Giuseppe Calafiore · M.C. Campi

Uncertain convex programs: randomized solutions and confidence levels

Abstract. Many engineering problems can be cast as optimization problems subject to convex constraints that are parameterized by an uncertainty or 'instance' parameter. Two main approaches are generally available to tackle constrained optimization problems in presence of uncertainty: robust optimization and chance-constrained optimization. Robust optimization is a deterministic paradigm where one seeks a solution which simultaneously satisfies all possible constraint instances. In chance-constrained optimization a probability distribution is instead assumed on the uncertain parameters, and the constraints are enforced up to a pre-specified level of probability. Unfortunately however, both approaches lead to computationally intractable problem formulations.

In this paper, we consider an alternative 'randomized' or 'scenario' approach for dealing with uncertainty in optimization, based on constraint sampling. In particular, we study the constrained optimization problem resulting by taking into account only a finite set of N constraints, chosen at random among the possible constraint instances of the uncertain problem. 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 and 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.

1. Introduction

Uncertain convex programming [4, 15] 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 $\delta \in \mathbf{\Delta} \subseteq \mathbb{R}^\ell$

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

where $x \in \mathcal{X}$ is the optimization variable, $\mathcal{X}$ is convex and closed, and the function $f(x, \delta) : \mathcal{X} \times \mathbf{\Delta} \to \mathbb{R}$ is convex in x for all $\delta \in \mathbf{\Delta}$. The function $f(x, \delta)$ is here

G. Calafiore: Dipartimento di Automatica e Informatica, Politecnico di Torino, corso Duca degli Abruzzi, 24, 10129 Torino, Italy. Tel.: +39-011-564 7071; Fax: +39-011-564 7099
e-mail: giuseppe.calafiore@polito.it

M.C. Campi: Dipartimento di Automatica per l'Automazione, Università di Brescia, via Branze 38, 25123 Brescia, Italy. e-mail: campi@ing.unibs.it

assumed to be scalar-valued without loss of generality, since multiple scalar-valued convex constraints $f_i(x, \delta) \leq 0$, $i = 1, \ldots, n_f$, may always be converted into a single scalar-valued convex constraint of the form $f(x, \delta) = \max_{i=1,\ldots,n_f} f_i(x, \delta) \leq 0$. Also, in the problem family (1) the optimization objective is assumed to be linear and 'certain' without loss of generality.

1.1. Current solution approaches

Two main and distinct philosophies of solution to uncertain programs are currently found in the literature: a probabilistic approach based on 'chance constraints', and a deterministic one based on 'robust optimization'. The chance-constrained approach has the longest history, dating back to the work of Charnes and Cooper for linear programs in 1959, [10]. The essence of this probabilistic approach is to consider the uncertainty parameter δ as a random variable and to enforce the constraints up to a desired level of probability. More precisely, if P is the probability on $\boldsymbol{\Delta}$, and $\epsilon \in [0, 1]$ is an acceptable 'risk' of constraint violation, the chance (or probability) constrained version of the uncertain program is the following program

$$\text{PCP} : \min_{x \in \mathcal{X} \subseteq \mathbb{R}^n} c^T x \quad \text{subject to } P\{f(x, \delta) > 0\} \leq \epsilon. \tag{2}$$

Unfortunately however, such kind of optimization problems turn out to be extremely difficult to solve exactly. Moreover, even if $f(x, \delta)$ is convex in x for all δ, the feasible set $\{x : P\{f(x, \delta) > 0\} \leq \epsilon\}$ may be nonconvex, and hence PCP is *not* a convex program in general. We direct the reader to the monograph by Prékopa [27] for an extensive presentation of many available results on chance-constrained optimization.

An alternative to the chance-constrained approach to the solution of uncertain programs is the so-called 'min-max' or 'worst-case' approach. While the worst-case paradigm is classical in statistical decision theory, numerically efficient algorithms (mainly interior point methods for convex programming) for the solution of worst-case optimization problems in some specific cases appeared only recently in the literature, see [3–5, 14, 15]. Perhaps due to the influence of robust control theory on this particular area of optimization, the term 'robust optimization' was employed in the above references to denote the min-max or worst-case approach.

In robust optimization one looks 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 robust convex program

$$\text{RCP:} \min_{x \in \mathbb{R}^n} c^T x \quad \text{subject to } x \in \mathcal{X} \cap \Omega, \tag{3}$$

where

$$\Omega \doteq \bigcap_{\delta \in \boldsymbol{\Delta}} \{x : f(x, \delta) \leq 0\} \tag{4}$$

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

Notable special cases of the above problem are robust linear programs [5], for which $f(x, \delta)$ is affine in x, and robust semidefinite programs [15], for which the set Ω is expressed as

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

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 have found applications in many contexts, such as truss topology design [3], robust antenna array design, portfolio optimization, and robust estimation and filtering, [13, 15]. 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 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, 17]), 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 [22]), and robust second order cone programs may result in semidefinite programs, [31, 32].

1.2. A computationally feasible paradigm: Sampled convex programs

Motivated by the computational complexity of the discussed methods for uncertain convex programming, in this paper we pursue a different philosophy of solution, which is based on randomization of the parameter δ. Similar to the probabilistic approach, we assume that the uncertain problem family (1) is parameterized by an instance parameter δ which is a random variable with probability P. Then, we collect N randomly chosen samples $\delta^{(1)}, \ldots, \delta^{(N)}$ of the instance parameter, and construct the sampled convex program

$$\text{SCP}_N : \ \min_{x \in \mathbb{R}^n} c^T x \text{ subject to } \ x \in \mathcal{X}$$

$$f(x, \delta^{(i)}) \leq 0, \ i = 1, \ldots, N. \tag{5}$$

This sampled (or 'randomized') program has a distinctive advantage over RCP and PCP: it is a standard convex program with N constraints, and hence it is typically efficiently solvable. However, a fundamental question need be addressed: what can we say about the constraint satisfaction for an optimal solution of SCP_N?

The feasible set of the randomized problem SCP_N is an outer approximation of the feasible set of RCP. Therefore, the randomized program yields an optimal objective

value that outperforms the optimal objective value of RCP. The price which is paid for this enhancement is that the randomized solution is feasible for many – but not all – of the instances of δ. In this connection, the crucial question to which this paper is devoted 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 (3) by sampling a sufficient number of constraints. This result credits the method with wide applicability. Moreover, we show that an optimal solution resulting from the sampled problem (5) is feasible (with high probability) for the chance-constrained problem (2).

Deterministic 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 [24]. Dynamic programming is considered in [18] where an approximated cost-to-go function is introduced to implement a linear programming-based solution with a low number of constraints. A similar approach has also been independently proposed in [28].

These mentioned contributions propose ad-hoc constraint reduction methods that exploit the specific structure of the problem at hand. A considerable body of literature also exists on so-called column generation methods, which are typically employed for linear programs with a very large number of variables. The dual analog of these methods can indeed be viewed as a constraint reduction technique, and it is related to Kelley's cutting plane methods for convex programming, [20]. We address the reader to the survey [23] and the references therein for further discussion on this topic. Other methods are also known in linear programming that start by solving a subproblem with a randomly chosen subset of the original constraints, and then iteratively update this subset by eliminating inactive constraints and adding violated ones, see for instance Section 9.10 of [25].

The literature on randomized methods for uncertain convex optimization problems is instead very scarce. A noteworthy contribution is [12], in which a constraint sample complexity evaluation for uncertain linear programs is derived, motivated by applications in dynamic programming. The bound on the sample complexity in [12] is based on the Vapnik-Chervonenkis (VC) theory, [33, 34], and this contribution has the important merit of bringing instruments from the statistical learning literature of uniform convergence into the realm of robust optimization. Following a similar approach, a sample complexity evaluation for a certain class of quadratic convex programs has also been independently derived in [7]. The contribution of the present paper is somehow different in spirit from [12] and [7]. We no longer rely on the VC theory, but instead our approach hinges upon the introduction of so-called 'support constraints' (see Definition 4). In this way we gain two fundamental advantages: i) generalizing the VC approach to different classes of convex programs (other than linear or quadratic) would require to determine

an upper bound on the VC-dimension for the specific problem class under consideration, which is in general a difficult task that can possibly lead to conservative estimates. Such an evaluation is not required along our approach, where the sample complexity can be straightforwardly computed; ii) more fundamentally, our results in Theorem 1 and Corollary 1 hold for *any* convex program, and therefore even for constraint sets having infinite VC-dimension, in which case the VC theory is not even applicable. As an additional remark, we mention that the sample complexity evaluation in [12] holds for all feasible solutions of the optimization problem and not just for the optimal solution, contrary to the evaluation derived here. On the one hand, this fact may introduce conservatism in the evaluation of [12], since the bound holds for other feasible solutions, besides the optimal one. On the other hand, having a sample complexity evaluation valid for all feasible solutions has interest in certain contexts, such as the ones studied in [7].

In the different – though strictly related – setting of feasibility determination, the idea of approximate feasibility in robust semidefinite programming has been discussed in [2], and stochastic algorithms for approximate feasibility are studied in [9]. Constraint sampling schemes for large scale uncertain programs have also recently been proposed in [26].

The paper is organized as follows. Section 2 contains the main result (Theorem 1), whose complete proof is reported in a separate section (Section 3). 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, while 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 for (1). The* probability of violation *of x is defined as*

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

(here, it is assumed that $\{\delta \in \mathbf{\Delta} : \ f(x, \delta) > 0\}$ 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 $V(x) \leq \epsilon$.* □

Notice that, by the above definition, any ϵ-level solution is a feasible solution for the chance-constrained optimization problem (2). 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 formally introduce a randomized convex program as follows.

Definition 3 (Sampled convex program). *Let $\delta^{(1)}, \dots, \delta^{(N)}$ be N independent identically distributed samples extracted according to probability P. The sampled convex program derived from (1) is*

$$\text{SCP}_N: \quad \min_{x \in \mathbb{R}^n} c^T x \text{ subject to } x \in \mathcal{X}$$

$$f(x, \delta^{(i)}) \leq 0, \ i = 1, \dots, N. \tag{6}$$

$\square$

For the time being, we assume that SCP_N admits a unique solution. Clearly, should SCP_N be unfeasible (i.e. $\cap_{i=1,\dots,N} \{x : f(x, \delta^{(i)}) \leq 0\} \cap \mathcal{X} = \emptyset$), 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 SCP_N. Since the constraints $f(x, \delta^{(i)}) \leq 0$ are randomly selected, the optimal solution $\hat{x}_N$ is a random variable that depends on the extraction of the multi-sample $\delta^{(1)}, \dots, \delta^{(N)}$.

The following key theorem pinpoints the properties of $\hat{x}_N$.

Theorem 1. *Let $\hat{x}_N$ be the (unique) solution to SCP_N. Then,*

$$E_{P^N}[V(\hat{x}_N)] \leq \frac{n}{N+1}, \tag{7}$$

where n is the size of x, and P^N ($= P \times \cdots \times P$, N times) is the probability measure in the space $\mathbf{\Delta}^N$ of the multi-sample extraction $\delta^{(1)}, \dots, \delta^{(N)}$. $\square$

The proof of Theorem 1, which requires the statement of some preliminary results, is given in Section 3.2.

An immediate consequence of Theorem 1 is that the average probability of violation of $\hat{x}_N$ is proportional to the size of the optimization variable x, and goes to zero linearly with the number N of sampled constraints.

From Theorem 1, we also derive the following corollary.

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

$$N \geq \frac{n}{\epsilon \beta} - 1. \tag{8}$$

Then, with probability no smaller than $1 - \beta$, the randomized problem SCP_N returns an optimal solution $\hat{x}_N$ which is ϵ-level robustly feasible. $\square$

Proof. To see that Corollary 1 follows from Theorem 1, notice that

$$P^N\{V(\hat{x}_N) > \epsilon\} \leq \frac{1}{\epsilon} E_{P^N}[V(\hat{x}_N)] \leq \frac{1}{\epsilon} \frac{n}{N+1},$$

where the first inequality is the Markov inequality, while the last one follows from (7). Then, using (8) we immediately obtain that

$$P^N\{V(\hat{x}_N) > \epsilon\} \leq \frac{1}{\epsilon}\frac{n}{N+1} \leq \frac{1}{\epsilon}\frac{n}{\frac{n}{\epsilon\beta}-1+1} = \beta,$$

which proves the statement. □

A subtle measurability issue arises regarding the definition of $V(\hat{x}_N)$. In fact, without any extra assumptions, there is no guarantee that $V(\hat{x}_N)$ is measurable, so that $E_{P^N}[V(\hat{x}_N)]$ may not be well-defined. Here and elsewhere, the measurability of $V(\hat{x}_N)$ is taken as an assumption.

We here remark that the 'sample complexity' of SCP$_N$ (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. For reasonable probabilistic levels, the required number of these constraints appears to be manageable by current convex optimization numerical solvers.

2.1. Discussion on main result

We next comment more closely on the proposed randomized approach.

2.1.1. 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 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 (8) is probability independent (i.e. it holds irrespective of the underlying probability P) and can therefore be applied even when P is unknown.

2.1.2. Feasibility vs. performance.
Efficient solution techniques for the RCP problem are known only for certain simple dependencies of f on δ, such as affine, polynomial or rational. In other cases, one should consider a probabilistic approach, for which the randomized technique 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 SCP$_N$ leaves instead a chance to the occurrence of δ's

which are violated by the solution. On the other hand, SCP_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 1. In certain problems, allowing even for a tiny probability ϵ of constraint violation can change significantly the problem solution and, possibly, result in a significant improvement of the optimization value for those instances that remain feasible. One should therefore bear in mind that the optimal objective obtained from a probabilistic approach can be significantly different from the optimal objective obtained from a robust approach, even if a very small violation probability is imposed in the probabilistic solution. An extreme example of this situation is the following. Let $\boldsymbol{\Delta} = [0, 1]$, $x \in \mathbb{R}$ and

$$f(x, \delta) = \begin{cases} \frac{1}{\alpha} - x, & \delta \in [0, \alpha] \\ -x, & \delta \in (\alpha, 1], \end{cases}$$

with δ uniformly distributed in $[0, 1]$, and where α is a given small positive number, say $\alpha = 10^{-6}$. Let further the minimization objective $c^T x$ be specified by $c = 1$. Then, the RCP problem would yield an optimal objective value equal to $1/\alpha = 10^6$. On the other hand, setting a violation probability $\epsilon > \alpha$, the probabilistic problem PCP would instead yield an optimal objective value equal to zero, since it would neglect the fact that the constraint is violated for uncertainties lying in a set of measure smaller than ϵ. Neglecting a 'bad set' of small probability thus resulted in a dramatic improvement in the attainable performance.

2.1.3. 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 Corollary 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 no 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 $\delta \in \boldsymbol{\Delta}$ 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, [19], states that

$$P^{\tilde{N}}\{|\hat{V}_{\tilde{N}}(\hat{x}_N) - V(\hat{x}_N)| \leq \tilde{\epsilon}\} \geq 1 - 2\exp(-2\tilde{\epsilon}^2\tilde{N}),$$

from which it follows 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

$$\tilde{N} \geq \frac{\log 2/\tilde{\beta}}{2\tilde{\epsilon}^2} \tag{9}$$

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.

2.1.4. Semi-infinite optimization. From a broader perspective, robust convex programs belong to the class of so-called 'semi-infinite' programs, i.e. optimization problems in which the number of constraints is infinite, see e.g. [16]. In this context, a usual solution approach consists in 'discretization' or gridding of the variable $\delta \in \mathbf{\Delta}$ that parameterizes the constraints, see e.g. [30] and the references therein. As it is well known, the number of grid points, and hence of constraints, grows exponentially with the dimension of $\mathbf{\Delta}$, thus making discretization unpractical in high dimensional problems. In this connection, using random constraint sampling as proposed in this paper has the important advantage that the required number of constraints as given by (8) is *independent* of the dimension of the parameter set $\mathbf{\Delta}$.

2.1.5. Stochastic optimization. We finally remark that the proposed randomized approach to uncertain convex problems is different in spirit from the classical stochastic approximation methods used in stochastic programming, [11, 29], though even in the latter context a scenario-like approach is often used. Indeed, in the classical approach one seeks to optimize the expectation of a utility function. Since the exact expectation is hard to compute (and hence to optimize), randomly generated instances (or scenarios, i.e. our δ's) are used to construct an empirical version of the expectation, and this is subsequently optimized. In this paper, we are not interested in optimizing on average; instead, we seek a solution that is optimal among all solutions that satisfy all but a few constraint instances. This latter approach is motivated by an extensive literature on chance-constrained optimization and robust optimization, see e.g. [4, 10, 12, 15, 27].

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.

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 $\xi_3, \ldots, \xi_{p+2}$.* $\qquad\square$

Proof. Choose any set S' composed of $p - 1$ points from S, and consider the bundle of hyperplanes passing through S'.[1] If this bundle has more than one degree of freedom, augment S' with additional arbitrary points, until the bundle has exactly one degree of

---

[1] A bundle of hyperplanes passing though a set of $p - 1$ points is simply the collection of all $(p - 1)$-dimensional affine subspaces containing the set of points.

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_1, x_2, x_3 the translated version of the initial points that were not in S'. 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 $\dim \mathcal{V} = 2$, 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 that $\alpha_1 \alpha_2 < 0$. We now have that

$$v_3^T x_1 = (\alpha_1 v_1 + \alpha_2 v_2)^T x_1 = \alpha_2 v_2^T x_1$$
$$v_3^T x_2 = (\alpha_1 v_1 + \alpha_2 v_2)^T x_2 = \alpha_1 v_1^T x_2,$$

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. $\qquad\square$

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

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

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

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

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 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 $c^T x_k^* < c^T x^*$.* $\qquad\square$

The following theorem holds.

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

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 $c^T \bar{x}_k^* = \alpha < c^T x^*$, 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. $\square$

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 $\mathbf{\Delta}$ with probability P and consider the following $N+1$ instances of SCP_N:

$$\mathrm{SCP}_N^k : \min_{x \in \mathbb{R}^n} c^T x \text{ subject to } x \in \mathcal{X}$$

$$f(x, z^{(i)}) \leq 0, \ i = 1, \ldots, k-1, k+1, \ldots, N+1.$$

For $k = 1, \ldots, N+1$, let $\hat{x}_N^k$ be the optimal solution of problem SCP_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 $f(\hat{x}_N^k, z^{(k)}) \leq 0$. Define

$$\bar{V}_N \doteq E_{P^N}[V(\hat{x}_N)], \tag{10}$$

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

$$\hat{\bar{V}}_N \doteq \frac{1}{N+1} \sum_{k=1}^{N+1} v_k. \tag{11}$$

We have that

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

which yields

$$E_{P^{N+1}}[\hat{\bar{V}}_N] = \bar{V}_N. \tag{12}$$

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

$$\hat{\bar{V}}_N(\bar{z}^{(1)}, \ldots, \bar{z}^{(N+1)}) \leq \frac{n}{N+1}. \tag{13}$$

Thus, by taking expectation we still have

$$E_{P^{N+1}}[\hat{\bar{V}}_N] \leq \frac{n}{N+1}. \tag{14}$$

and this concludes the proof in view of (10) and (12).

To show (13), consider the convex problem involving all the $N + 1$ constraints

$$\text{SCP}_{N+1} : \min_{x \in \mathbb{R}^n} c^T x \text{ subject to } x \in \mathcal{X}$$

$$f(x, \bar{z}^{(i)}) \leq 0, \ i = 1, \ldots, N + 1,$$

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 $\text{SCP}_N^k, k = 1, \ldots, N + 1$, 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 SCP_{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 (11) equation (13) follows. $\quad\square$

4. Extensions

4.1. Problems with multiple optimal solutions

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

Consider problem SCP_N (6). 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 $t_i(x)$, $i = 1, \ldots, p$, be given convex and continuous functions. Among the optimal solutions for SCP_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 $t_3(x), t_4(x), \ldots$. We assume that functions $t_i(x)$, $i = 1, \ldots, p$, 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 $t_1(x) = x_1, t_2(x) = x_2, \ldots$.* ☐

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 SCP_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 4 and Theorem 2 need suitable amendments. Precisely, we now have:

Definition 5 (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 5 is a generalization of Definition 4 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^*, $k = 1, \ldots, n + 1$, be the optimal solutions for the corresponding $\mathcal{P}_k$ problems. We show that $x^* \notin \mathrm{co}\{x_1^*, \ldots, x_{n+1}^*\}$, and therefore a $(n - 1)$-dimensional hyperplane separating x^* from $x_1^*, \ldots, x_{n+1}^*$ 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 $x^* \in \mathrm{co}\{x_1^*, \ldots, x_{n+1}^*\}$, and hence x^* can be written as $x^* = \sum_{i \in I \subset \{1, \ldots, n+1\}} \alpha_i x_i^*$, $0 < \alpha_i \leq 1$, $\sum_{i \in I} \alpha_i = 1$. 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 $t_2(x), \ldots, t_p(x)$,

we conclude that, for any i: $c^T x_i^* = c^T x^*$, $t_1(x_i^*) = t_1(x^*), \ldots, t_p(x_i^*) = t_p(x^*)$, but this is impossible since then $t_1(x), \ldots, t_p(x)$ would not give a tie-break rule. Thus, we have a contradiction and $x^* \notin \mathrm{co}\{x_1^*, \ldots, x_{n+1}^*\}$.

Consider now a $(n-1)$-dimensional hyperplane $\mathcal{H}$ separating x^* from $x_1^*, \ldots, x_{n+1}^*$ (and not touching x^*) and construct $\bar{x}_1^*, \ldots, \bar{x}_{n+1}^*$ 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 $\bar{x}_1^*, \ldots, \bar{x}_{n+1}^*$: $\bar{x}^* \in \mathrm{co}\{\bar{x}_1^*, \ldots, \bar{x}_{n+1}^*\}$. 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 $\bar{x}^* = \sum_{j \in J \subset \{1, \ldots, n+1\}} \beta_j \bar{x}_j^*$, $0 < \beta_j \leq 1$, $\sum_{j \in J} \beta_j = 1$. 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 $t_1(x), t_2(x), \ldots$. 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 $t_1(x), \ldots, t_p(x)$. 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 SCP_N has no solution. This may happen when the set $\cap_{i=1,\ldots,N} \{x : f(x, \delta^{(i)}) \leq 0\} \cap \mathcal{X}$ is unbounded in such a way that the optimal solution 'escapes' to infinity, while the original problem is constrained to a set $\cap_{\delta \in \Delta} \{x : f(x, \delta) \leq 0\} \cap \mathcal{X}$ 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 multi-sample $\delta^{(1)}, \ldots, \delta^{(N)}$ is rejected when no optimal solution exists, and another extraction is performed in this case. Then, on average on the accepted multi-samples, $V(\hat{x}_N)$ is no larger than $\frac{n}{N+1}$. In formal terms, this involves considering conditional probability as precisely stated in the next theorem.

Theorem 4. Let $\Delta_E^N \subseteq \Delta^N$ be the set where a solution of SCP_N exists. If $P^N(\Delta_E^N) > 0$, the result in Theorem 3 generalizes to the following result:

$$\frac{E_{P^N}[V(\hat{x}_N) \cap 1(\Delta_E^N)]}{P(\Delta_E^N)} \leq \frac{n}{N+1} \tag{15}$$

(note that when $\Delta_E^N = \Delta^N$ as assumed in Theorem 3, we recover the statement of Theorem 3). Moreover, in this case Corollary 1 still holds, provided that $1-\beta$ is intended as a lower bound on the conditional probability $P^N(\{V(\hat{x}_N) \leq \epsilon\} \cap \Delta_E^N)/P^N(\Delta_E^N)$. (the measurability of Δ_E^N is taken as an assumption). □

Proof. We sketch here how the proof of Theorem 3 can be amended to cope with the present setting. Let $\Delta_E^{N+1} \subseteq \Delta^{N+1}$ be the set where a solution of the problem with

$N + 1$ constraints exists, and note that $\mathbf{\Delta}_E^N \times \mathbf{\Delta} \subseteq \mathbf{\Delta}_E^{N+1}$ 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

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

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

$$
\frac{n}{N+1} P^{N+1}(\mathbf{\Delta}_E^{N+1}) \geq E_{P^{N+1}}\left[\frac{1}{N+1} \sum_{k=1}^{N+1} v_k\right]
$$

$$
= E_{P^{N+1}}[v_{N+1}] = P^{N+1}(\mathbf{\Delta}_E^{N+1} \cap (A \cup B)),
$$

with $A \doteq \{f(\hat{x}_N^{N+1}, z^{(N+1)}) > 0\}$, $B \doteq \{\hat{x}_N^{N+1} \text{ does not exist}\}$. 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

$$
\frac{n}{N+1} P^{N+1}(\mathbf{\Delta}_E^{N+1}) \geq P^{N+1}((\mathbf{\Delta}_E^N \times \mathbf{\Delta}) \cap A) + P^{N+1}(\mathbf{\Delta}_E^{N+1} - (\mathbf{\Delta}_E^N \times \mathbf{\Delta})). \quad (16)
$$

Finally, we have:

$$
\begin{aligned}
E_{P^N}[V(\hat{x}_N) \cap 1(\mathbf{\Delta}_E^N)] &= P^{N+1}((\mathbf{\Delta}_E^N \times \mathbf{\Delta}) \cap A) \\
&\leq \frac{n}{N+1} P^{N+1}(\mathbf{\Delta}_E^{N+1}) \\
&\quad - P^{N+1}(\mathbf{\Delta}_E^{N+1} - (\mathbf{\Delta}_E^N \times \mathbf{\Delta})) \quad \text{(using (16))} \\
&\leq \frac{n}{N+1} P^{N+1}(\mathbf{\Delta}_E^N \times \mathbf{\Delta}) \\
&= \frac{n}{N+1} P^N(\mathbf{\Delta}_E^N),
\end{aligned}
$$

from which statement (15) follows. The corresponding extension of Corollary 1 is easily obtained, following similar steps as in the proof of Corollary 1. □

4.3. Problems with a convex cost

Consider the robust convex program

$$
\min_{x \in \mathbb{R}^n} s(x) \text{ subject to } x \in \mathcal{X}
$$

$$
f(x, \delta) \leq 0, \ \forall \delta \in \mathbf{\Delta},
$$

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

$$
\min_{x, \gamma} \gamma \text{ subject to } x \in \mathcal{X}
$$

$$
f(x, \delta) \leq 0, \ \forall \delta \in \mathbf{\Delta}
$$

$$
s(x) - \gamma \leq 0.
$$

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 of Theorem 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

$$\min_{x \in \mathbb{R}^n} c^T x \quad \text{subject to} \quad A(\delta)x \leq b, \; \forall \delta \in \mathbf{\Delta}, \tag{17}$$

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 $\mathbf{\Delta}$ 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 $\mathbf{\Delta}$ 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 $\delta \in \mathbf{\Delta}$ 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 (17) has in this case an *exact* reformulation as the following second order cone program

$$\min_{x \in \mathbb{R}^n} c^T x \quad \text{subject to} \quad \hat{a}_i^T x + \|E_i x\| \leq b_i, \; i = 1, \ldots, m. \tag{18}$$

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 (8) and draw N samples $\delta^{(i)}, \dots, \delta^{(N)}$ of δ. The randomized counterpart of (17) is therefore given by the linear program

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

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

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

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 (18), 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 [14]. For given integer $n \geq 1$, we seek a polynomial of degree $n-1$, $p(t) = x_1 + x_2 t + \dots + x_n t^{n-1}$, that interpolates m given points (a_i, y_i), $i = 1, \dots, m$, 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

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

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

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

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

$$x^* = \arg \min_{x \in \mathbb{R}^n} \max_{\delta \in \boldsymbol{\Delta}} \|A(\delta)x - y\|^2, \tag{19}$$

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 (19) can be cast in standard robust convex programming format as

$$\min_{x,\gamma} \gamma \quad \text{subject to} \quad \|A(\delta)x - y\|^2 \le \gamma, \; \forall \delta \in \mathbf{\Delta}. \tag{20}$$

Due to the non-linear nature of the uncertainty entering the data matrix, it is not known how to solve problem (20) 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 [14].

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 [14] 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 (20) can then be expressed as the following semidefinite program

$$\min_{x,\gamma} \gamma \quad \text{subject to} \quad \begin{bmatrix} \gamma & (A(\delta^{(i)})x - y)^T \\ (A(\delta^{(i)})x - y) & I \end{bmatrix} \succeq 0, \; i = 1, \ldots, N. \tag{21}$$

Problem (21) 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 (9), 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 (21) 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

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

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

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

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

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

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 [21].

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

$$\text{SDP}_N : \min_{x \in \mathbb{R}^n} c^T x \ \text{ subject to } \ [z^{(i)T} F_1 z^{(i)} \cdots z^{(i)T} F_n z^{(i)}] x$$
$$\le -z^{(i)T} F_0 z^{(i)}, \ i = 1, \dots, N,$$

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

We remark that in the present case one wants to solve a *deterministic* problem, while the randomized approach results in a solution that fails in general to satisfy the deterministic constraint $F(x) \preceq 0$. Indeed, for purely deterministic problems, such as the one at hand, one cannot argue that 'bad sets' are negligible simply because the probability of violating the constraints is small. As a consequence, the randomized approach yields in this case only a lower bound on the optimal objective value of the original problem, and the former value can be far away from the latter one in some cases, see also Remark 1. Moreover, this issue may become more apparent in high dimensional cases. The randomized technique can however work satisfactorily for small dimensional problems, and it is here reported mainly for the purpose of illustration of our probabilistic methodology.

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

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

which corresponds to the SDP

$$\min_{x \in \mathbb{R}^q, \lambda} \lambda \quad \text{subject to} \quad A(x) \preceq \lambda I. \tag{22}$$

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 (22) 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 the number of optimization variables is $n = q + 1 = 4$) requires $N = 39,999$ sampled constraints. Notice that the uniform samples on the surface of the unit hyper-sphere can be easily generated as $z = \xi/\|\xi\|$, where ξ is normal with zero mean and unit variance. 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 2. 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. □

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 SCP_N of the uncertain 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.

In contrast to the NP-hardness of generic robust and chance-constrained 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. Apkarian, P., Tuan, H.D.: Parameterized LMIs in control theory. SIAM J. Control Optim **38** (4), 1241–1264 (2000)
2. Barmish, B.R., Sherbakov, P.: On avoiding vertexization of robustness problems: the approximate feasibility concept. IEEE Trans. Aut. Control **47**, 819–824 (2002)
3. Ben-Tal, A., Nemirovski, A.: Robust truss topology design via semidefinite programming. SIAM J. Optim **7** (4), 991–1016 (1997)
4. Ben-Tal, A., Nemirovski, A.: Robust convex optimization. Math. Oper. Res. **23** (4), 769–805 (1998)
5. Ben-Tal, A., Nemirovski, A.: Robust solutions of uncertain linear programs. Oper. Res. Lett. **25** (1), 1–13 (1999)
6. Ben-Tal, A., Nemirovski, A.: On tractable approximations of uncertain linear matrix inequalities affected by interval uncertainty. SIAM J. Optim. **12** (3), 811–833 (2002)
7. Calafiore, G., Campi, M.C., El Ghaoui, L.: Identification of reliable predictor models for unknown systems: a data-consistency approach based on learning theory. In: Proceedings of IFAC 2002 World Congress, Barcelona, Spain, 2002
8. Calafiore, G., Dabbene, F.: A probabilistic framework for problems with real structured uncertainty in systems and control. Automatica **38** (8), 1265–1276 (2002)
9. Calafiore, G., Polyak, B.: Stochastic algorithms for exact and approximate feasibility of robust LMIs. IEEE Trans. Aut. Control **46** (11), 1755–1759, November 2001
10. Charnes, A., Cooper, W.W.: Chance constrained programming. Manag. Sci. **6**, 73–79 (1959)
11. Dantzig, G.B.: Linear programming under uncertainty. Manage. Sci. **1**, 197–206 (1955)
12. de Farias, D.P., Van Roy, B.: On constraint sampling in the linear programming approach to approximate dynamic programming. Technical Report Dept. Management Sci. Stanford University, 2001
13. El Ghaoui, L., Calafiore, G.: Robust filtering for discrete-time systems with bounded noise and parametric uncertainty IEEE Trans. Aut. Control **46** (7), 1084–1089, July 2001
14. El Ghaoui, L., Lebret, H.: Robust solutions to least-squares problems with uncertain data. SIAM J. Matrix Anal. Appl. **18** (4), 1035–1064 (1997)
15. El Ghaoui, L., Lebret, H.: Robust solutions to uncertain semidefinite programs. SIAM J. Optim. **9** (1), 33–52 (1998)
16. Goberna, M.A., Lopez, M.A.: Linear Semi-Infinite Optimization. Wiley, 1998
17. Goemans, M.X., Williamson, D.P.: .878-approximation for MAX CUT and MAX 2SAT. Proc. 26th ACM Sym. Theor. Computing, 1994, pp. 422–431
18. Guestrin, C., Koller, D., Parr, R.: Efficient solution algorithms for factored MDPs. To appear in J. Artificial Intelligence Research, submitted 2002
19. Hoeffding, W.: Probability inequalities for sums of bounded random variables. J. Am. Statistical Association **58**, 13–30 (1963)
20. Kelley, J.E.: The cutting-plane method for solving convex programs. J. Soc. Ind. Appl. Math. **8** (4), 703–712 (1961)
21. Krishnan, K.: Linear programming (LP) approaches to semidefinite programming (SDP) problems. Rensselaer Polytechnic Institute, Ph.D. Thesis, Troy, New York, 2001
22. Lobo, M., Vandenberghe, L., Boyd, S., Lebret, H.: Applications of second-order cone programming. Linear Algebra and its Applications **284**, 193–228 November 1998
23. Luebbecke, M.E., Desrosiers, J.: Selected topics in column generation. Les Cahiers de GERAD G-2002-64, 2002, pp. 193–228
24. Morrison, J.R., Kumar, P.R.: New linear program performance bound for queuing networks J. Optim. Theor. Appl. **100** (3), 575–597 (1999)

25. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press, Cambridge, 1995
26. Mulvey, J.M., Vanderbei, R.J., Zenios, S.A.: Robust optimization of large-scale systems. Oper. Res. **43**, 264–281 (1995)
27. Prékopa, A.: Stochastic Programming. Kluwer, 1995
28. Schuurmans, D., Patrascu, R.: Direct value-approximation for factored MDPs. Adv. Neural Infor. Processing (NIPS), 2001
29. Shapiro, A., Homem-de-Mello, T.: On rate of convergence of Monte Carlo approximations of stochastic programs. SIAM J. Optim. **11**, 70–86 (2000)
30. Still, G.: Discretization in semi-infinite programming: Rate of convergence. Math. Program. Ser. A **91**, 53–69 (2001)
31. Todd, M.J.: Semidefinite optimization. Acta Numerica **10**, 515–560 (2001)
32. Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Rev. **38** (1), 49–95, March 1996
33. Vapnik, V.N., Ya.Chervonenkis, A.: On the uniform convergence of relative frequencies to their probabilities. Theor. Probab. Appl. **16** (2), 264–280 (1971)
34. Vidyasagar, M.: A theory of learning and generalization. Springer-Verlag, London, 1997