Uncertain Convex Programs: Randomized Solutions and Confidence Levels [*]

Giuseppe Calafiore[†] M.C. Campi[‡]

July 21, 2004

Abstract

Many engineering problems can be cast as optimization problems subject to convex constraints that are parameterized by an uncertainty or 'instance' parameter. A recently emerged successful paradigm for attacking these problems is robust optimization, where one seeks a solution which simultaneously satisfies all possible constraint instances. In practice, however, the robust approach is effective only for problem families with rather simple dependence on the instance parameter (such as affine or polynomial), and leads in general to conservative answers, since the solution is usually computed by transforming the original semi-infinite problem into a standard one, by means of relaxation techniques.

In this paper, we take an alternative 'randomized' or 'scenario' approach: by randomly sampling the uncertainty parameter, we substitute the original infinite constraint set with a finite set of N constraints.

We show that the resulting randomized solution fails to satisfy only a small portion of the original constraints, provided that a sufficient number of samples is drawn. Our key result is to provide an efficient explicit bound on the measure (probability or volume) of the original constraints that are possibly violated by the randomized solution. This volume rapidly decreases to zero as N is increased.

[*] This work is supported in part by the European Commission under the project HYBRIDGE IST-2001-32460, and by MIUR under the project "New methods for Identification and Adaptive Control for Industrial Systems"

[†] Dipartimento di Automatica e Informatica, Politecnico di Torino. Tel.: +39-011-564.7071; Fax: +39-011-564.7099. E-mail: calafiore@polito.it

[‡] Dipartimento di Automatica per l'Automozaione, Università di Brescia, via Branze 38, 25123 Brescia, Italy. E-mail: campi@ing.unibs.it. Internet: http://bsing.ing.unibs.it/~campi

1 Introduction

Uncertain convex programming [4, 13] deals with convex optimization problems in which the constraints are imprecisely known. In formal terms, an uncertain convex program (UCP) is a *family* of convex optimization problems whose constraints are parameterized by an uncertainty (or instance) parameter $\delta \in \mathbf{\Delta} \subset \mathbb{R}^\ell$:

$$\text{UCP} : \left\{ \min_{x \in \mathcal{X} \subseteq \ ^n} c^T x \quad \text{subject to } f(x,\delta) \le 0,\ \delta \in \mathbf{\Delta} \right\}, \tag{1}$$

where $x \in \mathcal{X}$ is the optimization variable, $\mathcal{X}$ is convex and closed, the function $f(x,\delta) : \mathcal{X} \times \mathbf{\Delta} \to \mathbb{R}^p$ is continuous and convex in x for all δ, and the inequality $f(x,\delta) \le 0$ is intended element-wise. In the above problem family the optimization objective is assumed to be linear and 'certain' without loss of generality.

A paradigm that emerged around 1997 for the solution of uncertain programs, and that is now the mainstream approach to the problem, is that of *robustness* [3, 4, 5, 12, 13]: in robust convex programming one searches for a solution which is feasible for *all* possible instances of the uncertain parameter δ, and hence for all problem instances belonging to the family UCP. This amounts to solving the following program:

$$\text{RCP:} \ \min_{x \in \ ^n} c^T x \ \text{ subject to } x \in \mathcal{X} \cap \Omega, \tag{2}$$

where

$$\Omega \doteq \bigcap_{\delta \in \mathbf{\Delta}} \{x : \ f(x,\delta) \le 0\} \tag{3}$$

(throughout, we assume that $\mathcal{X} \cap \Omega \neq \emptyset$).

Special cases of the above problem are robust linear programs [5], for which $f(x,\delta)$ is affine in x, and robust semidefinite programs [13], for which the set Ω is expressed as

$$\Omega \doteq \bigcap_{\delta \in \mathbf{\Delta}} \{x : \ F(x,\delta) \preceq 0\},$$

where $F(x,\delta) = F_0(\delta) + \sum_{i=1}^n x_i F_i(\delta)$, $F_i(\delta) = F_i^T(\delta)$, and '$\preceq$' means 'negative semidefinite'.

Robust convex programs arise in many contexts, and have found applications in, to mention but a few, truss topology design [3], robust antenna array design, portfolio optimization, and robust estimation and filtering, [13, 11]. In the context of systems and control engineering, robust semidefinite programs proved to be useful in constructing Lyapunov functions for uncertain systems, and in the design of robust controllers, see e.g. [1].

The RCP problem is still a convex optimization problem, but since it involves an infinite number of constraints, it is in general numerically hard to solve, [4]. For this reason, in all the previously cited literature particular *relaxations* of the original problem are sought in order to transform the original semi-infinite optimization problem into a standard convex optimization one. Typical relaxation methods require the introduction of additional 'multiplier' or 'scaling' variables, over which the optimization is to be performed. The projection of the feasible set of the relaxed problem onto the space of original problem variables is in general an 'inner' approximation of the original feasible set, and therefore relaxation techniques provide an upper bound on the actual optimal solution of RCP. The main difficulties with the relaxation approach are that the sharpness of the approximation is in general unknown (except for particular classes of problems, see [6, 14]), and that the method itself can be applied only when the dependence of f on δ has a particular and simple functional form, such as affine, polynomial or rational. As an additional remark, we note that the standard convex optimization problem achieved through relaxation often belongs to a more complex class of optimization problems than the original one, that is relaxation lifts the problem class. For example, robust linear programs may result in second order cone programs (see for instance [18]), and robust second order cone programs may result in semidefinite programs ([22, 21]).

In this paper, we pursue a different philosophy of solution, which is based on randomization of the parameter δ. The key idea is to assume that the uncertain problem family (1) is parameterized by an instance parameter δ which is a random variable. Then, by drawing

N random samples of the instance parameter, we substitute the infinite constraint set of RCP with a finite set of N constraints, and look for an optimal solution which is feasible with respect to this sampled constraint set. Solving this randomized or 'sampled' counterpart of RCP amounts to solving a standard convex program with N constraints. The feasible set of the randomized problem is an outer approximation of the feasible set of RCP, as opposed to the inner approximation obtained via the relaxation approach. Therefore, the randomized approach yields a solution that outperforms the optimal objective value of RCP. The price which is paid is that the randomized solution is feasible for many – but not all – the instances of δ.

This 'constraint sampling' approach is not new, as it is equivalent to the 'scenario' approach used in stochastic programming, see for instance [20]. In this setting, the crucial question to which this paper is devoted (and which is completely open, to the best of the authors knowledge) is the following

> How many samples (scenarios) need to be drawn in order to guarantee that the resulting randomized solution violates only a 'small portion' of the constraints?

Using statistical learning techniques, we provide an explicit bound on the measure (probability or volume) of the set of original constraints that are possibly violated by the randomized solution. This volume rapidly decreases to zero as N is increased, and therefore the obtained randomized solution can be made *approximately feasible* for the robust problem by sampling a sufficient number of constraints. A further advantage of the method is that the original problem is not lifted to a more complex class since, for instance, the randomized version of a robust linear program is still a linear program.

Constraint reduction methods have been proposed by other researchers in different contexts. Approximate linear programs for queuing networks with a reduced number of constraints have been studied in [19]. Dynamic programming is considered in [15] where an approximated cost-to-go function is introduced to implement a linear program-based solution with a low number of constraints. These mentioned contributions propose ad-hoc constraint reduction methods that exploit the specific structure of the problem at hand.

The literature on probabilistic methods for general convex optimization problems is very scarce. Independently of the present work, a general constraint sample complexity evaluation for linear programs has been derived in [10], motivated by applications in dynamic programming and adaptive control. Admittedly, the sample complexity bounds derived in [10] are loose, since they are directly based on the Vapnik-Chervonenkis uniform convergence theory, and hence suffer from the conservatism of this theory. In the different – though strictly related – setting of feasibility determination, the idea of approximate feasibility has been discussed in [2], where a method for reducing the problem of finding an approximately feasible x point is reduced to a convex optimization problem. For another contribution in a similar direction, see [9]. This paper presents a general theory for robust convex programming in a probabilistic setting. We show that a generic uncertain convex program can be solved by resorting to random sampling, and that a rapidly decreasing bound for the sample size exists, which credits the method with wide applicability.

This paper is organized as follows. Section 2 contains the main result (Theorem 1), whose complete proof is reported in a separate section (Section 3.2). In Section 4 the main result is extended to problems with non-unique optimal solutions (Theorem 3) and to problems with convex objective. Section 5 presents numerical examples and applications to robust linear programming, robust least-squares problems, and semidefinite programming. Conclusions are finally drawn in Section 6.

2 Randomized Approach to Uncertain Convex Programming

Consider (1), and assume that the support $\mathbf{\Delta}$ for δ is endowed with a σ-algebra $\mathcal{D}$ and that a probability measure P over $\mathcal{D}$ is also assigned. Depending on the situation at hand, P can have different interpretations. Sometimes, it is the actual probability that the uncertainty parameter δ takes on value in a certain set. Other times, P simply describes the relative importance we attribute to different instances.

Definition 1 (Violation probability) *Let $x \in \mathcal{X}$ be a candidate solution, then the* probability of violation *of x is defined as*

$$V(x) \doteq P\{\delta \in \Delta : \ f(x,\delta) > 0\}$$

(here, it is assumed that $\{\delta \in \Delta : \ f(x,\delta) > 0\}$ is an element of the σ-algebra $\mathcal{D}$). ⋆

For example, if a uniform (with respect to Lebesgue measure) probability density is assumed, then $V(x)$ measures the volume of 'bad' parameters δ such that the constraint $f(x,\delta) \leq 0$ is violated. Clearly, a solution x with small associated $V(x)$ is feasible for 'most' of the problem instances in the UCP family. We have the following definition.

Definition 2 (ϵ-level solution) *Let $\epsilon \in [0,1]$. We say that $x \in \mathcal{X}$ is an ϵ-level robustly feasible solution if $V(x) \leq \epsilon$.* ⋆

Our goal is to devise an algorithm that returns a ϵ-level solution, where ϵ is any fixed small level. To this purpose, we now introduce the randomized counterpart of the robust problem (2).

Assume that N independent identically distributed samples $\delta^{(1)}, \ldots, \delta^{(N)}$ are drawn according to probability P and consider the convex optimization problem

$$\begin{aligned} \mathrm{RCP}_N : \quad \min_{x \in {}^{n}} c^T x \quad \text{subject to} \quad & x \in \mathcal{X} \\ & f(x, \delta^{(i)}) \leq 0, \ i = 1, \ldots, N. \end{aligned} \tag{4}$$

For the time being, we assume that RCP_N admits a unique solution. Clearly, should RCP_N be unfeasible (i.e. $\cap_{i=1,\ldots,N} \left\{x : f(x,\delta^{(i)}) \leq 0\right\} \cap \mathcal{X} = \emptyset$), then RCP would be unfeasible too. The uniqueness assumption is instead temporarily made for clarity in the presentation and proof of the main result, and it is removed in the later Section 4.1.

Let then $\hat{x}_N$ be the unique solution of problem RCP_N. Since the constraints $f(x,\delta^{(i)}) \leq 0$ are randomly selected, $\hat{x}_N$ is a random variable. The following key theorem pinpoints the properties of $\hat{x}_N$.

Theorem 1 *Fix two real numbers $\epsilon \in [0,1]$ (level parameter) and $\beta \in [0,1]$ (confidence parameter) and let*

$$N \geq \frac{n}{\epsilon\beta} - 1 \tag{5}$$

(remember that n = size of x). Then, with probability not smaller than $1-\beta$, the randomized optimization problem RCP_N returns an optimal solution $\hat{x}_N$ which is ϵ-level robustly feasible. $\star$

In the theorem, probability $1-\beta$ refers to the probability P^N $(= P \times \cdots \times P$, n times) of extracting a 'bad' multisample, i.e. a multisample $\delta^{(1)}, \ldots \delta^{(N)}$ such that $\hat{x}_N$ does not meet the ϵ-level feasibility property. A subtle measurability issue arises regarding the definition of this probability. In fact, without any extra assumptions, there is no guarantee that the set of multisamples such that $V(\hat{x}_N) \leq \epsilon$ is measurable, so that its probability may not be well-defined. Here and elsewhere, the measurability of this set is taken as an assumption.

The proof Theorem 1, which requires the statement of some preliminary results, is given in Section 3.2. We here remark that the 'sample complexity' of the algorithm (i.e. the number N of random samples that need to be drawn in order to achieve the desired probabilistic level in the solution) scales linearly with respect to $1/\epsilon\beta$, and with respect to the number n of decision variables. The original semi-infinite problem is therefore replaced by a standard convex problem with many constraints. For reasonable probabilistic levels, the required number of these constraints appears to be manageable by current convex optimization numerical solvers.

Remark 1 (Role of probability P) Probability P plays a double role in our approach: on the one hand, it is the probability according to which the uncertainty is sampled; on the other hand, it is the probabilistic measure according to which the probabilistic levels of quality mentioned in the above theorem are assessed.

In certain problems, P is the probability of occurrence of the different instances of the uncertain parameter δ. In other cases, it more simply represents the different importance we place on different instances. Extracting δ samples according to a given probability measure P is not always a simple task to accomplish, see [8] for a discussion of this topic and polynomial-time algorithms for the sample generation in some matrix norm-bounded sets.

In some applications (see e.g. [7]), probability P is not explicitly known and the sampled constraints are directly made available as observations. In this connection, it is important to note that the bound (5) is probability independent (i.e. it holds irrespective of the underlying probability P) and can therefore be applied even when P is unknown. □

Remark 2 (Feasibility vs. performance) Solution methodologies for the RCP problem are known only for certain simple dependencies of f on δ, such as affine, polynomial or rational. In other cases, the randomized approach offers a practicable way of proceeding in order to compute a solution.

Even when solving the RCP problem is possible, the randomized approach can offer advantages that should be considered when choosing a solution methodology. In fact, solving RPC gives 100% deterministic guarantee that the constraints are satisfied, no matter what $\delta \in \mathbf{\Delta}$ is. Solving RCP$_N$ leaves instead a chance to the occurrence of δ's which are violated by the solution. On the other hand, RCP$_N$ provides a solution (for the satisfied constraints) that outperforms the solution obtained via RCP. In this context, fixing a suitable level ϵ is sometimes a matter of trading probability of unfeasibility against performance. □

Remark 3 (A-priori and a-posteriori assessments) It is worth noticing that a distinction should be made between the a-priori and a-posteriori assessments that one can make regarding the probability of constraint violation. Indeed, *before* running the optimization, it is guaranteed by Theorem 1 that if $N \geq n/\epsilon\beta - 1$ samples are drawn, the solution of the randomized program will be ϵ-level robustly feasible, with probability not smaller than $1-\beta$. However, the a-priori parameters ϵ, β are generally chosen not too small, due to technological limitations on the number of constraints that one specific optimization software can deal with.

On the other hand, once a solution has been computed (and hence $x = \hat{x}_N$ is fixed), one can make an a-posteriori assessment of the level of feasibility using Monte-Carlo techniques. In this case, a new batch of $\tilde{N}$ independent random samples of $\delta \in \Delta$ is generated, and the *empirical probability* of constraint violation, say $\hat{V}_{\tilde{N}}(\hat{x}_N)$, is computed according to the formula $\hat{V}_{\tilde{N}}(\hat{x}_N) = \frac{1}{\tilde{N}} \sum_{i=1}^{\tilde{N}} 1(f(\hat{x}_N, \delta^{(i)})) \leq 0)$, where $1(\cdot)$ is the indicator function. Then, the classical Hoeffding's inequality, [16], guarantees that

$$|\hat{V}_{\tilde{N}}(\hat{x}_N) - V(\hat{x}_N)| \leq \tilde{\epsilon}$$

holds with confidence greater than $1 - \tilde{\beta}$, provided that

$$\tilde{N} \geq \frac{\log 2/\tilde{\beta}}{2\tilde{\epsilon}^2} \tag{6}$$

test samples are drawn. This latter a-posteriori test can be easily performed using a large sample size $\tilde{N}$ because no optimization problem is involved in such an evaluation. □

3 Technical preliminaries and proof of Theorem 1

This section is technical and contains the machinery needed for the proof of Theorem 1. The reader not interested in the details may skip this section and pass to the numerical examples section.

3.1 Preliminaries

We start with a a technical lemma.

Lemma 1 *Given a set S of $p+2$ points in $\mathbb{R}^p$, there exist two points among these, say ξ_1, ξ_2, such that the line segment $\overline{\xi_1 \xi_2}$ intersects the hyperplane (or one of the hyperplanes if indetermination occurs) generated by the remaining p points $\xi_3, \ldots, \xi_{p+2}$.* ⋆

Proof. Choose any set S' composed of $p-1$ points from S, and consider the bundle of hyperplanes passing through S'. If this bundle has more than one degree of freedom, augment S' with additional arbitrary points, until the bundle has exactly one degree of freedom. Consider now the translation which brings one point of S' to coincide with the origin, and let S'' be the translated point set. The points in S'' lie now in a subspace $\mathcal{F}$ of dimension $p-2$, and all the hyperplanes of the (translated) bundle are of the form $v^T x = 0$, where $v \in \mathcal{V}$, being $\mathcal{V}$ the subspace orthogonal to $\mathcal{F}$, which has dimension 2.

Call $x_4, \ldots, x_{p+2}$ the points belonging to S'', and x_1, x_2, x_3 the remaining points. Consider three fixed hyperplanes H_1, H_2, H_3 belonging to the bundle generated by S'', which pass through x_1, x_2, x_3, respectively; these hyperplanes have equations $v_i^T x = 0$, $i = 1, 2, 3$. Since $\dim \mathcal{F} = 2$, one of the v_i's (say v_3) must be a linear combination of the other two, i.e. $v_3 = \alpha_1 v_1 + \alpha_2 v_2$.

Suppose that one of the hyperplanes, say H_1, leaves the points x_2, x_3 on the same open half-space $v_1^T x > 0$ (note that assuming $v_1^T x > 0$, as opposed to $v_1^T x < 0$ is a matter of choice since the sign of v_1 can be arbitrarily selected). Suppose that also another hyperplane, say H_2, leaves the points x_1, x_3 on the same open half-space $v_2^T x > 0$. Then, it follows that $v_1^T x_3 > 0$, and $v_2^T x_3 > 0$. Since $v_3^T x_3 = 0$, it follows also that $\alpha_1 \alpha_2 < 0$. We now have that

$$\begin{aligned} v_3^T x_1 &= (\alpha_1 v_1 + \alpha_2 v_2)^T x_1 = \alpha_2 v_2^T x_1 \\ v_3^T x_2 &= (\alpha_1 v_1 + \alpha_2 v_2)^T x_2 = \alpha_1 v_1^T x_2, \end{aligned}$$

where the first term has the same sign as α_2, and the second has the same sign as α_1. Thus, $v_3^T x_1$ and $v_3^T x_2$ do not have the same sign. From this reasoning it follows that not all the three hyperplanes can leave the complementary two points on the same open half-space, and the result is proved. □

We now come to a key instrumental result. Consider the convex optimization program

$$\mathcal{P} : \min_{x \in \ {}^n} c^T x \ \text{ subject to } \ x \in \mathcal{X}_i, \ i = 1, \ldots, m,$$

where $\mathcal{X}_i$, $i = 1, \ldots, m$ are closed convex sets. Let the convex programs $\mathcal{P}_k$, $k = 1, \ldots, m$, be obtained from $\mathcal{P}$ by removing the k-th constraint:

$$\mathcal{P}_k : \min_{x \in {}^n} c^T x \text{ subject to } x \in \mathcal{X}_i,\ i = 1, \ldots, k-1, k+1, \ldots, m.$$

Let x^* be any optimal solution of $\mathcal{P}$ (assuming it exists), and let x^*_k be any optimal solution of $\mathcal{P}_k$ (again, assuming it exists). We have the following definition.

Definition 3 (Support constraints) *The k-th constraint $\mathcal{X}_k$ is a* support *constraint for $\mathcal{P}$ if problem $\mathcal{P}_k$ has an optimal solution x^*_k such that $c^T x^*_k < c^T x^*$.* $\star$

The following theorem holds.

Theorem 2 *The number of support constraints for problem $\mathcal{P}$ is at most n.* $\star$

Proof. We prove the statement by contradiction. Suppose then that problem $\mathcal{P}$ has $n_s > n$ support constraints and choose any $(n+1)$-tuple of constraints among these.

Then, there exist $n+1$ points (say, without loss of generality, the first $n+1$ points) x^*_k, $k = 1, \ldots, n+1$, which are optimal solutions for problems $\mathcal{P}_k$, and which lie all in the same open half-space $\{x : c^T x < c^T x^*\}$. We show next that, if this is the case, then x^* is not optimal for $\mathcal{P}$, which constitutes a contradiction.

Consider the line segments connecting x^* with each of the x^*_k, $k = 1, \ldots, n+1$, and consider a hyperplane $\mathcal{H} \doteq \{c^T x = \alpha\}$ with $\alpha < c^T x^*$, such that $\mathcal{H}$ intersects all the line segments. Let $\bar{x}^*_k$ denote the point of intersection between $\mathcal{H}$ and the segment $\overline{x^* x^*_k}$. Notice that, by convexity, the point $\bar{x}^*_k$ certainly satisfies the constraints $\mathcal{X}_1, \ldots, \mathcal{X}_{k-1}, \mathcal{X}_{k+1}, \ldots, \mathcal{X}_{n+1}$, but it does not necessarily satisfy the constraint $\mathcal{X}_k$.

Suppose first that there exists an index k such that $\bar{x}^*_k$ belongs to the convex hull $\mathrm{co}\{\bar{x}^*_1, \ldots, \bar{x}^*_{k-1}, \bar{x}^*_{k+1}, \ldots, \bar{x}^*_{n+1}\}$. Then, since $\bar{x}^*_1, \ldots, \bar{x}^*_{k-1}, \bar{x}^*_{k+1}, \ldots, \bar{x}^*_{n+1}$ all satisfy the k-th constraint, so do all points in $\mathrm{co}\{\bar{x}^*_1, \ldots, \bar{x}^*_{k-1}, \bar{x}^*_{k+1}, \ldots, \bar{x}^*_{n+1}\}$ and hence $\bar{x}^*_k \in \mathrm{co}\{\bar{x}^*_1, \ldots, \bar{x}^*_{k-1}, \bar{x}^*_{k+1}, \ldots, \bar{x}^*_{n+1}\}$ satisfies the k-th constraint. On the other hand, as it has been mentioned above, $\bar{x}^*_k$ satisfies all other constraints $\mathcal{X}_1, \ldots, \mathcal{X}_{k-1}, \mathcal{X}_{k+1}, \ldots, \mathcal{X}_{n+1}$, and therefore

$\bar{x}_k^*$ satisfies *all* constraints. From this it follows that $\bar{x}_k^*$ is a feasible solution for problem $\mathcal{P}$, and has an objective value $c^T x_k^* = \alpha < c^T x^*$, showing that x^* is not optimal for $\mathcal{P}$. Since this is a contradiction, we are done.

Consider now the complementary case in which there does not exist a $\bar{x}_k^* \in \mathrm{co}\{\bar{x}_1^*, \ldots, \bar{x}_{k-1}^*, \bar{x}_{k+1}^*, \ldots, \bar{x}_{n+1}^*\}$. Then, we can always find two points, say $\bar{x}_1^*, \bar{x}_2^*$, such that the line segment $\overline{\bar{x}_1^* \bar{x}_2^*}$ intersects at least one hyperplane passing through the remaining $n-1$ points $\bar{x}_3^*, \ldots, \bar{x}_{n+1}^*$. Such couple of points always exist by virtue of Lemma 1. Denote with $\bar{x}_{1,2}^*$ the point of intersection (or any point in the intersection, in case more than one exists). Notice that $\bar{x}_{1,2}^*$ certainly satisfies all constraints, except possibly the first and the second. Now, $\bar{x}_{1,2}^*, \bar{x}_3^*, \ldots, \bar{x}_{n+1}^*$ are n points in a flat of dimension $n-2$. Again, if one of these points belongs to the convex hull of the others, then this point satisfies all constraints, and we are done. Otherwise, we repeat the process, and determine a set of $n-1$ points in a flat of dimension $n-3$.

Proceeding this way repeatedly, either we stop the process at a certain step (and then we are done), or we proceed all way down until we determine a set of three points in a flat of dimension one. In this latter case we are done all the same, since out of three points in a flat of dimension one there is always one which lies in the convex hull of the other two.

Thus, in any case we have a contradiction and this proves that $\mathcal{P}$ cannot have $n+1$ or more support constraints. □

We are now ready to present a proof for Theorem 1.

3.2 Proof of Theorem 1

Consider $N+1$ independent random variables $z^{(1)}, \ldots, z^{(N+1)}$ taking value in $\boldsymbol{\Delta}$ with probability P and consider the following $N+1$ instances of RCP$_N$:

$$\mathrm{RCP}_N^k : \min_{x \in \ ^n} c^T x \quad \text{subject to} \quad \begin{array}{l} x \in \mathcal{X} \\ f(x, z^{(i)}) \leq 0, \ i = 1, \ldots, k-1, k+1, \ldots, N+1. \end{array}$$

For $k = 1, \ldots, N+1$, let $\hat{x}_N^k$ be the optimal solution of problem RCP_N^k, and notice that $\hat{x}_N^k$ is such that $f(\hat{x}_N^k, z^{(i)}) \leq 0$, for $i = 1, \ldots, k-1, k+1, \ldots, N+1$, but it does not necessarily hold that $f(\hat{x}_N^k, z^{(k)}) \leq 0$.

The idea of the proof is as follows: first we notice that $V(\hat{x}_N)$ is a random variable belonging to the interval $[0, 1]$. Then, we show that the expected value of $V(\hat{x}_N)$ is close to 0, and from this we infer a lower bound on the probability of having $V(\hat{x}_N)$ smaller than ϵ. Define

$$\bar{V}_N \doteq E_{P^N}[V(\hat{x}_N)], \tag{7}$$

where E is the expectation operator, and, for $k = 1, \ldots, N+1$, let

$$v_k \doteq \begin{cases} 1, & \text{if } f(\hat{x}_N^k, z^{(k)}) > 0 \\ 0, & \text{otherwise,} \end{cases}$$

i.e. the random variable v_k is equal to one if $\hat{x}_N^k$ fails to satisfy the constraint $f(\hat{x}_N^k, z^{(k)}) \leq 0$, and it is zero otherwise. Let also

$$\hat{\bar{V}}_N \doteq \frac{1}{N+1} \sum_{k=1}^{N+1} v_k. \tag{8}$$

We have that

$$\begin{aligned} E_{P^{N+1}}[v_k] &= E_{P^N}\left[E_P[v_k | z^{(1)}, \ldots, z^{(k-1)}, z^{(k+1)}, \ldots, z^{(N+1)}]\right] \\ &= E_{P^N}\left[P\{z^{(k)} \in \mathbf{\Delta} : f(\hat{x}_N^k, z^{(k)}) > 0\}\right] \\ &= E_{P^N}[V(\hat{x}_N^k)] \\ &= \bar{V}_N, \end{aligned}$$

which yields

$$E_{P^{N+1}}[\hat{\bar{V}}_N] = \bar{V}_N. \tag{9}$$

The key point is now to determine an upper bound for $E_{P^{N+1}}[\hat{\bar{V}}_N]$.

To this purpose, we proceed as follows. Fix a realization $\bar{z}^{(1)}, \ldots, \bar{z}^{(N+1)}$ of variables $z^{(1)}, \ldots, z^{(N+1)}$. We show that, for any choice of $\bar{z}^{(1)}, \ldots, \bar{z}^{(N+1)}$ it holds that

$$\hat{\bar{V}}_N(\bar{z}^{(1)}, \ldots, \bar{z}^{(N+1)}) \leq \frac{n}{N+1}. \tag{10}$$

Thus, by taking expectation we still have

$$E_{P^{N+1}}[\hat{\hat{V}}_N] \leq \frac{n}{N+1}. \tag{11}$$

To show (10), consider the convex problem involving all the $N+1$ constraints

$$\begin{aligned} \mathrm{RCP}_{N+1} : \min_{x \in {}^{n}} c^T x \quad \text{subject to} \quad & x \in \mathcal{X} \\ & f(x, \bar{z}^{(i)}) \leq 0, \ i = 1, \ldots, N+1. \end{aligned}$$

and let $\hat{x}_{N+1}$ be the corresponding optimal solution. Also consider the optimal solutions $\hat{x}_N^k$, $k = 1, \ldots, N+1$, of programs RCP_N^k, $k = 1, \ldots, N+1$, obtained by removing one by one the constraints $f(x, \bar{z}^{(i)}) \leq 0$. Now, from Theorem 2 we know that at most n of the constraints when removed from RCP_{N+1} will change the optimal solution and improve the objective. From this it follows that there exist at most n optimal points $\hat{x}_N^k$ such that the constraint $f(\hat{x}_N^k, z^{(k)}) \leq 0$ is violated. Hence, at most n of the v_k's can be equal to one, and from (8) equation (10) follows.

Now, given $\epsilon > 0$, we can bound the expectation $E_{P^N}[V(\hat{x}_N)]$ from below as

$$E_{P^N}[V(\hat{x}_N)] \geq \epsilon P^N\{V(\hat{x}_N) > \epsilon\}. \tag{12}$$

Letting $\bar{\beta} \doteq P^N\{V(\hat{x}_N) > \epsilon\}$, combining equations (7), (9), (11), and (12), we finally obtain

$$\epsilon\bar{\beta} \leq \frac{n}{N+1},$$

from which the statement of the theorem easily follows. □

4 Extensions

4.1 Problems with multiple optimal solutions

In this section we drop the assumption that the optimal solution of RCP_N is unique.

Consider problem RCP_N (4). If more than one optimal solution exists for this problem, we assume that a solution selection procedure (tie-break rule) is applied in order to single out a specific optimal solution $\hat{x}_N$. The selection rule goes as follows.

Rule 1 *Let $t_i(x)$, $i = 1, \dots, p$, be given convex and continuous functions. Among the optimal solutions for RCP$_N$, select the one that minimizes $t_1(x)$. If indetermination still occurs, select among the x that minimize $t_1(x)$ the solution that minimizes $t_2(x)$, and so on with $t_3(x), t_4(x), \dots$. We assume that functions $t_i(x)$, $i = 1, \dots, p$, are selected so that the tie is broken within p steps at most. As a simple example of a tie-break rule, one can consider $t_1(x) = x_1, t_2(x) = x_2, \dots$.* ⋆

From now on, for any convex optimization problem considered, by optimal solution we mean either the unique optimal solution, or the solution selected according to Rule 1, in case the problem admits more than one optimal solution. The following theorem extends Theorem 1 to the present setting.

Theorem 3 *The result in Theorem 1 holds also in case when RCP$_N$ has multiple optimal solutions, provided that $\hat{x}_N$ is selected according to Rule 1.* ⋆

Proof. The proof follows the same line as the one for Theorem 1 except that Definition 3 and Theorem 2 need suitable amendments. Precisely, we now have:

Definition 4 (Support constraints) *The k-th constraint $\mathcal{X}_k$ is a* support *constraint for $\mathcal{P}$ if problem $\mathcal{P}_k$ has an optimal solution x_k^* such that $x_k^* \neq x^*$.* ⋆

Definition 4 is a generalization of Definition 3 since, in the case of single optimal solutions, $x_k^* \neq x^*$ is equivalent to $c^T x_k^* < c^T x^*$.

The statement of Theorem 2 remains unaltered with the above definition of support constraint (this needs a proof - see below) and then all other parts of the proof of Theorem 1 goes through to prove Theorem 3. Hereafter, we sketch a proof of Theorem 2 in the present context.

As in the original proof of Theorem 2, suppose that there are $n+1$ support constraints and let $x_k^*, k = 1, \dots, n+1$, be the optimal solutions for the corresponding $\mathcal{P}_k$ problems. We show that $x^* \notin \text{co}\{x_1^*, \dots, x_{n+1}^*\}$, and therefore a $(n-1)$-dimensional hyperplane separating

x^* from $x_1^*, \dots, x_{n+1}^*$ can be constructed (this part is new and the separating hyperplane replaces $\mathcal{H}$ in the original proof).

Suppose, for the purpose of contradiction, that $x^* \in \mathrm{co}\{x_1^*, \dots, x_{n+1}^*\}$, and hence x^* can be written as $x^* = \sum_{i \in I \subset \{1,\dots,n+1\}} \alpha_i x_i^*$, $0 < \alpha_i \leq 1$, $\sum_{i \in I} \alpha_i = 1$. Note that $c^T x_i^* \leq c^T x^*$, $\forall i \in I$. If $c^T x_i^* < c^T x^*$, for some $i \in I$, we then have: $c^T x^* = c^T \sum_{i \in I} \alpha_i x_i^* = \sum_{i \in I} \alpha_i c^T x_i^* < c^T x^*$, which is impossible, and therefore $c^T x_i^* = c^T x^*$, $\forall i \in I$. In turn, $t_1(x_i^*) \leq t_1(x^*)$, $\forall i \in I$. If $t_1(x_i^*) < t_1(x^*)$, for some $i \in I$, we then have: $t_1(x^*) = t_1(\sum_{i \in I} \alpha_i x_i^*) \leq \sum_{i \in I} \alpha_i t_1(x_i^*) < t_1(x^*)$, which is again impossible, and therefore $t_1(x_i^*) = t_1(x^*)$, $\forall i \in I$. Proceeding in a similar way for $t_2(x), \dots, t_p(x)$, we conclude that, for any i: $c^T x_i^* = c^T x^*, t_1(x_i^*) = t_1(x^*), \dots, t_p(x_i^*) = t_p(x^*)$, but this is impossible since then $t_1(x), \dots, t_p(x)$ would not give a tie-break rule. Thus, we have a contradiction and $x^* \notin \mathrm{co}\{x_1^*, \dots, x_{n+1}^*\}$.

Consider now a $(n-1)$-dimensional hyperplane $\mathcal{H}$ separating x^* from $x_1^*, \dots, x_{n+1}^*$ (and not touching x^*) and construct $\bar{x}_1^*, \dots, \bar{x}_{n+1}^*$ similarly to the original proof of Theorem 2. In the original proof of Theorem 2, we have proven that a point, say $\bar{x}^*$, exists in $\mathcal{H}$ that satisfies all constraints. A bit of inspection of that proof reveals that $\bar{x}^*$ is in fact in the convex hull of $\bar{x}_1^*, \dots, \bar{x}_{n+1}^*$: $\bar{x}^* \in \mathrm{co}\{\bar{x}_1^*, \dots, \bar{x}_{n+1}^*\}$. We conclude the proof by showing that such $\bar{x}^*$ would outperform x^* in the $\mathcal{P}$ problem so that x^* would not be the optimal solution of $\mathcal{P}$. Since this is a contradiction, we then have that no $n+1$ support constraints can exist.

Let $\bar{x}^* = \sum_{j \in J \subset \{1,\dots,n+1\}} \beta_j \bar{x}_j^*$, $0 < \beta_j \leq 1$, $\sum_{j \in J} \beta_j = 1$. Begin by observing that $c^T \bar{x}_j^* \leq c^T x^*$, $\forall j \in J$. Indeed, $\bar{x}_j^* = \alpha x_j^* + (1-\alpha) x^*$ with $0 < \alpha \leq 1$, so that $c^T \bar{x}_j^* = c^T(\alpha x_j^* + (1-\alpha)x^*) = \alpha c^T x_j^* + (1-\alpha) c^T x^* \leq c^T x^*$. If $c^T \bar{x}_j^* < c^T x^*$ for some $j \in J$, we then have: $c^T \bar{x}^* = c^T \sum_{j \in J} \beta_j \bar{x}_j^* = \sum_{j \in J} \beta_j c^T \bar{x}_j^* < c^T x^*$ and $\bar{x}^*$ outperforms x^*. If $c^T \bar{x}_j^* = c^T x^*$, $\forall j \in J$, one proceeds to consider $t_1(x), t_2(x), \dots$. Following a similar rationale, one then concludes that $\bar{x}^*$ outperforms x^* at some step for, otherwise, the tie between x^* and the x_j^*'s would not be broken by $t_1(x), \dots, t_p(x)$. This concludes the proof. □

4.2 Problems with no solution

Notice that even if problem RCP attains an optimal solution, a further technical difficulty may arise when a randomized problem instance RCP$_N$ has no solution. This may happen when the set $\cap_{i=1,\ldots,N}\left\{x : f(x,\delta^{(i)}) \leq 0\right\} \cap \mathcal{X}$ is unbounded in such a way that the optimal solution 'escapes' to infinity, while the original problem is constrained to a set $\cap_{\delta\in\boldsymbol{\Delta}}\{x : f(x,\delta) \leq 0\} \cap \mathcal{X}$ such that the optimal solution is attained. In this case, Theorem 3 still holds with a little modification, as explained below.

Suppose that a random extraction of a multisample $\delta^{(1)}, \ldots \delta^{(N)}$ is rejected when no optimal solution exists, and another extraction is performed in this case. Then, the probability of ending up with a multisample such that $V(\hat{x}_N) \leq \epsilon$ is not smaller than $1-\beta$. In formal terms, this probability is a conditional probability to the event where a solution exists.

Theorem 4 *Let $\boldsymbol{\Delta}_E^N \subseteq \boldsymbol{\Delta}^N$ be the set where a solution of RCP$_N$ exists. If $P^N(\boldsymbol{\Delta}_E^N) > 0$, the result in Theorem 3 holds, provided that $1-\beta$ is intended as a lower bound on the conditional probability $P^N(\{V(\hat{x}_N) \leq \epsilon\} \cap \boldsymbol{\Delta}_E^N)/P^N(\boldsymbol{\Delta}_E^N)$. (the measurability of $\boldsymbol{\Delta}_E^N$ is taken as an assumption).* $\star$

Proof. We sketch here how the proof of Theorem 3 can be amended to cope with the present setting. Let $\boldsymbol{\Delta}_E^{N+1} \subseteq \boldsymbol{\Delta}^{N+1}$ be the set where a solution of the problem with $N+1$ constraints exists, and note that $\boldsymbol{\Delta}_E^N \times \boldsymbol{\Delta} \subseteq \boldsymbol{\Delta}_E^{N+1}$ for, if N constraints avoid escape to infinity of the solution, this is still true after adding one more constraint. Next, with the symbols having the same meaning as in the proof of Theorem 1, let

$$v'_k \doteq \begin{cases} 1, & \text{if } f(\hat{x}_N^k, z^{(k)}) > 0 \quad \text{or} \quad \hat{x}_N^k \text{ does not exist} \\ 0, & \text{otherwise,} \end{cases}$$

and let $v_k \doteq v'_k \cdot 1(\boldsymbol{\Delta}_E^{N+1})$, $1(\cdot)$ being the indicator function. It is then not difficult to adapt the proof of Theorem 1 to conclude that

$$\frac{n}{N+1}P^{N+1}(\boldsymbol{\Delta}_E^{N+1}) \geq E_{P^{N+1}}\left[\frac{1}{N+1}\sum_{k=1}^{N+1} v_k\right] = E_{P^{N+1}}[v_{N+1}] = P^{N+1}(\boldsymbol{\Delta}_E^{N+1} \cap (A \cup B)),$$

with $A \doteq \{f(\hat{x}_N^k, z^{(k)}) > 0\}$, $B \doteq \{\hat{x}_N^k$ does not exist$\}$. Since $\mathbf{\Delta}_E^{N+1} \cap (A \cup B) = ((\mathbf{\Delta}_E^N \times \mathbf{\Delta}) \cap A) \;\cup\; (\mathbf{\Delta}_E^{N+1} - (\mathbf{\Delta}_E^N \times \mathbf{\Delta}))$, we then have

$$\frac{n}{N+1} P^{N+1}(\mathbf{\Delta}_E^{N+1}) \geq P^{N+1}((\mathbf{\Delta}_E^N \times \mathbf{\Delta}) \cap A) + P^{N+1}(\mathbf{\Delta}_E^{N+1} - (\mathbf{\Delta}_E^N \times \mathbf{\Delta})). \tag{13}$$

Finally, with the notation $\bar{\beta} \doteq P^N(\{V(\hat{x}_N) > \epsilon\} \cap \mathbf{\Delta}_E^N)$, we have:

$$\begin{aligned}
\epsilon\bar{\beta} &\leq E_{P^N}[\{V(\hat{x}_N) > \epsilon\} \cap \mathbf{\Delta}_E^N] \\
&= P^{N+1}((\mathbf{\Delta}_E^N \times \mathbf{\Delta}) \cap A) \\
&\leq \frac{n}{N+1} P^{N+1}(\mathbf{\Delta}_E^{N+1}) - P^{N+1}(\mathbf{\Delta}_E^{N+1} - (\mathbf{\Delta}_E^N \times \mathbf{\Delta})) \quad \text{(using (13))} \\
&\leq \frac{n}{N+1} P^{N+1}(\mathbf{\Delta}_E^N \times \mathbf{\Delta}) \\
&= \frac{n}{N+1} P^N(\mathbf{\Delta}_E^N),
\end{aligned}$$

from which the statement follows. □

4.3 Problems with a convex cost

Consider the robust convex program

$$\begin{aligned}
\min_{x \in {}^{n}} s(x) \quad \text{subject to} \quad & x \in \mathcal{X} \\
& f(x,\delta) \leq 0, \; \delta \in \mathbf{\Delta},
\end{aligned}$$

where $s(x)$ is a convex and continuous function. As it is well known, this problem is equivalent to the following program in epigraphic form, having linear cost

$$\begin{aligned}
\min_{x,\gamma} \gamma \quad \text{subject to} \quad & x \in \mathcal{X} \\
& f(x,\delta) \leq 0, \; \delta \in \mathbf{\Delta} \\
& s(x) - \gamma \leq 0.
\end{aligned}$$

Theorem 1 can be applied to this latter program to conclude that $N \geq \frac{n+1}{\epsilon\beta} - 1$ constraints suffice to obtain an ϵ-level solution with probability $1 - \beta$ (note that we have $n+1$ since the problem now has $n+1$ variables: $[\gamma \; x^T]^T \in \mathbb{R}^{n+1}$).

However, we observe that this epigraphic reformulation is not necessary for the application of Theorem 1. As a matter of fact, the same reasoning as in the proof given in Section 4.1 can be directly applied to the initial program with convex cost, to conclude that $N \geq \frac{n}{\epsilon\beta} - 1$ constraints are still sufficient in this case.

5 Applications and Numerical Examples

5.1 Robust linear programs

To illustrate the theory, we consider first a very specialized family of robust convex programs, namely robust linear programs of the form

$$\min_{x \in \ ^{n}} c^T x \quad \text{subject to} \quad A(\delta)x \leq b, \ \forall \delta \in \Delta, \tag{14}$$

with $A(\delta) \in \mathbb{R}^{p,n}$ and $\mathcal{X} = \mathbb{R}^n$. For particular uncertainty structures (for instance, when $A(\delta)$ is affine in δ, and the set Δ is the direct product of ellipsoids) the above problem can be recast exactly as a convex program with a finite number of constraints and decision variables, and therefore efficiently solved by standard numerical techniques, see [5]. However, if the dependence of A on δ is not affine, and the uncertainty set Δ has a generic structure, only approximated (conservative) solutions can be obtained through relaxation.

For comparison purposes, we discuss here an example for which an exact solution can be computed via standard methods. In particular, we assume that each row $a_i^T(\delta)$ of $A(\delta)$ belongs to an ellipsoid, i.e.

$$a_i(\delta) = \hat{a}_i + E_i\delta_i, \ \|\delta_i\| \leq 1, \ i = 1, \ldots, m,$$

where $\hat{a}_i \in \mathbb{R}^n$ is the center of the ellipsoid, $E_i = E_i^T \in \mathbb{R}^{n,n}$ is the 'shape' matrix, and $\delta = [\delta_1^T \cdots \delta_m^T]^T \in \mathbb{R}^{mn}$. Then, we notice that the constraint $a_i^T(\delta)x \leq b_i$ holds for all $\delta \in \Delta$ if and only if

$$\max_{\|\delta_i\| \leq 1} \hat{a}_i^T x + \delta_i^T E_i x \leq b_i,$$

which in turn holds if and only if $\hat{a}_i^T x + \|E_i x\| \leq b_i$. Therefore, the robust linear program (14) has in this case an *exact* reformulation as the following second order cone program

$$\min_{x\in {}^n} c^T x \quad \text{subject to} \quad \hat{a}_i^T x + \|E_i x\| \leq b_i,\ i = 1,\ldots,m. \tag{15}$$

On the other hand, to pursue the randomized approach, we assume that each vector δ_i is uniformly distributed over the ball $\|\delta_i\| \leq 1$, and, for fixed ϵ, β, we determine N according to (5) and draw N samples $\delta^{(i)},\ldots,\delta^{(N)}$ of δ. The randomized counterpart of (14) is therefore given by the linear program

$$\min_{x\in {}^n} c^T x \quad \text{subject to} \quad A(\delta^{(i)})x \leq b,\ i = 1,\ldots,N.$$

To make a simple example, let us consider the following numerical data

$$A(\delta) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} \delta_1^T \\ \delta_2^T \\ \delta_3^T \\ \delta_4^T \end{bmatrix},\ \|\delta_i\| \leq 1,\ i = 1,\ldots,4,$$

and $b = \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}^T$, $c = \begin{bmatrix} -1 & -1 \end{bmatrix}$. For this data, the exact robust solution computed according to (15), is $x^* = [0.7795 \ \ 0.7795]^T$, with corresponding optimal objective $c^T x^* = -1.5590$. For the randomized counterpart, we selected probabilistic levels $\epsilon = \beta = 0.01$, which requires $N = 19.999$ randomized constraints. The resulting linear program was readily solved on a PC using Matlab LP routine, yielding the solution $\hat{x}_N = [0.7798 \ \ 0.7795]^T$, resulting in the objective value $c^T \hat{x}_N = -1.5594$.

5.2 Robust least-squares problems

We next consider a problem of robust polynomial interpolation borrowed from [12]. For given integer $n \geq 1$, we seek a polynomial of degree $n-1$, $p(t) = x_1 + x_2 t + \cdots + x_n t^{n-1}$, that interpolates m given points (a_i, y_i), $i = 1,\ldots,m$, with minimal squared interpolation error, that is it minimizes $\|Ax - y\|^2$, where

$$A = \begin{bmatrix} 1 & a_1 & \cdots & a_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & a_m & \cdots & a_m^{n-1} \end{bmatrix},\quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix},\quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}.$$

If the data values (a_i, y_i) are known exactly, this problem is a standard least-squares problem. Now, assume that the interpolation points are not known exactly. For instance, we assume that the y_i's are known exactly, while there is interval uncertainty on the abscissae

$$a_i(\delta) = a_i + \delta_i,\ i = 1, \ldots, m,$$

where δ_i are assumed to be uniformly distributed in the intervals $[-\rho, \rho]$, i.e.

$$\Delta = \{\delta = [\delta_1 \cdots \delta_m]^T :\ \|\delta\|_\infty \leq \rho\}.$$

We then seek an interpolant that minimizes the worst-case squared interpolation error, i.e.

$$x^* = \arg \min_{x \in {}^n} \max_{\delta \in \Delta} \|A(\delta)x - y\|^2, \tag{16}$$

where

$$A(\delta) = \begin{bmatrix} 1 & a_1(\delta) & \cdots & a_1^{n-1}(\delta) \\ \vdots & \vdots & & \vdots \\ 1 & a_m(\delta) & \cdots & a_m^{n-1}(\delta) \end{bmatrix}.$$

Clearly, the min-max problem (16) can be cast in standard robust convex programming format as

$$\min_{x,\gamma} \gamma \quad \text{subject to} \quad \|A(\delta)x - y\|^2 \leq \gamma,\ \forall \delta \in \Delta. \tag{17}$$

Due to the non-linear nature of the uncertainty entering the data matrix, it is not known how to solve problem (17) exactly in polynomial time, but it is possible to efficiently minimize an upper bound on the optimal worst-case residual via semidefinite programming, as it is shown in [12].

Considering the numerical data

$$(a_1, y_1) = (1, 1),\ (a_2, y_2) = (2, -0.5),\ (a_3, y_3) = (4, 2),$$

with uncertainty level $\rho = 0.2$ and $n = 3$, the semidefinite relaxation approach of [12] yielded a sub-optimal solution with worst-case (guaranteed) residual error equal to 1.1573.

To apply our randomized approach, we assumed uniform distribution for the uncertain parameters, and selected probabilistic levels $\epsilon = \beta = 0.1$, which requires $N = 399$ random samples of δ. The randomized counterpart of (17) can then be expressed as the following semidefinite program

$$\min_{x,\gamma} \gamma \quad \text{subject to} \quad \begin{bmatrix} \gamma & (A(\delta^{(i)})x - y)^T \\ (A(\delta^{(i)})x - y) & I \end{bmatrix} \succeq 0, \; i = 1, \ldots, N. \qquad (18)$$

Problem (18) was easily solved on a PC using standard software, and yielded the solution $\hat{x}_N = [3.7539 \;\; -3.5736 \;\; 0.7821]^T$, with corresponding residual equal to 0.6993. This residual makes a $\sim$ 40% improvement over the one resulting from the deterministic semidefinite relaxation approach. Of course, this improvement comes at some cost: the computed residual is not guaranteed against *all possible* uncertainties, but only *for most* of them.

Since we used a relatively small number of samples to determine the randomized solution, we proceed with an a-posteriori Monte-Carlo test in order to determine a more precise estimate of the violation probability for the computed solution. Running this a-posteriori test with $\tilde{N} = 10^6$ on the solution $\hat{x}_N$ resulted in an estimated violation probability $\hat{V}_{\tilde{N}}(\hat{x}_N) = 0.0042$. Moreover, by the Hoeffding bound (6), we are 99.99% confident that the actual violation probability is close to the estimated one, up to $\tilde{\epsilon} = 0.002$. To summarize the results, the randomized program (18) yielded a solution which provides a $\sim$ 40% performance improvement in the residual error with respect to the semidefinite robust relaxation method, at the expense of a maximum $\sim$ 0.6% risk of constraint violation.

5.3 Solving semidefinite programs using linear programming

In this latter example, we show an application of the randomized methodology to a problem where the semi-infinite constraints do not arise in consequence of actual uncertainty in the problem data, but are 'artificially' introduced by a suitable reformulation of the problem. Consider a standard formulation of a semidefinite program

$$\text{SDP:} \; \min_{x \in {}^n} \; c^T x \;\; \text{subject to} \quad F(x) \preceq 0,$$

where $F(x) = F_0 + \sum_{i=1}^{n} x_i F_i$, $F_i = F_i^T$. Clearly, the linear matrix inequality constraint $F(x) \preceq 0$ can be reformulated as a semi-infinite (or robust) constraint of the form

$$z^T F(x) z \leq 0, \; \forall z : \|z\| = 1.$$

The above constraint actually represents an infinite set of *linear* constraints on the problem variable x:

$$[z^T F_1 z \cdots z^T F_n z] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \leq -z^T F_0 z, \quad \forall z : \|z\| = 1,$$

and therefore SDP can be represented as a robust linear program. This type of representation and its consequences in relation to bundle solution methods have been recently studied in [17].

Now, assuming that the z's are sampled according to some probability distribution (for instance, uniform over the surface of the unit hyper-sphere), we can state the randomized counterpart of SDP as

$$\mathrm{SDP}_N : \min_{x \in \ ^n} c^T x \quad \text{subject to} \quad [z^{(i)T} F_1 z^{(i)} \cdots z^{(i)T} F_n z^{(i)}] x \leq -z^{(i)T} F_0 z^{(i)}, \; i = 1, \ldots, N,$$

which is indeed a linear program in n variables and N constraints.

As a simple example, let us consider the problem of minimizing the largest eigenvalue of a symmetric matrix $A(x)$ of the form

$$A(x) = A_0 + x_1 A_1 + \cdots + x_p A_p, \; A_i = A_i^T \in \mathbb{R}^{m,m},$$

which corresponds to the SDP

$$\min_{x \in \ ^p, \lambda} \lambda \quad \text{subject to} \quad A(x) \preceq \lambda I. \tag{19}$$

For the following numerical data

$$A_0 = \begin{bmatrix} 18 & -1 & 4 & -3 & -2 \\ -1 & -2 & -5 & 14 & -5 \\ 4 & -5 & 16 & 12 & -1 \\ -3 & 14 & 12 & -4 & -3 \\ -2 & -5 & -1 & -3 & -16 \end{bmatrix}, \quad A_1 = \begin{bmatrix} -12 & -17 & 0 & 1 & -7 \\ -17 & 8 & 4 & -2 & 3 \\ 0 & 4 & 0 & -3 & 1 \\ 1 & -2 & -3 & -6 & 2 \\ -7 & 3 & 1 & 2 & -14 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -14 & 6 & -5 & 3 & -3 \\ 6 & -6 & 14 & 3 & -3 \\ -5 & 14 & 12 & -3 & 12 \\ 3 & 3 & -3 & -8 & 1 \\ -3 & -3 & 12 & 1 & -6 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 8 & 5 & 7 & -5 & 5 \\ 5 & 18 & -5 & -3 & -12 \\ 7 & -5 & -10 & 14 & 3 \\ -5 & -3 & 14 & -14 & -14 \\ 5 & -12 & 3 & -14 & 18 \end{bmatrix},$$

the solution of (19) using a standard SDP solver yielded an optimal objective $\lambda^* = 20.8026$ with corresponding $x^* = [0.5765\ 0.0037\ \ -0.2673]^T$.

For the randomized problem, we assumed uniform distribution for z, and selected probabilistic levels $\epsilon = \beta = 0.01$, which (since $n = p + 1 = 4$) requires $N = 39,999$ sampled constraints. Notice that the uniform samples on the surface of the unit hypersphere can be easily generated as $z = \xi/\|\xi\|$, where ξ is normal with zero mean. Solving the linear program SDP$_N$ yielded the optimal objective $\hat{\lambda}^* = 20.7269$, which is indeed a lower bound on λ^*, with corresponding $\hat{x}_N = [0.5424\ \ -0.0124\ \ -0.3050]^T$. The resulting matrix $A(\hat{x}_N)$ has a maximum eigenvalue $\lambda_{max} = 20.8455$.

Remark 4 Let us take a closer look at the above numerical example. The randomized approach yields a solution $(\hat{x}_N, \hat{\lambda}^*)$ that does *not* satisfy the constraint $A(\hat{x}_N) - \hat{\lambda}^* I \preceq 0$, since it is only probabilistically guaranteed in the linear program reformulation. Thus, $\hat{\lambda}^*$ is a lower bound for the optimal value of the original problem. Then, the largest eigenvalue λ_{max} of $A(\hat{x}_N)$ has been determined and, by construction, λ_{max} does satisfy relation $A(\hat{x}_N) - \lambda_{max} I \preceq 0$. So, the final result is that by the randomized approach we have determined a sub-optimal – but close to optimal – solution to the original problem.

This randomized approach seems particularly effective for determining approximate solutions in problems with relatively small number of variables n and size of the matrices F_i so large to be intractable by means of current SDP solvers. In fact, one may observe that the bound (5) depends only on n, and that (5.3) are scalar constraints, no matter how large the size of the F_i's is. □

6 Conclusions

In this paper, the concept of ϵ-level solution for an uncertain convex problem has been introduced. This concept is based on the assumption that the 'instance parameter' δ that parameterizes the constraint family is a random variable. In this case, we have proven that a randomized version RCP_N of the problem returns a solution which is feasible for 'most' of the constraints in the family (i.e. an ϵ-level solution) with high probability, provided that a sufficient number N of samples is drawn. Moreover, an efficient bound for N which scales linearly with the problem dimension n and is inversely proportional to the product of the probability levels $\epsilon\beta$, is derived.

In contrast to the NP-hardness of many robust convex programs, this paper shows that, if a small risk of failure is accepted, the uncertain convex problem can be solved efficiently in the ϵ-level sense by a randomized algorithm, no matter the way in which the uncertainty enters the data, and irrespective of the structure of the uncertainty set $\boldsymbol{\Delta}$.

References

[1] P. Apkarian and H.D. Tuan. Parameterized LMIs in control theory. *SIAM Journal on Control and Optimization*, 38(4):1241–1264, 2000.

[2] B.R. Barmish and P. Sherbakov. On avoiding vertexization of robustness problems: the approximate feasibility concept. *IEEE Trans. Aut. Control*, 47:819–824, 2002.

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

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

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

[6] A. Ben-Tal and A. Nemirovski. On tractable approximations of uncertain linear matrix inequalities affected by interval uncertainty. *SIAM Journal on Optimization*, 12(3):811–833, 2002.

[7] G. Calafiore and M.C. Campi. Interval predictors for unknown dynamical systems: an assessment of reliability. In *Proc. IEEE Conf. on Decision and Control*, Las Vegas, USA, December 2002.

[8] G. Calafiore and F. Dabbene. A probabilistic framework for problems with real structured uncertainty in systems and control. *Automatica*, 38(8):1265–1276, 2002.

[9] G. Calafiore and B. Polyak. Stochastic algorithms for exact and approximate feasibility of robust LMIs. *IEEE Trans. Aut. Control*, 46(11):1755–1759, November 2001.

[10] D.P. de Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *unpublished manuscript, http://www.stanford.edu/bvr/psfiles/sampling.pdf*, 2002.

[11] L. El Ghaoui and G. Calafiore. Robust filtering for discrete-time systems with bounded noise and parametric uncertainty. *IEEE Trans. Aut. Control*, 46(7):1084–1089, July 2001.

[12] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM J. on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

[13] L. El Ghaoui and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM J. Optim.*, 9(1):33–52, 1998.

[14] M.X. Goemans and D.P. Williamson. .878-approximation for MAX CUT and MAX 2SAT. *Proc. 26th ACM Sym. Theor. Computing*, pages 422–431, 1994.

[15] C. Guestrin, D. Koller, and R. Parr. Efficient solution algorithms for factored MDPs. *unpublished manuscript*, 2002.

[16] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[17] K. Krishnan. *Linear programming (LP) approaches to semidefinite programming (SDP) problems.* Ph.D. Thesis. Rensselaer Polytechnic Institute, Troy, New York, 2001.

[18] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, November 1998.

[19] J.R. Morrison and P.R. Kumar. New linear program performance bound for queuing networks. *Journal of Optimization Theory and Applications*, 100(3):575–597, 1999.

[20] J.M. Mulvey, R.J. Vanderbei, and S.A. Zenios. Robust optimization of large-scale systems. *Operations Research*, 43:264–281, 1995.

[21] M.J. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

[22] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, March 1996.