+q) \mid P^Tx+D^T\lambda=0,\lambda\geq 0\right\}\leq 0. \end{array}$$

Exploiting the fact that $P^Tx+D^T\lambda=0$, we arrive at the following tractable formulation:

$$\begin{array}{ll} \underset{x\in\mathcal{X},\lambda}{\text{minimize}} & f(x) \\ \text{subject to} & a^Tx+\lambda^Tq\leq b \\ & P^Tx+D^T\lambda=0,\lambda\geq 0, \end{array}$$

whose constraints are a (finitely generated) polyhedral set. This approach has been generalized to other forms of polyhedral constraints that we summarize in Table 1.

Unfortunately, these reformulations often result in significantly more-complex constraints, as illustrated by the following example.

Example 4.4. *Robust convex quadratic optimization problems of the form*

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\left\{c^Tx \mid \frac{1}{2}x^TQx+x^Tg+\gamma\leq 0, \quad \forall(Q,g,\gamma)\in\mathcal{U}\right\} \tag{4.21}$$

were first reformulated as equivalent semidefinite programming (SDP) problems in [11]. Problems of this form can also be cast as equivalent second-order cone programming (SOCP) problems. For example, [42] show that for polytopic and factorable uncertainty sets, as well as affine uncertainty sets of the form

$$\mathcal{U} = \Big\{ (Q, g, \gamma) \mid Q = Q^0 + \sum_{i=1}^{p} \lambda_i Q^i,\ (g, \gamma) = (g^0, \gamma^0) + \sum_{i=1}^{p} v_i (g^i, \gamma^i), \\ \|\lambda\| \le 1,\ \|v\| \le 1,\ Q^i \succeq 0,\ \forall i \Big\}, \tag{4.22}$$

one obtains an equivalent SOCP formulation for (4.21).

Classes of Tractable Robust Constraints. An overview of tractable robust constraints is found in Table 1. The reformulations depend on the specific form of the uncertain constraint $c(x;u)$ and the specific form of the uncertainty set $\mathcal{U}$. These reformulations may be computationally more expensive than other approaches [18]. A more detailed form of Table 1 can be found in Tables 1 and 2 of [9], which provide classes of tractable reformulations for uncertainty sets and problem functions, respectively.

Table 1.: Reformulations of robust constraints.

Uncertain Constraint	Uncertainty Set	Tractable Constraint	Ref.
Affine $(a+Pu)^T x \le b$	Polyhedral $\{Du + q \ge 0\}$	Polyhedral $a^T x + q^T \lambda \le b$ $P^T x + D^T \lambda = 0, \lambda \ge 0$	[11]
Affine $(a+Pu)^T x \le b$	ℓ_∞-Box $\{\|u\|_\infty \le \rho\}$	Polyhedral $a^T x + \rho \|P^T x\|_1 \le b$	[11]
Affine $(a+Pu)^T x \le b$	ℓ_2-Ball $\{\|u\|_2 \le \rho\}$	Conic constraint $a^T x + \rho \|P^T x\|_2 \le b$	[11]
Affine $(a+Pu)^T x \le b$	Closed convex pointed cone K $\{Du + q \in K\}$	Conic constraint $a^T x + q^T \lambda \le b$ $P^T x + D^T \lambda = 0, \lambda \in K^*$	[13]
Convex quadratic $x^T A(u) x + b(u)^T x + c \le 0$	Convex set, $u \in C$ $A(u) = A + U, b(u) = b + U\hat{b}$	Semidefinite constraint	[58]
Conic quadratic $\sqrt{x^T A(u) x} + b(u)^T x + c \le 0$	Convex set, $u \in C$ $A(u) = A + U, b(u) = b + U\hat{b}$	Semidefinite constraint	[58]
Convex quadratic	Ellipsoid	SDP	[11]
Conic QP	Ellipsoid	SDP	[11]
SDP	Structured ellipsoid	SDP	[11]
Convex function	Convex set	Conjugate convex	[9]

Classes of Intractable Robust Constraints. Classes of robust-constraint and uncertainty-set combinations that result in optimization problems that are NP-hard, and hence deemed intractable, are also identified in [11]. Examples include the intersection of ellipsoids and general semi-definite robust constraints.

A special class of problems is decision-dependent uncertainties. However, it is not clear whether tractable reformulations exist, even in the most simple cases of uncertainty set. To illustrate this point, we consider a general form of Example 3.12, which has an affine robust constraint over an affine set. In general, we can formulate constraints of this form as

$$(a + Pu)^T a \leq b, \quad \forall u \in \mathcal{U}(x) := \{u \mid Du + q + Ex \geq 0\}.$$

In the tractable formulations discussed above, we had assumed that $E = 0$. If this is not the case, then we can still apply the duality mechanism to derive an equivalent finite set of constraints as

$$a^T x + q^T \lambda + \lambda^T E x \leq b, \ P^T x + D^T \lambda = 0, \ \lambda \geq 0.$$

Unfortunately, this set is no longer polyhedral, because of the presence of the bilinear term $\lambda^T E x$, which is in general a nonconvex term.

4.3. Methods of Outer Approximations

A method of outer approximations with first-order convergence guarantees the existence of a solution of (1.1), under the assumptions that

O1 $f(\cdot)$ and $c(\cdot, \cdot)$ are continuous on an open set containing $\mathcal{X}$.
O2 $\nabla_x f(\cdot)$ and $\nabla_x c(\cdot, \cdot)$ exist for all $x \in \mathcal{X}$ and are continuous on an open set containing $\mathcal{X}$.
O3 The uncertainty set $\mathcal{U}$ is compact.

Notice that these assumptions do not impose convexity of f, c, or $\mathcal{U}$. The idea of the iterative method is fairly straightforward; in the kth iteration, given a point (current iterate) $x_k \in \mathbb{R}^n$, a finite sample set $S_k \subseteq \mathcal{U}$ is algorithmically determined. This particular sample induces a subproblem

$$\begin{aligned} \underset{x \in \mathcal{X}}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x; u) \leq 0, \quad \forall u \in S_k. \end{aligned} \tag{$P(S_k)$}$$

The subproblem ($P(S_k)$) is a relaxation of (1.1). It is assumed that, given a set S_k, the subproblem ($P(S_k)$) is tractable and a nonlinear optimization algorithm exists for which an ϵ-accurate solution in terms of first-order KKT stationarity can be obtained. Additionally, problem structure could be leveraged here; for instance, in the case where $f(x)$ is a convex function and the constraints of ($P(S_k)$) define a convex set, then a convex solver can be applied to attain a global minimum. Additionally, if the relaxation ($P(S_k)$) is infeasible, then (1.1) is clearly infeasible and the algorithm may terminate early.

The outer approximations method also requires, in each iteration, a selection of a set $U_k \subseteq \mathcal{U}$. For convergence results to hold, one must be able to compute, given a current iterate x_k, the global maximum to $\{\max c(x_k; u) : u \in U_k\}$. For problems where $c(x_k; u)$ is concave in u for all $u \in \mathcal{U}$, then selecting $U_k = \mathcal{U}$ for all k may not be unreasonable. Similarly, if one knows from particular problem structure that $c(x_k; u)$ is concave on a subset $\mathcal{U}'_k \subset \mathcal{U}$, then one might select $U_k = \mathcal{U}'_k$ and solve the maximization problem to global optimality. In general, however, if global optimality cannot be guaranteed, such as in the case of (2.6) where f is nonconvex, then one may use finite sets $U_k \subset \mathcal{U}$. For convergence in the general case, however, one needs to ensure that $U_k \to \mathcal{U}$, namely, that some form of asymptotic density holds.

A statement of an inexact method of outer approximations is given in Algorithm 1.

```
Choose initial point x_1 ∈ ℝ^n.
Choose a sequence ϵ_k → 0 such that ϵ_k > 0 for all k, and set k ← 0.
Choose s_0 ∈ 𝒰.
for k = 0, 1, 2, ... do
    Choose a finite set S_k satisfying {s_0, ..., s_k} ⊂ S_k.
    Let x_k be an ϵ_k-accurate solution to (P(S_k)).
    Choose U_k ⊂ 𝒰.
    Let s_k be a global maximizer of arg max_{u∈U_k} c(x_k; u).
    k ← k + 1.
end
```

Algorithm 1: Method of Outer Approximations.

It has been proved (e.g., in Chapter 3.5 of [63]), that every accumulation point of Algorithm 1 applied to (1.1), under a rigorous version of the previously stated assumptions, satisfies a first-order stationary condition.

4.4. Algorithms for Robust Optimization with Implementation Errors

An algorithm for solving NROs with implementation errors is proposed in [20]. The algorithm iteratively solves a sequence of second-order conic optimization subproblems intending to find descent directions as in Figure 2a. The algorithm assumes access only to an oracle capable of function and gradient evaluations of f. Provided f is convex and continuously differentiable, the authors show that the proposed algorithm converges to the global optimum of (2.6). This method is augmented by a simulated annealing method in [19] in an attempt to offer asymptotic global guarantees for nonconvex problems.

The black-box algorithm in [27] considers (2.6) when only function evaluations (i.e., no gradient evaluations) of f are available. The algorithm in that work alternates between obtaining approximate local minima and maxima to the outer and inner problems of (2.6), respectively, via smooth model-based trust-region optimization methods. No convergence guarantees on the proposed method are made. A recent unpublished work [59] considers (2.6) again under black-box assumptions. The algorithm in that work is based on a method of outer approximations, as in Section 4, and solves a sequence of nonsmooth optimization problems over local surrogate models. The iterates of the algorithm are shown to cluster at Clarke stationary points of the minimax function of (2.6), a condition that is directly related to the robust local minima of [20] but for more general $\mathcal{U}(x)$. The convergence result in [59] does not assume f to be convex.

A recently published work [69] handles (2.6) with (optional) additional robust constraints of the form

$$c(x+u) \leq 0 \;\forall u \in \mathcal{U}. \tag{4.23}$$

The method targets more general robust optimization problems and proposes building separate metamodels of the objective in (2.6) and the constraints in (4.23) via kriging. Solving these surrogate problems is then passed to a (relatively) inexpensive global optimization operation.

5. Software for Robust Optimization

Modeling languages for robust optimization serve as an intermediate layer between the modeler and numerical solvers. Such languages usually implement several strategies for instantiating an uncertain problem into a tractable certain problem. We list here relevant major modeling languages.

- AIMMS [23] is an integrated combination of a modeling language, a graphical user interface, and numerical solvers. It supports deterministic robust optimization and distributionally robust optimization (the model includes chance constraints whose probability is associated with the specific distribution).
- JuMPeR [1] is an algebraic modeling toolbox for robust and adaptive optimization in Julia. It extends the syntax of JuMP. Its resolution techniques include cutting planes.
- ROC [21] is a C++ software package for formulating and solving distributionally adaptive optimization models.
- ROME [40, 41] (Robust Optimization Made Easy) is an algebraic modeling toolbox in Matlab. It implements distributionally robust optimization (parameterized by classical properties, such as moments, support, and directional deviations) and robust decision-making, in which the uncertainties are progressively revealed.
- ROPI [39] is a C++ library for solving robust mixed-integer linear problems modeled in the MPS file format.
- SIPAMPL [70] is an environment that interfaces AMPL with a semi-infinite programming solver. Uncertain parameters (or infinite variables) are represented by names starting with `t`, and constraints that involve uncertain parameters are represented by names starting with `t` (see Listing 1 for the code for Example 3.1).

Listing 1: SIPAMPL model

```
var x {1..2} >= 0;                          # decision variables
var t;                                      # uncertain parameter

minimize fx: (x[1] - 4)^2 + (x[2] - 1)^2;   # deterministic objective
```

```
subject to
   tcons: x[1]*sqrt(t) - x[2]*t <= 2;        # robust constraint
   bounds: 0.25 <= t <= 2;                   # uncertainty set
```

- YALMIP [56, 57] is a free Matlab toolbox, developed initially to model SDP problems and solve them by interfacing external solvers. It was later extended to deterministic robust optimization (see Listing 2) and distributionally robust optimization. YALMIP implements several strategies (called filters) for instantiating an uncertain problem into a tractable certain problem, including duality, enumeration, explicit maximization, conservative approximation, and elimination.

Listing 2: YALMIP model

```
sdpvar x1 x2 t                          % decision variables

constraints = [x1*sqrt(t) - x2*t <= 2,  % robust constraint
               x1 >= 0, x2 >= 0,
               0.25 <= t <= 2,          % uncertainty set
               uncertain(t)];           % uncertain parameter
objective = (x1 - 4)^2 + (x2 - 1)^2;    % deterministic objective
solvesdp(constraints, objective)
```

Table 2, inspired by [31], compares their features. To the best of our knowledge, no modeling language supports generalized semi-infinite optimization, including the case when we have decision-dependent uncertainty sets that depend on x.

Table 2.: Robust optimization modeling toolboxes.

Solver	Language	Open	Solvers	Uncertainty Sets	Constraints	Examples
AIMMS	AIMMS	✗	Many	box, ellipsoidal, convex	linear	[6]
JuMPeR	Julia	✓	Many	polyhedral, ellipsoidal, custom	linear	[2]
ROC	C++	✓	CPLEX	polyhedral, ellipsoidal	linear	
ROME	Matlab	✓	SDPT3, MOSEK, CPLEX	polyhedral, ellipsoidal	linear	[3]
ROPI	C++	✓	CPLEX, Gurobi, Xpress	finite set of scenarios	linear	
SIPAMPL	AMPL/Matlab	✓	NSIPS	box	nonlinear	[4]
YALMIP	Matlab	✓	Many	polyhedral, ellipsoidal, conic	linear, quadratic, 2nd order, semidefinite cone	[5]

6. Nonconvexity and Global Optimization

We are interested in investigating the challenges involved in extending the work surveyed in the preceding sections to nonconvex robust optimization. We consider only nonconvexities in the robust constraint and the uncertainty set, namely,

$$c(x;u) \leq 0, \quad \forall u \in \mathcal{U} := \{u \mid g(u) \leq 0\}.$$

We are interested in the tractability of this set of constraints.

The best-case situation arises when Assumptions **C2** and **C3** from Section 3.2 are satisfied. Under these assumptions, the problems arising in the sampling/outer approximations approach are convex minimization and concave maximization problems. Moreover, the duality gap is zero, and we can apply the reformulations of the MPEC section. Table 3 summarizes how this situation deteriorates if we relax the assumptions on $c(x;u)$. In general, the problems marked as nonconvex in Table 3 require global optimization techniques such as branch and bound, making them significantly harder than the best-case.

Table 3.: Properties of problems in Polak's outer approximation approach under different convexity assumptions, assuming that $\mathcal{U}$ is convex.

Property of $c(x;u)$		Sampling	Outer
x-convex	u-concave	Problem	Approx.
✓	✓	concave max.	convex min.
✓	✗	nonconvex	convex min.
✗	✓	concave max.	nonconvex
✗	✗	nonconvex	nonconvex

We note that nonconvexity/nonconcavity in x/u requires convex/concave under-/over-estimators to be built that are parameterized in u/x, respectively, as the following example illustrates.

Example 6.1. *Consider the following robust constraint,*

$$c(x;u) := x_2 - x_1^2 u \leq 0, \quad \forall u \in \left[\frac{1}{2}, \frac{3}{2}\right],$$

and assume that $\mathcal{X} = [0,1]^2$. *Then it follows that we can build secant relaxations for every* u *as*

$$\hat{c}(x;u) := x_2 - x_1 u \leq 0, \quad \forall u \in \left[\frac{1}{2}, \frac{3}{2}\right].$$

However, the situation becomes harder if the class of underestimator depends on the value of u. *For example, for* $\mathcal{U} = [-\frac{1}{2}, \frac{1}{2}]$, *it follows that* $c(x;u)$ *is convex in* x *for* $u \leq 0$, *and we need only the underestimator for* $u \geq 0$.

Table 3 also shows that the situation is more difficult for robust optimization problems that involve implementation errors, because $c(x;u) = c(x+u)$ is convex in x and concave in u if and only if it is affine.

We turn now to the case of nonconvex uncertainty sets $\mathcal{U}$. In this case, only the separable and partially linear case is easy, because

$$c(x;u) := c_1(x) + b^T u \leq 0, \ \forall u \in \mathcal{U} \quad \Leftrightarrow \quad c_1(x) + \max_{u \in \mathcal{U}} b^T u \leq 0.$$

Now, observe that a linear function attains its maximum at an extreme point of the feasible set, so we can equivalently maximize over the convex hull of $\mathcal{U}$, namely,

$$c(x;u) := c_1(x) + b^T u \leq 0, \ \forall u \in \mathcal{U} \quad \Leftrightarrow \quad c_1(x) + \max_{u \in \mathrm{conv}(\mathcal{U})} b^T u \leq 0.$$

Of course, finding the convex hull $\mathrm{conv}(\mathcal{U})$ is not trivial. Even worse, this simple trick already fails if we relax the linearity in u to concavity in u, namely, for $c(x;u) := c_1(x) + c_2(u)$ with $c_2(u)$ concave, because the maximum of a concave function no longer occurs at an extreme point, and replacing $\mathcal{U}$ by its convex hull will overestimate the uncertainty set and result in a conservative estimate.

7. Conclusion

Overall, nonlinear robust optimization problems, while naturally occurring and apparently of practical importance - see our discussion of various practical applications, implementation errors, and distributionally robust optimization - have not yet matured to the state of conic robust optimization problems. This maturity level is not surprising when viewed through the lens of reformulation. As we have discussed, affine constraints coupled with uncertainty sets of favorable shape lead to linear or conic optimization problems and are hence deemed tractable from the perspective of convergence rates to global optimality, whereas something as seemingly innocuous as decision-dependent uncertainty may lead to NP-hard reformulations. As the state of the art continues to improve for nonconvex (nonsmooth) optimization, more problems may become tractable in practice via solutions to MPEC reformulations of bilevel problems or applications of outer approximation methods.

The literature on nonlinear robust optimization is expanding and encompasses areas we did not consider in this survey. In some applications, we have the opportunity to take corrective action after (part of) the uncertainty is revealed. Robust optimization problems with this sort of structure fall into the class of two-stage decision problems [40, 41], where the decision variables x are first-stage, or here-and-now, decisions, and a second set of variables $y(u)$ represents wait-and-see, second-stage, or recourse decisions. Robust optimization is connected to robust model-predictive control [29, 30, 48, 50]. These are problems where the decision variables are time-dependent state and control variables that are governed by a system of differential algebraic equations that describe the dynamics of the underlying physical system. Robust optimization problems can also include integer decision variables [34].

To summarize, much opportunity exists for growth and novel research in the field of nonlinear robust optimization, driven by relevance and practical application, which will also spur further developments in general nonconvex (and NP-hard) optimization.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357.

References

[1] JuMPeR: algebraic modeling for robust and adaptive optimization. `https://github.com/IainNZ/JuMPeR.jl`.

[2] JuMPeR example - portfolio optimization. `http://jumper.readthedocs.io/en/latest/jumper.html#worked-example-portfolio-optimization`.

[3] ROME examples. `http://www.robustopt.com/examples.html`.

[4] SIPAMPL examples. `http://www.norg.uminho.pt/aivaz/binaries/Software/sipmod.zip`.

[5] YALMIP example - model predictive control. `https://yalmip.github.io/example/robustmpc/`.

[6] L. El Ghaoui A. Ben-Tal and A. Nemirovski. AIMMS example - production planning. `https://aimms.com/english/developers/resources/examples/practical-examples/production-planning-ro`.

[7] M. Anitescu. Global convergence of an elastic mode approach for a class of mathematical programs with complementarity constraints. SIAM Journal on Optimization, 16(1):120–145, 2005.

[8] A. Ben-Tal and A. Nemirovski. Robust truss topology design via semidefinite programming. SIAM Journal on Optimization, 7(4):991–1016, 1997.

[9] Aharon Ben-Tal, Dick den Hertog, and Jean-Philippe Vial. Deriving robust counterparts of nonlinear uncertain inequalities. Mathematical Programming, 149(1):265–299, Feb 2015.

[10] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. Robust Optimization. Princeton University Press, 2009.

[11] Aharon Ben-Tal and Arkadi Nemirovski. Robust convex optimization. Mathematics of Operations Research, 23(4):769–805, 1998.

[12] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of uncertain linear programs. Operations Research Letters, 25(1):1–13, 1999.

[13] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. Mathematical Programming, 88(3):411–424, 2000.

[14] Aharon Ben-Tal and Arkadi Nemirovski. Robust optimization–methodology and applications. Mathematical Programming, 92(3):453–480, 2002.

[15] H. Benson, A. Sen, D. F. Shanno, and R. V. D. Vanderbei. Interior-point algorithms, penalty methods and equilibrium problems. Computational Optimization and Applications, 34(2):155–182, 2006.

[16] D. Bertsimas and J. Sethuraman. Moment problems and semidefinite optimization. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, Handbook of Semidefinite Programming, pages 469–510. Kluwer Academic Publishers, 2000.

[17] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. SIAM Review, 53(3):464–501, 2011.

[18] Dimitris Bertsimas, Iain Dunning, and Miles Lubin. Reformulation versus cutting-planes for robust optimization. Computational Management Science, 13(2):195–217, 2016.

[19] Dimitris Bertsimas and Omid Nohadani. Robust optimization with simulated annealing. Journal of Global Optimization, 48(2):323–334, 2010.

[20] Dimitris Bertsimas, Omid Nohadani, and Kwong Meng Teo. Robust optimization for unconstrained simulation-based problems. Operations Research, 58(1):161–178, 2010.

[21] Dimitris Bertsimas, Melvyn Sim, and Meilin Zhang. A practically efficient approach for solving adaptive distributionally robust linear optimization problems. Management Science, 2018. To appear.

[22] Hans-Georg Beyer and Bernhard Sendhoff. Robust optimization – A comprehensive survey. Computer Methods in Applied Mechanics and Engineering, 196(33):3190–3218, 2007.

[23] Johannes J Bisschop and Robert Entriken. AIMMS: The modeling system. Paragon Decision Technology BV, 1993.

[24] Jonathan M. Borwein and Adrian S. Lewis. Convex Analysis and Nonlinear Optimization, Theory and Examples. Springer, 2000.

[25] Renato Bruni and Fabio Celani. A robust optimization approach for magnetic spacecraft attitude stabilization. Journal of Optimization Theory and Applications, 173(3):994–1012, nov 2016.

[26] Angelo Ciccazzo, Vittorio Latorre, Giampaolo Liuzzi, Stefano Lucidi, and Francesco Rinaldi. Derivative-free robust optimization for circuit design. Journal of Optimization Theory and Applications, 164(3):842–861, 2015.

[27] Andrew R. Conn and Luis N. Vicente. Bilevel derivative-free optimization and its application to robust optimization. Optimization Methods and Software, 27(3):561–577, 2012.

[28] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. Operations Research, 58:595–612, 2010.

[29] Moritz Diehl, Hans Georg Bock, and Ekaterina Kostina. An approximation technique for robust nonlinear optimization. Mathematical Programming, 107(1-2):213–230, 2006.

[30] Moritz Diehl, Johannes Gerhard, Wolfgang Marquardt, and Martin Mönnigmann. Numerical solution approaches for robust nonlinear optimal control problems. Computers & Chemical Engineering, 32(6):1279–1292, 2008.

[31] Iain Robert Dunning. Advances in robust and adaptive optimization: algorithms, software, and insights. PhD thesis, Massachusetts Institute of Technology, 2016.

[32] Peyman M. Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. Mathematical Programming, pages 1–52, 2017.

[33] Sabrina Fiege, Andrea Walther, Kshitij Kulshreshtha, and Andreas Griewank. Algorith-

mic differentiation for piecewise smooth functions: A case study for robust optimization. Optimization Methods and Software, pages 1–16, 2018. To appear.

[34] Matteo Fischetti and Michele Monaci. Cutting plane versus compact formulations for uncertain (integer) linear programs. Mathematical Programming Computation, 4(3):239–273, April 2012.

[35] R. Fletcher and S. Leyffer. Solving mathematical program with complementarity constraints as nonlinear programs. Optimization Methods and Software, 19(1):15–40, 2004.

[36] Roger Fletcher, Sven Leyffer, Danny Ralph, and Stefan Scholtes. Local convergence of SQP methods for mathematical programs with equilibrium constraints. SIAM Journal on Optimization, 17(1):259–286, 2006.

[37] A.V. DeMiguel M.P. Friedlander, F. Nogales, and S. Scholtes. A two-sided relaxation scheme for mathematical programs with equilibrium constraints. SIAM Journal on Optimization, 16(1):587–609, 2005.

[38] Virginie Gabrel, Cécile Murat, and Aurélie Thiele. Recent advances in robust optimization: An overview. European Journal of Operational Research, 235(3):471– 483, 2014.

[39] Marc Goerigk. ROPI - a robust optimization programming interface for C++. Optimization Methods and Software, 29(6):1261–1280, 2014.

[40] Joel Goh and Melvyn Sim. Distributionally robust optimization and its tractable approximations. Operations Research, 58(4):902–917, 2010.

[41] Joel Goh and Melvyn Sim. Robust optimization made easy with ROME. Operations Research, 59(4):973–985, 2011.

[42] D. Goldfarb and G. Iyengar. Robust convex quadratically constrained programs. Mathematical Programming, 97(3):495–515, 2003.

[43] D. Goldfarb and G. Iyengar. Robust portfolio selection problems. Mathematics of Operations Research, 28(1):1–38, 2003.

[44] Bram L Gorissen, İhsan Yanıkoğlu, and Dick den Hertog. A practical guide to robust optimization. Omega, 53:124–137, 2015.

[45] Warren Hare and Mason Macklem. Derivative-free optimization methods for finite minimax problems. Optimization Methods and Software, 28(2):300–312, 2013.

[46] Warren Hare and J. Nutini. A derivative-free approximate gradient sampling algorithm for finite minimax problems. Computational Optimization and Applications, 56(1):1–38, 2013.

[47] R. Hettich and K. O. Kortanek. Semi-infinite programming: Theory, methods, and applications. SIAM Review, 35(3):380–429, 1993.

[48] A Ilzhöfer, Boris Houska, and Moritz Diehl. Nonlinear MPC of kites under varying wind conditions for a new class of large-scale wind power generators. International Journal of Robust and Nonlinear Control, 17(17):1590–1599, 2007.

[49] Seung-Jean Kim and Stephen Boyd. A minimax theorem with applications to machine learning, signal processing, and finance. SIAM Journal on Optimization, 19(3):1344–1367, 2008.

[50] Oliver Lass and Stefan Ulbrich. Model order reduction techniques with a posteriori error control for nonlinear robust optimization governed by partial differential equations. SIAM Journal on Scientific Computing, 39(5):S112–S139, 2017.

[51] S. Leyffer. Mathematical programs with complementarity constraints. SIAG/OPT Views and News, 14(1):15–18, 2003.

[52] S. Leyffer. Complementarity constraints as nonlinear equations: Theory and numerical experience. In S. Dempe and V. Kalashnikov, editors, Optimization and Multivalued Mappings, pages 169–208. Springer, 2006.

[53] Sven Leyffer, Gabriel Lopez-Calva, and Jorge Nocedal. Interior methods for mathematical programs with complementarity constraints. SIAM Journal on Optimization, 17(1):52–77, 2006.

[54] Zukui Li and Marianthi Ierapetritou. Process scheduling under uncertainty: Review and challenges. Computers & Chemical Engineering, 32(4):715–727, 2008.

[55] X. Liu, G. Perakis, and J. Sun. A robust SQP method for mathematical programs with linear complementarity constraints. Computational Optimization and Applications, 34(1):5–33, 2005.

[56] Johan Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In Computer Aided Control Systems Design, 2004 IEEE International Symposium on, pages 284–289. IEEE, 2004.

[57] Johan Löfberg. Modeling and solving uncertain optimization problems in YALMIP. IFAC Proceedings Volumes, 41(2):1337–1341, 2008.

[58] Ahmadreza Marandi, Aharon Ben-Tal, Dick den Hertog, and Bertrand Melenberg. Extending the scope of robust quadratic optimization. Technical report, Optimization Online, June 2017.

[59] Matt Menickelly and Stefan M. Wild. Derivative-free robust optimization by outer approximations. Preprint ANL/MCS-P9004-1017, Argonne National Laboratory, Mathematics and Computer Science Division, 2017.

[60] John M. Mulvey, Robert J. Vanderbei, and Stavros A. Zenios. Robust optimization of large-scale systems. Operations Research, 43(2):264–281, 1995.

[61] Omid Nohadani and Kartikey Sharma. Optimization under decision-dependent uncertainty. Technical Report 1611.07992, ArXiv, 2016.

[62] D. Pal, G. N. Iyengar, and J. M. Cioffi. A new method of channel shortening with applications to discrete multi-tone (DMT) systems. In Proceedings of the IEEE International Conference on Communications, volume 2, pages 763–768. IEEE, 1998.

[63] Elijah Polak. Optimization. Springer New York, 1997.

[64] A. Raghunathan and L. T. Biegler. An interior point method for mathematical programs with complementarity constraints (MPCCs). SIAM Journal on Optimization, 15(3):720–750, 2005.

[65] A. Shapiro. Distributionally robust stochastic programming. SIAM Journal on Optimization, 27:2258–2275, 2017.

[66] O. Stein and G. Still. Solving semi-infinite optimization problems with interior point techniques. SIAM Journal on Control and Optimization, 42(3):769–788, 2003.

[67] Oliver Stein. Bi-Level Strategies in Semi-Infinite Programming, volume 71. Springer Science & Business Media, 2013.

[68] Oliver Stein and Paul Steuermann. The adaptive convexification algorithm for semi-infinite programming with arbitrary index sets. Mathematical Programming, 136(1):183–207, 2012.

[69] Samee ur Rehman and Matthijs Langelaar. Expected improvement based infill sampling for global robust optimization of constrained problems. Optimization and Engineering, 18(3):723–753, 2017.

[70] A. Ismael F. Vaz, Edite M. G. P. Fernandes, and M. Paula S. F. Gomes. SIPAMPL: semi-infinite programming with AMPL. ACM Transactions on Mathematical Software, 30(1):47–61, 2004.

12/31/2007