A general scenario theory for non-convex optimization and decision making

Marco C. Campi[1], Simone Garatti[2], and Federico A. Ramponi[1]

***Abstract*—The scenario approach is a general methodology for data-driven optimization that has attracted a great deal of attention in the past few years. It prescribes that one collects a record of previous cases (scenarios) from the same setup in which optimization is being conducted and makes a decision which attains optimality for the seen cases. Scenario optimization is by now very well understood for *convex* problems, where a theory exists that rigorously certifies the generalization properties of the solution, that is, the ability of the solution to perform well in connection to new situations. This theory supports the scenario methodology and justifies its use. This paper considers *non-convex* problems. While other contributions in the non-convex setup already exist, we here take a major departure from previous approaches. We suggest that the generalization level is evaluated only after the solution is found and its complexity in terms of the length of a support sub-sample (a notion precisely introduced in this paper) is assessed. As a consequence, the generalization level is stochastic and adjusted case by case to the available scenarios. This fact is key to obtain tight results. The approach adopted in this paper applies not only to optimization, but also to generic decision problems where the solution is obtained according to a rule which is not necessarily the optimization of a cost function. Accordingly, in our presentation we adopt a general stance of which optimization is just seen as a particular case.**

Keywords: Scenario approach, stochastic programming, non-convex optimization, robust decision-making, robust control.

I. Introduction and goal of the paper

Many problems in the theory and practice of systems and control can be formulated as decision problems. For instance, in PID controller tuning, the proportional, integral and derivative gains may be seen as decision variables that must be selected so as to satisfy given performance specifications. In optimal input design, instead, the decision variable is the input signal, which must be decided so as to minimize some given cost functional. Likewise, optimal state filtering can be seen as a decision problem where one minimizes the state prediction error (e.g. in the mean square sense), and the decision variables are the filter parameters.

In this paper, we deal with *data-driven* decision-making where a procedure generates a decision based on a collection of observations coming from previous experience. The observations are used to account for the variability of the conditions to which the decision can possibly be applied.

[1]Marco C. Campi and Federico Alessandro Ramponi are with the Dipartimento di Ingegneria dell'Informazione, Università di Brescia, Italy. Emails: {marco.campi,federico.ramponi}@unibs.it

[2]Simone Garatti is with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy. Email: simone.garatti@polimi.it

Some definitions help to rapidly focus on the main ideas. Let Δ be a probability space, endowed with a σ-algebra $\mathcal{D}$ and a probability measure P. An element $\delta \in \Delta$ is interpreted as a potential situation to which the decision can be applied, while P describes the chance of such a situation to occur. Moreover, let $(\Delta^m, \mathcal{D}^m, \mathsf{P}^m)$ be the m-fold Cartesian product of Δ equipped with the product σ-algebra $\mathcal{D}^m$ and the product probability $\mathsf{P}^m = \mathsf{P} \times \cdots \times \mathsf{P}$ (m times). A point in $(\Delta^m, \mathcal{D}^m, \mathsf{P}^m)$ is thus a sample $(\delta^{(1)}, \cdots, \delta^{(m)})$ of m elements drawn independently from Δ according to the same probability P.[1] Each $\delta^{(i)}$ is regarded as an observation, and in the following we will also call it a *scenario*. A set Θ, called the *decision space*, contains the decisions. It can possibly be infinite, and no particular structure, e.g. that of vector space or convex set, is assumed. The decision-maker is equipped with a procedure to make a decision based on $(\delta^{(1)}, \cdots, \delta^{(m)})$.[2] Later, we shall provide various examples of procedures. Formally, the procedure is modeled as a family of functions $\mathcal{A}_m : \Delta^m \to \Theta$, indexed by the size $m = 0, 1, \ldots$ of the sample,[3] and the decision $\theta^*_m := \mathcal{A}_m(\delta^{(1)}, \cdots, \delta^{(m)})$ is called the *scenario decision*.

The following assumption is in force throughout this paper.

Assumption 1: To every $\delta \in \Delta$ there is associated a constraint set $\Theta_\delta \subseteq \Theta$, which identifies the decisions that are admissible for the situation represented by δ. For all $m = 1, 2, \ldots$ and for any sample $(\delta^{(1)}, \cdots, \delta^{(m)})$, it holds that $\mathcal{A}_m(\delta^{(1)}, \cdots, \delta^{(m)}) \in \Theta_{\delta^{(i)}}$ for all $i = 1, \cdots, m$. □

Remark 1: The requirement that $\mathcal{A}_m(\delta^{(1)}, \cdots, \delta^{(m)}) \in \Theta_{\delta^{(i)}}$ for all i is natural in many problems where this requirement prescribes that the decision is admissible for all the collected situations, see e.g. the examples below. Note that this requirement establishes a link, albeit weak, between the functions $\cdots, \mathcal{A}_{m-1}, \mathcal{A}_m, \mathcal{A}_{m+1}, \cdots$. □

Remark 2: The requirement that $\mathcal{A}_m$ is a *function* amounts to

[1]One could as well introduce a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and define $\delta^{(i)}$ as independent random elements over this probability space. This is completely equivalent to the construction considered in this paper, since $(\Omega, \mathcal{F}, \mathbb{P})$ can be taken as $(\Delta^\infty, \mathcal{D}^\infty, \mathsf{P}^\infty)$, which always exists thanks to the Ionescu-Tulcea theorem, [35].

[2]The decision-maker has access to $(\delta^{(1)}, \cdots, \delta^{(m)})$, and her/his decision is therefore based on knowledge that comes from experience. S/he is not required instead to know P in order to apply the results of this paper, that is, all theoretical certificates hold independently of P.

[3]For $m = 0$, $\mathcal{A}_0$ has no argument and it is meant that it gives a fixed element in Θ.

requiring that the solution to the decision problem is *unique*. To conform to this condition, when a decision problem admits multiple solutions, one has to implement a "tie-break rule" to single out one solution. For example, if Θ is a normed vector space, a simple tie-break rule is to choose the solution with minimum norm. In this paper, the tie-break rule is seen as an inherent part of the decision process, included in $\mathcal{A}_m$. □

The present setup is quite broad and encompasses problems of various kinds. We next give some examples (optimization, feasibility, ...) that are of particular interest to us. A more concrete example in control is presented in Section IV. In Section V we come back to the generality of the setup introduced by Assumption 1 and show that this assumption can be applied also to problems that are not born in an optimization context.

Example 1 (Optimization): Let Θ be a subset of $\mathbb{R}^d$ ($\mathbb{R}$ is the set of real numbers), $f : \Theta \to \mathbb{R}$ be any function and, for each $\delta \in \Delta$, let Θ_δ be a subset of $\mathbb{R}^d$. Given $(\delta^{(1)}, \cdots, \delta^{(m)}) \in \Delta^m$, consider the following constrained optimization program:

$$\begin{array}{l} \min\limits_{\theta \in \Theta} f(\theta) \\ \text{subject to } \theta \in \Theta_{\delta^{(i)}} \text{ for all } i = 1, \cdots, m. \end{array} \tag{1}$$

Assuming that a unique solution θ^*_m exists, possibly after applying a tie-break rule, (1) defines a map $\mathcal{A}_m$ that associates θ^*_m to $(\delta^{(1)}, \cdots, \delta^{(m)})$. □

When f, Θ, and Θ_δ are convex, program (1) is a convex scenario program in the form that has been studied in [5], [6], [11]. These seminal papers have introduced the so-called scenario approach, which, as witnessed by many contributions like e.g. [12], [39], [22], [33], [15], [28], [16], [29], [17], [43], has rapidly gained recognition, and has found application to various problems in control, [36], [37], [21], [31]. The optimization program (1) is much more general than the setup of [5], [6], [11] since no assumptions on f, Θ, and Θ_δ are made. It includes mixed-integer constrained optimization as a particular case, which we shall consider more in detail in Section IV. An example of application to a control problem is given in Section IV-A, while system identification problems along a similar approach have been considered in [14].

Example 2 (Algorithms for optimization): In Example 1, the decision is the solution to an optimization problem. However, obtaining the optimal solution can be difficult, especially in a non-convex setting. In practice, one often uses a numerical algorithm $\mathcal{A}_m$ to compute a solution $\tilde{\theta}^*_m$ that can as well be a sub-optimal solution. The algorithm $\mathcal{A}_m$ can be seen as a map from $(\delta^{(1)}, \cdots, \delta^{(m)})$ to $\tilde{\theta}^*_m$, and the theory of this paper can be applied to the sub-optimal solution $\tilde{\theta}^*_m$ returned by $\mathcal{A}_m$. □

Example 3 (Feasibility problems): Suppose that one wants to find a feasible point for a set of constraints, that is,

$$\begin{array}{l} \text{find } \theta \in \Theta \\ \text{subject to } \theta \in \Theta_{\delta^{(i)}} \text{ for all } i = 1, \cdots, m, \end{array}$$

and that a rule is set to determine one such feasible point. Again, this defines a map $\theta^*_m = \mathcal{A}_m(\delta^{(1)}, \cdots, \delta^{(m)})$. □

A. Goal of the paper

Up to here, we have considered m scenarios to introduce Assumption 1 where m was a running variable and the requirement $\mathcal{A}_m(\delta^{(1)}, \cdots, \delta^{(m)}) \in \Theta_{\delta^{(i)}}$, $i = 1, \ldots, m$, had to hold for any m. We henceforth call N the actual, fixed, number of scenarios that we observe in a given application. The goal of this paper is to study how well a scenario decision $\theta^*_N = \mathcal{A}_N(\delta^{(1)}, \cdots, \delta^{(N)})$ generalizes to yet unseen situations $\delta \in \Delta$. This is important to certify how "robust" θ^*_N is against new situations in which θ^*_N may be applied. To explain what "how well" means, we start by introducing the terminology that θ^*_N *generalizes* to $\delta \in \Delta$ if $\theta^*_N \in \Theta_\delta$; in the opposite, we say that θ^*_N *violates* δ. "How well" is formalized in probabilistic terms as follows.

Definition 1: The *violation probability* of a given decision $\theta \in \Theta$ is defined as

$$\mathsf{V}(\theta) := \mathsf{P}\{\delta \in \Delta : \ \theta \notin \Theta_\delta\}.$$

For a given reliability parameter $\varepsilon \in (0, 1)$, we say that $\theta \in \Theta$ is ε-feasible (or ε-robust) if $\mathsf{V}(\theta) \leq \varepsilon$. □

The violation of the scenario decision $\mathsf{V}(\theta^*_N)$, which is the composition of $\mathsf{V}(\cdot)$ with $\theta^*_N = \mathcal{A}_N(\delta^{(1)}, \cdots, \delta^{(N)})$ is a random variable defined over Δ^N. We want to study the distribution of $\mathsf{V}(\theta^*_N)$ and find a suitable confidence bound $1 - \beta$ for the validity of the relation $\mathsf{V}(\theta^*_N) \leq \varepsilon$.[4] Depending on the problem at hand, violating a constraint means that a control performance (a settling time, a certain level of noise rejection, etc.), a prediction result (the next point is within a given prediction interval), or a correct classification (the case at hand is classified within the right class) is not achieved, and knowing a bound on $\mathsf{V}(\theta^*_N)$ provides guarantees on the chance of this to happen. In the context of optimization (Example 1), establishing that $\mathsf{V}(\theta^*_N) \leq \varepsilon$ can be interpreted as an assessment of the feasibility of θ^*_N for a chance-constrained problem at level ε, see e.g. [32], [19], [30], [27], [4], [34] for contributions on chance-constrained optimization. We do not further dwell on the interpretation of the violation probability and for this we refer the reader to the existing literature, e.g. [13], [9], [10], [18]. In particular, paper [6] discusses a number of applications to control. Later in Sections IV and V we shall exhibit various examples to illustrate the theory of this paper, and this will provide further examples of the concept of violation.

[4] The perspective of this paper, as suggested by the fact that θ^*_N is required to belong to all the $\Theta_{\delta^{(i)}}$'s, is that the smaller the violation, the better the solution. In some cases, especially in optimization, this may not be true, since too small a violation may correspond to obtaining a poor performance. If this is the case, alternative approaches can be adopted to accommodate the requirement that $\mathsf{V}(\theta^*_N)$ should not be too small. For instance, one may want to allow that θ^*_N fails to belong to some of the $\Theta_{\delta^{(i)}}$'s, see e.g. [12], [22]. This is not further investigated here and is left for future research.

B. Discussion on existing results

The distribution of $\mathsf{V}(\theta_N^*)$ has been the object of intense study for the case when θ_N^* is obtained as the solution of a convex optimization program, [5], [6], [11]. The deepest result is established in [11], where it is shown that the distribution of $\mathsf{V}(\theta_N^*)$ is dominated by a Beta distribution, namely,

$$\mathsf{P}^N\{\mathsf{V}(\theta_N^*) > \varepsilon\} \leq \beta, \tag{2}$$

where

$$\beta = \sum_{i=0}^{d-1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i}, \tag{3}$$

and d is the number of optimization variables. This result is tight in that (2) holds with equality for a whole class of convex optimization problems, those named fully-supported in [11]. Moreover, the result is distribution-free, that is, it holds for any P, which is important to make the theory of [11] practical and applicable in a purely observation-based framework, where no information on P is available other than that carried by $\delta^{(1)}, \cdots, \delta^{(N)}$.

The fundamental fact on which the theory of [11] stands is that the number of support constraints[5] in a convex optimization problem with d optimization variables never exceeds d. This fact fails to be true in non-convex optimization; an example is given in Figure 1, where the removal of any of the 6 constraints generates a new solution. Hence, routes alternative to that used in [11] have to be pursued in a non-convex context.

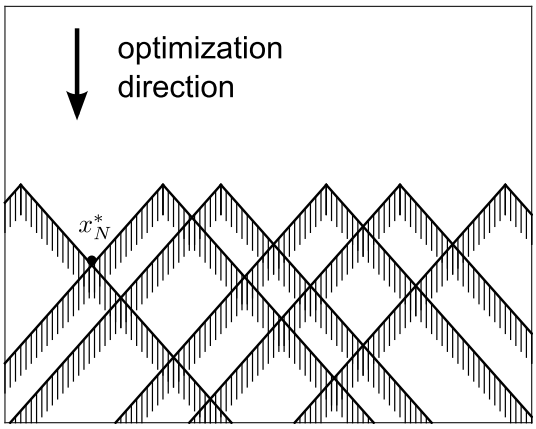


Figure 1. In this non-convex program, all constraints are of support since eliminating any one of them generates a new feasible point that outperforms the solution with all constraints in place.

Some previous attempts to address a non-convex setup are the following. Paper [1] uses concepts from the statistical learning literature, [40], [41], [42], to bound the probability that $\mathsf{V}(\theta_N^*) \leq \varepsilon$ in non-convex scenario optimization. While inspiring, this approach suffers from the conservatism inherent in the Vapnik-Chervonenkis theory, [38]. A non-convex cost function optimized under convex constraints is instead considered in [23]. In this paper, the feasibility domain is restricted to a region which is obtained as the convex hull of few points to enable the application of the result from [11]. Again, the result is conservative, besides being applicable to a restricted class of problems only. Papers [7], [20] consider mixed-integer problems, and a theory akin to that of [11] is applied after showing that the number of support constraints can be a priori upper bounded in mixed integer problems. However, this bound turns out to be very large.

[5]A constraint $\theta \in \Theta_{\delta^{(i)}}$ is said to be a *support constraint* if the program obtained by removing that constraint, while keeping all the others, has a solution different from the solution of the initial program.

C. The approach of this paper

In this paper, we address the evaluation of the feasibility of θ_N^* along a different route, which, in a somewhat different context, has been recently discovered by two of the authors of the present contribution, [8]. We abandon the idea that the number of support scenarios is computed a priori; instead, we assume that after computing θ_N^* one is able to isolate a sub-sample of scenarios sufficient to yield the same solution θ_N^* that is obtained with all the scenarios in place (we show that this task can be accomplished at a relatively low computational cost). In the new approach, the reliability guarantee depends on the cardinality s_N^* of the sub-sample of scenarios, and *the smaller the cardinality, the higher the reliability*. More precisely, the obtained result takes the form:

$$\mathsf{P}^N\{\mathsf{V}(\theta_N^*) > \varepsilon(s_N^*)\} \leq \beta, \tag{4}$$

which closely resembles (2), albeit with the fundamental difference that ε is here no longer fixed in advance and it depends on s_N^*. Along this approach, the assertion on $\mathsf{V}(\theta_N^*)$ is adjusted to the seen scenarios and this by and large improves over previous evaluations established for the non-convex case.

It is worth remarking that the result in (4) does not allow us to a priori compute a number N of scenarios sufficient to obtain a chance-constrained solution at a given level ε. This is because the level depends on the probabilistic outcome and can only be a posteriori computed. This sets a fundamental difference with the results of [11] where a priori conditions are established such that, with high confidence, the solution is a chance-constrained solution at a specified level ε. While this fact may appear to weaken the quality of the result established here compared to previous achievements in a convex setup, we remark that this is due to the generality of the problem considered in this paper, where an a priori bound on the number of support scenarios does not exist. On the other hand, a posteriori establishing the level of violation has great importance in the practical use of scenario-based solutions because, based on the a posteriori value for the violation probability and also in the light on the cost value that has been achieved, one can decide whether the solution is or is not satisfactory and therefore is or is not adopted.

It is further worth highlighting that the fundamental difference between the present paper and [8] is that the latter paper deals with optimization problems under a crucial *non-degeneracy* assumption. In the language of this paper (see Definition 2

in the next Section II), such assumption is phrased as: *with probability* 1*, the problem has a unique irreducible support sub-sample, consisting precisely of the support constraints*. In paper [8], the emphasis is on convex optimization problems where this assumption is very mild. In contrast, in a non-convex setup this assumption is very restrictive, a fact that is discussed in detail in Section 8 of [8]. In this paper, we succeed in removing the non-degeneracy assumption. Moreover, the results we obtain are very general and apply to generic decision problems and not only to optimization. However, we must also mention that the theory of this paper does not allow one to recover as a particular case the results of [11], which is instead possible by using the results of [8]. This is the price we pay for generality, and it is a fact that the results in [11] fail to be true at the level of generality adopted in this paper. See Appendix A where a more detailed discussion on this point is provided.

D. Structure of the paper

Section II provides the technical background and states the main result in formal terms. After the proof of the main result is given in Section III, Section IV revisits mixed-integer scenario optimization in the light of the new theory of this paper. A more general perspective is then taken in Section V, which presents a collection of other problems to which the results of this paper can be applied.

II. Generalization result

We start with the definition of support sub-sample.

Definition 2: Given a sample $(\delta^{(1)}, \cdots, \delta^{(N)}) \in \Delta^N$, a *support sub-sample* S for $(\delta^{(1)}, \cdots, \delta^{(N)})$ is a k-tuple of elements extracted from $(\delta^{(1)}, \cdots, \delta^{(N)})$, i.e. $S = (\delta^{(i_1)}, \cdots, \delta^{(i_k)})$ with $i_1 < i_2 < \cdots < i_k$, which gives the same solution as the original sample, that is,

$$\mathcal{A}_k(\delta^{(i_1)}, \cdots, \delta^{(i_k)}) = \mathcal{A}_N(\delta^{(1)}, \cdots, \delta^{(N)}).$$

□

A support sub-sample $S = (\delta^{(i_1)}, \cdots, \delta^{(i_k)})$ is said to be *irreducible* if no element can be further removed from S leaving the solution unchanged. In general, multiple irreducible support sub-samples can be found for the same sample $(\delta^{(1)}, \cdots, \delta^{(N)})$.

To apply the results of this paper, the user has to determine a support sub-sample for the problem at hand. Clearly, the whole sample $(\delta^{(1)}, \cdots, \delta^{(N)})$ is itself a support sub-sample. In general, the smaller the support sub-sample, the stronger the generalization result; the goal is therefore that of determining a small support sub-sample, possibly an irreducible one, or even the irreducible support sub-sample with minimal length. Finding a minimal-length irreducible support sub-sample can be computationally intensive and it may require brute-force exploration. We stress, however, that, while failing to find a minimal-length support sub-sample leads to results that are not the strongest possible, the conclusions of this paper hold rigorously for non-minimal support sub-samples as well. A greedy algorithm to search for a support sub-sample, which in many cases is computationally efficient and effective, is as follows ($|L|$ denotes the length of a sequence L, and $L \setminus \delta^{(i)}$ is the sub-sequence obtained by removing $\delta^{(i)}$ from L):

1) Set $L \leftarrow (\delta^{(1)}, \cdots, \delta^{(N)})$ and compute the solution $\theta_N^* \leftarrow \mathcal{A}_N(L)$;
2) For all $i = 1, \cdots, N$:
   - set $L' \leftarrow L \setminus \delta^{(i)}$ and compute the solution $\bar{\theta} \leftarrow \mathcal{A}_{|L'|}(L')$;
   - if $\bar{\theta} = \theta_N^*$, then set $L \leftarrow L'$;
3) Output the set $\{i_1, \cdots, i_k\}$, $i_1 < \cdots < i_k$, of the indexes of the elements in L.

For scenario optimization programs in the form (1), it is easy to prove that this algorithm returns an irreducible (although possibly not minimal) support sub-sample. For more general scenario decision problems there is no guarantee that the algorithm returns an irreducible support sub-sample. In these cases, one can iterate over the above algorithm, each time initializing with the value of L returned by step 3 of the previous iteration; this procedure will eventually converge to an irreducible support sub-sample. The greedy algorithm above requires to solve a decision problem N times. At worst, each time one has to deal with N scenarios while, in typical cases, the size of the scenario set decreases as elements $\delta^{(i)}$ get removed from L. In some situations, solving even one problem is time-consuming (e.g. when one deals with a non-convex optimization problem) so that running the greedy algorithm can become computationally intensive. In these cases, alternative algorithmic choices can be conceived to achieve better computational efficiency at the price of obtaining a larger support sub-sample, but we do not dwell on further describing this issue here because it is problem dependent. It may also be of interest to note that in some specific problems (see e.g. the problems in Section V), there is not even the need to run any greedy algorithm, since the size of the minimal support sub-sample is immediately evident from the structure of the problem.

An algorithm to determine a support sub-sample like the one above can be regarded as a function $\mathcal{B}_N : (\delta^{(1)}, \cdots, \delta^{(N)}) \mapsto \{i_1, \cdots, i_k\}$, $i_1 < \cdots < i_k$, such that $(\delta^{(i_1)}, \cdots, \delta^{(i_k)})$ is a support sub-sample. Let

$$s_N^* := |\mathcal{B}_N(\delta^{(1)}, \cdots, \delta^{(N)})|$$

be the cardinality of $\mathcal{B}_N(\delta^{(1)}, \cdots, \delta^{(N)})$ (i.e., the length of the support sub-sample $(\delta^{(i_1)}, \cdots, \delta^{(i_k)})$). Since $\mathcal{B}_N(\delta^{(1)}, \cdots, \delta^{(N)})$ is a random variable over Δ^N, so is s_N^*.

We are now ready to state our main result.

Theorem 1: Suppose that Assumption 1 holds true, and set a value $\beta \in (0, 1)$ (confidence parameter). Let ε :

$\{0, \cdots, N\} \to [0,1]$ be a function such that

$$\begin{aligned} &\varepsilon(N) = 1; \\ &\sum_{k=0}^{N-1} \binom{N}{k} (1-\varepsilon(k))^{N-k} = \beta. \end{aligned} \tag{5}$$

Then, for any $\mathcal{A}_N$, $\mathcal{B}_N$, and probability P, it holds that

$$\mathsf{P}^N \{\mathsf{V}(\theta_N^*) > \varepsilon(s_N^*)\} \leq \beta. \tag{6}$$

□

The proof of Theorem 1 is postponed to Section III.

A simple choice of $\varepsilon(\cdot)$ obtained by splitting β evenly among the N terms in the sum (5) is

$$\varepsilon(k) := \begin{cases} 1 & \text{if } k = N, \\ 1 - \sqrt[N-k]{\dfrac{\beta}{N\binom{N}{k}}} & \text{otherwise.} \end{cases} \tag{7}$$

Figure 2 shows a plot of this $\varepsilon(k)$ for $N = 500$, $N = 1000$, and $N = 2000$, with $\beta = 10^{-6}$.

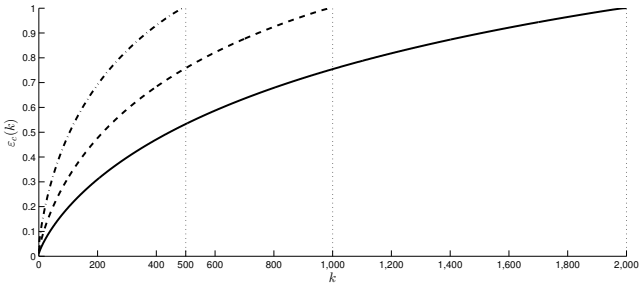


Figure 2. Plot of $\varepsilon(k)$ in (7) for $N = 500$ (dash-dotted line), $N = 1000$ (dashed line), and $N = 2000$ (solid line) ($\beta = 10^{-6}$).

The interpretation of Theorem 1 is as follows. The decision-maker computes the decision θ_N^* along with the length s_N^* of the support sub-sample. The violation of θ_N^* is judged to be no bigger than $\varepsilon(s_N^*)$. For example, with $N = 1000$ and $\beta = 10^{-6}$, if the support sub-sample has $s_N^* = 6$ elements, then from the graph in Figure 2 one obtains $\varepsilon(6) = 5.4\%$ and the claim is that θ_N^* is 5.4%-feasible. If, instead, $s_N^* = 11$, then $\varepsilon(11) = 7.7\%$ and the claim would be that θ_N^* is 7.7%-feasible. Theorem 1 asserts that the claim is true with high confidence $1-\beta$, that is, with confidence $1-10^{-6}$ in the present case. When β is so close to 0 to become practically negligible, one achieves "practical certainty" that the claim is true.

For a given problem, s_N^* is stochastic since it depends on the scenarios $\delta^{(1)}, \delta^{(2)}, \cdots, \delta^{(N)}$, so that the conclusion drawn about the violation of the solution depends on the stochastic realization. This is not surprising and reflects the fact that the solution itself is stochastic. In the example in Section IV, s_N^* has a tendency to be small as compared to N, and the same happens in various problems of the type discussed in Section V. Still, in general it is not always possible to find a support sub-sample that has a priori a small cardinality for any N and it is indeed possible that s_N^* goes to ∞ as $N \to \infty$. An example is offered by the problem in section V.C where, if the probability distributes over infinitely many symbols, then s_N^* goes to ∞ as N grows unbounded. On the other hand, it should also be noted that the fact that s_N^* goes to ∞ does not mean that the violation goes to 1 since the violation is governed by the mutual size of s_N^* and N according to the result in Theorem 1. Finally, there are cases where s_N^* goes to ∞ at the same rate as N for which one cannot draw any good conclusion about the violation rate (and indeed the violation rate remains high even for very large values of N); the reader is referred to the last part of Section V.C for an example. The fundamental message conveyed by Theorem 1 is that one does not need to a priori upper bound s_N^* and the violation can be judged by a posteriori computing the value taken case by case by s_N^*. In other words, given the specific realization of the program one has just solved, the value taken by s_N^* can be computed, and, based on this value, one can draw useful conclusions on the actual violation probability for the program at hand. This sets the fundamental contribution of the paper: even if one cannot a priori claim a chance-constrained result, the actual level of violation probability can be a posteriori evaluated for the obtained solution.

Remark 3: Note that $\varepsilon(k)$ in (7) satisfies

$$\begin{aligned} \varepsilon(k) &= 1 - \exp\left(\log\left(\sqrt[N-k]{\frac{\beta}{N\binom{N}{k}}}\right)\right) \\ &= 1 - \exp\left(-\frac{1}{N-k}\log\frac{1}{\beta} - \frac{1}{N-k}\log N\binom{N}{k}\right) \\ &\leq \frac{1}{N-k}\log\frac{1}{\beta} + \frac{1}{N-k}\log N\binom{N}{k}, \end{aligned} \tag{8}$$

where the last inequality follows from the relation $1-e^{-x} \leq x$. This inequality reveals that $\varepsilon(k)$ has a logarithmic dependence on β, so that a very small value for β ("practical certainty") can be obtained without significantly affecting $\varepsilon(k)$. This is

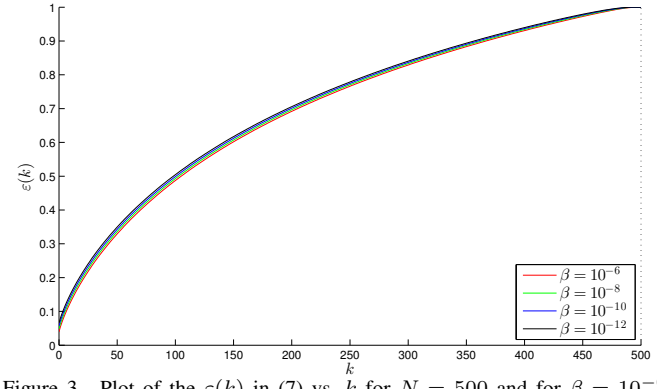


Figure 3. Plot of the $\varepsilon(k)$ in (7) vs. k for $N = 500$ and for $\beta = 10^{-6}$, 10^{-8}, 10^{-10}, 10^{-12}.

clearly visible in Figure 3, which displays the graphs of the $\varepsilon(k)$ in (7) for different values of β when $N = 500$. The weak dependence on β was one of the main advantages of the scenario approach for convex optimization problems, [6],

[2], and is here preserved.

Remark 4: Choices for $\varepsilon(\cdot)$ other than (7) are possible and, at times, advisable. For example, if from the structure of the problem it was known that s_N^* is always less than some $\bar{s}$, then it would make sense to deliberately ignore all the situations where $s_N^* \geq \bar{s}$, thus allowing for stronger claims when $s_N^* < \bar{s}$. One possible choice, where β is split evenly among the terms of (5) corresponding to $k < \bar{s}$, is

$$\varepsilon(k) := \begin{cases} 1 & \text{if } k \geq \bar{s}, \\ 1 - \sqrt[N-k]{\frac{\beta}{\bar{s}\binom{N}{k}}} & \text{otherwise.} \end{cases} \tag{9}$$

Nevertheless, we notice here that any possible improvement over the $\varepsilon(\cdot)$ in (7) has an almost negligible payoff. This is easily understood because, even assigning the whole β to just one k (thus providing the maximum possible improvement for the corresponding $\varepsilon(k)$), yields

$$\varepsilon(k) = 1 - \sqrt[N-k]{\frac{\beta}{\binom{N}{k}}},$$

which is only marginally different from the $\varepsilon(k)$ in (7) (repeating the computation in (8) one gets $\log\binom{N}{k}$ in place of $\log N\binom{N}{k}$). For example, with $N = 1000$ and $\beta = 10^{-6}$, for the choice in (7) we have $\varepsilon(10) = 7.26\%$, while assigning the whole β to $k = 10$ yields $\varepsilon(10) = 6.61\%$.

III. Proof of Theorem 1

Let I_k be a selection of k indexes $\{i_1, \cdots, i_k\}$, $i_1 < \cdots < i_k$, from $\{1, \cdots, N\}$, and let

$$\theta_{I_k} = \mathcal{A}_k(\delta^{(i_1)}, \cdots, \delta^{(i_k)}).$$

Consider the subsets $\Delta_0^N, \cdots, \Delta_N^N$ defined as follows:

$$\Delta_k^N = \left\{(\delta^{(1)}, \cdots, \delta^{(N)}) \in \Delta^N : \ |\mathcal{B}_N(\delta^{(1)}, \cdots, \delta^{(N)})| = k\right\}.$$

The subsets $\Delta_0^N, \cdots, \Delta_N^N$ form a partition of Δ^N. Let us refine such partition by defining for each $k = 0, \cdots, N$ and for any I_k the set $\Delta_{k,I_k}^N \subseteq \Delta_k^N$ according to the following rule: $(\delta^{(1)}, \cdots, \delta^{(N)}) \in \Delta_{k,I_k}^N$ if and only if $\mathcal{B}_N(\delta^{(1)}, \cdots, \delta^{(N)}) = I_k$. It holds that $\Delta_k^N = \bigcup_{I_k} \Delta_{k,I_k}^N$ and

$$\Delta^N = \bigcup_{k=0}^{N} \bigcup_{I_k} \Delta_{k,I_k}^N.$$

Let moreover

$$B = \{(\delta^{(1)}, \cdots, \delta^{(N)}) \in \Delta^N : \ \mathsf{V}(\theta_N^*) > \varepsilon(s_N^*)\}$$

and

$$B_{I_k} = \{(\delta^{(1)}, \cdots, \delta^{(N)}) \in \Delta^N : \ \mathsf{V}(\theta_{I_k}) > \varepsilon(k)\}.$$

We have that

$$\begin{aligned} B = \Delta^N \cap B &= \bigcup_{k=0}^{N} \bigcup_{I_k} \Delta_{k,I_k}^N \cap \{\mathsf{V}(\theta_N^*) > \varepsilon(s_N^*)\} \\ &= [\text{in } \Delta_{k,I_k}^N, \ s_N^* = k \text{ and } \theta_N^* = \theta_{I_k}] \\ &= \bigcup_{k=0}^{N} \bigcup_{I_k} \Delta_{k,I_k}^N \cap \{\mathsf{V}(\theta_{I_k}) > \varepsilon(k)\} \\ &= [\varepsilon(N) = 1 \text{ so that } \{\mathsf{V}(\theta_{I_N}) > \varepsilon(N)\} = \varnothing] \\ &= \bigcup_{k=0}^{N-1} \bigcup_{I_k} \Delta_{k,I_k}^N \cap \{\mathsf{V}(\theta_{I_k}) > \varepsilon(k)\} \\ &= \bigcup_{k=0}^{N-1} \bigcup_{I_k} \Delta_{k,I_k}^N \cap B_{I_k}. \end{aligned}$$

Now focus on any selection I_k of k indexes; to fix ideas, consider $I_k = \{1, \cdots, k\}$. Since the definition of $B_{\{1,\cdots,k\}}$ only involves the first k components, $B_{\{1,\cdots,k\}}$ is a cylinder with base in Δ^k, the Cartesian product of the first k sets Δ. Suppose that $(\bar{\delta}^{(1)}, \cdots, \bar{\delta}^{(k)})$ is a point in the base of such a cylinder; then, a necessary condition for a point $(\bar{\delta}^{(1)}, \cdots, \bar{\delta}^{(k)}, \delta^{(k+1)}, \cdots, \delta^{(N)})$ to belong to $\Delta_{k,\{1,\cdots,k\}}^N \cap B_{\{1,\cdots,k\}}$ is the satisfaction of the constraints $\theta_{\{1,\cdots,k\}} \in \Theta_{\delta^{(k+1)}}, \cdots, \theta_{\{1,\cdots,k\}} \in \Theta_{\delta^{(N)}}$.[6] On the other hand, by the definition of $B_{\{1,\cdots,k\}}$, for any $(\bar{\delta}^{(1)}, \cdots, \bar{\delta}^{(k)})$ in the base of the aforementioned cylinder, it holds that

$$\mathsf{V}(\theta_{\{1,\cdots,k\}}) = \mathsf{P}\left\{\delta \in \Delta : \ \theta_{\{1,\cdots,k\}} \notin \Theta_\delta\right\} > \varepsilon(k).$$

Therefore, by the independence of $\delta^{(k+1)}, \cdots, \delta^{(N)}$, we obtain

$$\begin{aligned} &\mathsf{P}^{N-k}\Big\{(\delta^{(k+1)}, \cdots, \delta^{(N)}) : (\bar{\delta}^{(1)}, \cdots, \bar{\delta}^{(k)}, \delta^{(k+1)}, \cdots, \delta^{(N)}) \\ &\qquad \in \Delta_{k,\{1,\cdots,k\}}^N \cap B_{\{1,\cdots,k\}}\Big\} \\ &\leq \ \mathsf{P}^{N-k}\Bigg\{\bigcap_{i=k+1}^{N}\Big\{(\delta^{(k+1)}, \cdots, \delta^{(N)}) : \\ &\qquad \theta_{\{1,\cdots,k\}} \in \Theta_{\delta^{(i)}}\Big\}\Bigg\} \\ &= \ \prod_{i=k+1}^{N} \mathsf{P}\left\{\delta^{(i)} : \ \theta_{\{1,\cdots,k\}} \in \Theta_{\delta^{(i)}}\right\} \\ &\leq \ \prod_{i=k+1}^{N} (1 - \varepsilon(k)) = (1 - \varepsilon(k))^{N-k}. \end{aligned}$$

Integrating over the base of the cylinder $B_{\{1,\cdots,k\}}$ now yields

$$\begin{aligned} &\mathsf{P}^N\left\{\Delta_{k,\{1,\cdots,k\}}^N \cap B_{\{1,\cdots,k\}}\right\} \\ &\leq (1 - \varepsilon(k))^{N-k} \ \mathsf{P}^k\left\{\text{base of } B_{\{1,\cdots,k\}}\right\} \\ &\leq (1 - \varepsilon(k))^{N-k}. \end{aligned}$$

[6]Note that, contrary to scenario *optimization*, in the general setup of this paper this condition is not sufficient to guarantee that $(\bar{\delta}^{(1)}, \cdots, \bar{\delta}^{(k)}, \delta^{(k+1)}, \cdots, \delta^{(N)}) \in \Delta_{k,\{1,\cdots,k\}}^N \cap B_{\{1,\cdots,k\}}$ since it may happen that, after adding some satisfied constraints, the decision procedure $\mathcal{A}$ returns a solution which is not θ_{I_k} anymore. Hence, the condition of constraint satisfaction is here only necessary. This is one reason why arguments like those used in [11] are not applicable in the context of this paper.

Recall that the choice $I_k = \{1, \cdots, k\}$ was made for the sake of exemplification. In fact, using the same argument, we obtain that $\mathsf{P}^N\left\{\Delta^N_{k,I_k} \cap B_{I_k}\right\} \leq (1-\varepsilon(k))^{N-k}$ for any I_k. Therefore, by sub-additivity,

$$\begin{aligned}\mathsf{P}^N\left\{\mathsf{V}(\theta^*_N) > \varepsilon(s^*_N)\right\} &= \mathsf{P}^N\{B\} \\ &\leq \sum_{k=0}^{N-1}\sum_{I_k}(1-\varepsilon(k))^{N-k} \\ &= \left[\text{there are } \binom{N}{k} \text{ choices of } I_k\right] \\ &= \sum_{k=0}^{N-1}\binom{N}{k}(1-\varepsilon(k))^{N-k} = \beta.\end{aligned}$$

□

IV. Example: mixed-integer scenario optimization and application to control with quantized input

We have already observed that our setup contains as a particular case *mixed-integer* scenario optimization problems. These are programs of the form

$$\begin{aligned}&\min_{\theta\in\Theta'\cap\left(\mathbb{R}^{d_1}\times\mathbb{Z}^{d_2}\right)} f(\theta) \\ &\text{subject to } \theta \in \Theta_{\delta^{(i)}} \text{ for all } i = 1, \cdots, N,\end{aligned} \tag{10}$$

where $\Theta' \subseteq \mathbb{R}^{d_1+d_2}$ is a closed subset and $\mathbb{Z}$ is the set of integer numbers. Program (10) is an instance of (1) where $\Theta = \Theta' \cap \left(\mathbb{R}^{d_1} \times \mathbb{Z}^{d_2}\right)$. Its peculiarity is that the optimization vector θ is partitioned in two parts, the second of which has integer components, namely $\theta = (\theta_1, \theta_2)$, where $\theta_1 \in \mathbb{R}^{d_1}$ and $\theta_2 \in \mathbb{Z}^{d_2}$.

Mixed-integer restrictions to decision variables are often encountered in practice, and scenario programs as in (10) find application in manifold contexts. On the other hand, developing a generalization theory for mixed-integer scenario optimization along "classical" routes where one a priori bounds the length of the support sub-sample leads to conservative results. In [7] it is shown that, when $f(\theta) = c^T\theta$, Θ' is convex and Θ_δ are convex for all δ, the length of a minimal support sub-sample is bounded by $(d_1+1)2^{d_2}-1$, see also [3]. The exponential growth in d_2 poses severe limitations to the applicability of this result to problems other than those with a low dimensional optimization vector, [20]. Things get worse if the convexity assumption on Θ' and Θ_δ is relaxed since no bounds to the length of the minimal support sub-sample are available in this case.

Despite the large a priori bound $(d_1+1)2^{d_2}-1$, often a support sub-sample with way fewer elements than $(d_1+1)2^{d_2}-1$ is a posteriori found. Hence, by adjusting the value of ε to the length of the support sub-sample computed a posteriori as the theory developed in this paper suggests, one can draw significant conclusions about the violation of the solution θ^*_N. All these aspects are more concretely presented on an example for the control of an uncertain linear system with quantized inputs.

A. Control with quantized inputs

Consider the discrete-time uncertain linear system

$$x(t+1) = Ax(t) + Bu(t), \qquad x(0) = x_0, \tag{11}$$

where $x(t) \in \mathbb{R}^2$ is the state variable, $u(t) \in \mathbb{R}$ is the control input, $B = \begin{bmatrix} 0 & 0.5 \end{bmatrix}^\top$ is deterministic, and $A \in \mathbb{R}^{2\times 2}$ is uncertain, with independent Gaussian entries with means

$$\bar{A} = \begin{bmatrix} 0.8 & -1 \\ 0 & -0.9 \end{bmatrix},$$

and standard deviation 0.02 each. Here, we identify a matrix A with a δ in the general theory. The initial state of the system is $x_0 = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$. Moreover, due to actuation constraints, the input is chosen from a *finite* set: $u(t) \in \mathcal{U} := \{-5, \cdots, -1, 0, 1, \cdots, 5\}$.

The control objective is that of driving the system state close to the origin in $T = 8$ time instants by choosing a suitable input sequence $u(0), \cdots, u(T-1)$. Since $x(T) = A^T x_0 + \sum_{t=0}^{T-1} A^{T-1-t}Bu(t)$, if we let

$$R = \begin{bmatrix} B & AB & \cdots & A^{T-1}B \end{bmatrix}$$

and

$$\boldsymbol{u} = \begin{bmatrix} u(T-1) & u(T-2) & \cdots & u(0) \end{bmatrix}^\top$$

the problem can be formulated as that of selecting $\boldsymbol{u}$ in order to make $\left\|A^T x_0 + R\boldsymbol{u}\right\|_\infty = \|x(T)\|_\infty$ as small as possible, where $\|\cdot\|_\infty$ is the maximum norm. Finite-horizon, open-loop problems like this one are common as single steps of more complex receding-horizon MPC schemes; other times, they arise as stand-alone problems in sensor-less environments in which no feedback is possible (e.g. positioning of an end-effector when no exteroceptive sensors are available). The example here is a toy version of these problems used for the purpose of illustrating the theory.

Figure 4 shows the final states $x(8)$ for $N = 1000$ draws of $A^{(i)}$ when: (a) no control action is applied ($u(t) = 0$ for $t = 0, \cdots, 7$); (b) the optimal control sequence for the nominal system $(\bar{A}, B)$, which is $\hat{\boldsymbol{u}} = \begin{bmatrix} -2 & 3 & -2 & 4 & 3 & -5 & 2 & -5 \end{bmatrix}^\top$, is applied. Figure 4(a) shows that relying on the state contraction property alone does not suffice to get close to the origin in 8 time instants, and Figure 4(b) gives evidence of the fact that relying on a nominal controller design is inappropriate because there is too much dispersion due to uncertainty in the final state. Hence, some robustness must be incorporated in the design.

To this purpose, we resorted to the scenario approach. Precisely, the $N = 1000$ scenarios $A^{(i)}$'s were used to construct the scenario program:

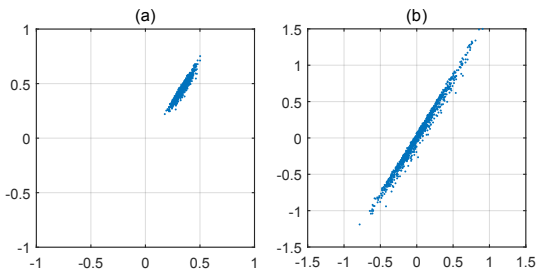


Figure 4. Final state for 1000 systems: (a) no control action; (b) nominal controller.

$$\min_{h\in\mathbb{R},\boldsymbol{u}\in\mathcal{U}} h$$
$$\text{subject to } \left\|(A^{(i)})^T x_0 + R^{(i)}\boldsymbol{u}\right\|_\infty \le h \text{ for all } i = 1,\cdots,N, \tag{12}$$

which aims at finding a discrete control sequence $\boldsymbol{u}$ so as to minimize the largest deviation (for the various $A^{(i)}$'s) of $x(8)$ from the origin. Program (12) is a mixed-integer program in the form (10), with $d_1 = 1$ corresponding to h and $d_2 = 8$ corresponding to $\boldsymbol{u}$. It can be tackled by means of standard numerical solvers like those supported by the optimization modeling interfaces `YALMIP` [26] and `CVX` [25], [24]. We used `YALMIP` equipped with `IBM ILOG CPLEX`™ and the solution was $(h^*, \boldsymbol{u}^*)$ with $h^* = 0.0257$ and $\boldsymbol{u}^* = \left[\begin{array}{cccccccc} 1 & -1 & -4 & 3 & 5 & -4 & -2 & 4 \end{array}\right]^\top$.

Figure 5(a) displays the final states $x(8)$ for the 1000 $A^{(i)}$'s used in (12) (note the different scale on the axes of this figure as compared to Figure 4) when the controller obtained from (12) is used. The same figure also represents the box in the

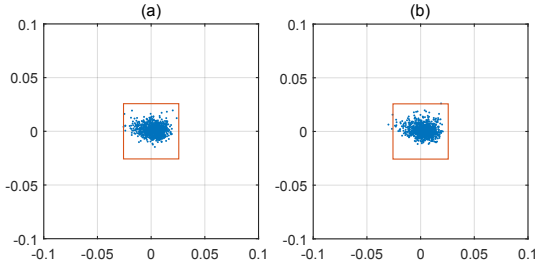


Figure 5. Final state for 1000 systems: (a) scenario controller; (b) validation test.

maximum norm of size $h^* = 0.0257$.

The final states plotted in Figure 5(a) refer to situations that have been used in (12) to determine $(h^*, \boldsymbol{u}^*)$. A natural question to ask is how well $\boldsymbol{u}^*$ performs when it is applied to a new matrix A. This question refers to the robustness of the method against cases that have not been incorporated in the design. In answering this question, we feel advisable to compare alternative approaches. The upper bound of [7] to the length of the minimal support sub-sample is $(d_1+1)2^{d_2} - 1 = 511$, which is too large to draw any meaningful conclusion. On the other hand, by resorting to the greedy algorithm $\mathcal{B}_N$ of Section II – which here consists in removing one constraint

$$\left\|(A^{(i)})^T x_0 + R^{(i)}\boldsymbol{u}\right\|_\infty \le h$$

at a time in succession and to discard it if the solution remains the same – we were left with an irreducible support sub-sample with length $s^*_{1000} = 3$. Hence, choosing $\beta = 10^{-6}$ (practical certainty), and using the function $\varepsilon(\cdot)$ in (7), we found $\varepsilon(s^*_{1000}) = \varepsilon(3) = 0.039$. According to Theorem 1, with confidence at least $1-\beta$, the solution $(h^*, \boldsymbol{u}^*)$ is $\varepsilon(s^*_{1000})$-feasible, which in the present context means that $\|x(T)\|_\infty = \left\|A^T x_0 + R\boldsymbol{u}^*\right\|_\infty > h^*$ happens with probability at most $\varepsilon(s^*_{1000})$. In our case this becomes $\mathsf{P}\{\|x(8)\|_\infty > 0.0257\} \le 3.9\%$, i.e. $x(8)$ is in the box in Figure 5(a) with probability at least 96.1%.[7] To further illustrate this point, Figure 5(b) shows the final state reached by a new sample of 1000 simulations.

length of s^*_N	3	4	5	6	7
emp. frequency - $N = 250$	16%	32%	38%	10%	4%
emp. frequency - $N = 500$	9%	35%	40%	13%	3%
emp. frequency - $N = 1000$	5%	26%	35%	26%	8%

Table I
EMPIRICAL FREQUENCIES WITH WHICH s^*_N TOOK VALUE $3,\ldots,7$ FOR $N = 250$, $N = 500$, AND $N = 1000$.

The whole problem was then repeated 100 times, each time with a new sample of 1000 scenarios $A^{(i)}$. Different $\boldsymbol{u}^*$ were obtained, but h^* was always within the range $[0.0211, 0.0326]$, and s^*_{1000} was always between 3 and 7, resulting in $\varepsilon(s^*_{1000})$ within the interval $[0.039, 0.0591]$. We also verified whether the claim $\mathsf{P}\{\|x(8)\|_\infty > h^*\} \le \varepsilon(s^*_{1000})$ was true and this was so in all the experiments. This behavior was expected since Theorem 1 guarantees that $\mathsf{P}\{\|x(8)\|_\infty > h^*\} \le \varepsilon(s^*_{1000})$ holds true with very high confidence $1 - 10^{-6}$.

Finally, the sensitivity of s^*_N to the sample size N was tested via Monte-Carlo simulation with $N = 250, 500, 1000$. The value of s^*_N was always between 3 and 7, and Table I gives the empirical frequencies with which s^*_N took each of these values. One can notice a slight tendency to have longer support sub-samples for larger values of N. This tendency is however very moderate and the growth of N outdoes that of s^*_N so that the guarantee $\varepsilon(s^*_N)$ turns out to be systematically higher for larger values of N.

V. MISCELLANEA OF OTHER PROBLEMS

This section is meant to illustrate the generality of the theory and a selection of decision problems taken from various fields, including number theory, computer science, and geometry, is presented to which the results of this paper are applied.

A. Greatest Common Divisor and Least Common Multiple

Let $\Delta = \mathbb{N} = \{1, 2, 3, \cdots\}$, equipped with a discrete probability P. Let $\Theta = \mathbb{N}$ and, for any $\delta \in \Delta$, let Θ_δ be

[7] Notice that to rigorously obtain this result we do not have to require that our software returns the optimal solution to the problem.

the set of all the divisors of δ, that is, $\Theta_\delta = \{n \in \mathbb{N} : \ n \mid \delta\}$ where $n \mid \delta$ means that n divides δ. Consider an independent sample $(\delta^{(1)}, \cdots, \delta^{(N)})$ and construct the following scenario-based optimization problem[8]

$$\theta_N^* = \arg\max_{n \in \mathbb{N}} n$$
$$\text{subject to } n \mid \delta^{(i)} \text{ for all } i = 1, \cdots, N.$$

Its unique solution is of course the Greatest Common Divisor (GCD) of the numbers $\delta^{(1)}, \cdots, \delta^{(N)}$.

In this problem, θ_N^* violates Θ_δ if θ_N^* does not divide δ. Hence, the interpretation of the statement $\mathsf{P}^N \{\mathsf{V}(\theta_N^*) > \varepsilon(s_N^*)\} \leq \beta$ in Theorem 1 is that the probability of extracting a number not divisible by θ_N^* is with confidence $1-\beta$ less than or equal to $\varepsilon(s_N^*)$, s_N^* being the cardinality of the smallest sub-sample of $(\delta^{(1)}, \cdots, \delta^{(N)})$ having the same GCD as $(\delta^{(1)}, \cdots, \delta^{(N)})$.

Similarly, let $\Delta = \mathbb{N}$, $\Theta = \mathbb{N}$, $\Theta_\delta = \{n \in \Theta : \ \delta \mid n\}$ be the set of all the multiples of δ. The corresponding scenario-based optimization problem

$$\theta_N^* = \arg\min_{n \in \mathbb{N}} n$$
$$\text{subject to } \delta^{(i)} \mid n \text{ for all } i = 1, \cdots, N$$

yields the Least Common Multiple (LCM) of $(\delta^{(1)}, \cdots, \delta^{(N)})$ as its unique solution. Theorem 1 establishes that the probability of extracting a number which does not divide θ_N^* is with confidence $1-\beta$ less than or equal to $\varepsilon(s_N^*)$, s_N^* being the length of the smallest sub-sample of $(\delta^{(1)}, \cdots, \delta^{(N)})$ having the same LCM as $(\delta^{(1)}, \cdots, \delta^{(N)})$.

To illustrate this application we generated $N = 4000$ integers from a geometric distribution with $p = 0.85$ and obtained for the LCM problem a support sub-sample of length 12, whose elements were 23, 27, 29, 31, 32, 33, 34, 38, 39, 41, 42, 50. The corresponding LCM was $\theta_{4000}^* = 5920545668637600$. Using Theorem 1 with $\beta = 10^{-6}$ and the $\varepsilon(\cdot)$ in (7), we obtain that a further extraction will divide the LCM that has been found with probability at least $1 - \varepsilon(12) = 1 - 2.52\% = 97.48\%$.

B. Subspaces and bases

Let Δ be a vector space (not necessarily finite-dimensional), equipped with a probability. Let Θ be the set of all the linear subspaces of Δ, and, for any $\delta \in \Delta$, let $\Theta_\delta = \{\theta \in \Theta : \ \delta \in \theta\}$. Let moreover $f(\theta) = \dim \theta$, the dimension of the subspace θ. Consider now an independent random extraction $(\delta^{(1)}, \cdots, \delta^{(N)})$ and consider the following scenario-based problem

$$\theta_N^* = \arg\min_{\theta \in \{\text{subspace of } \Delta\}} \dim \theta$$
$$\text{subject to } \delta^{(i)} \in \theta \text{ for all } i = 1, \cdots, N,$$

whose unique solution is

$$\theta_N^* = \text{span}\{\delta^{(1)}, \cdots, \delta^{(N)}\}.$$

An irreducible support sub-sample for this problem is a sub-sample of $(\delta^{(1)}, \cdots, \delta^{(N)})$ whose elements form a basis for $\text{span}\{\delta^{(1)}, \cdots, \delta^{(N)}\}$, and the length of such a sub-sample is $s_N^* = \dim \theta_N^*$. Theorem 1 establishes that the probability of extracting a vector which is not a linear combination of $\delta^{(1)}, \cdots, \delta^{(N)}$ is with confidence $1-\beta$ less than or equal to $\varepsilon(\dim \theta_N^*)$.

As an example of use of this result, suppose that a linear system $\frac{dx(t)}{dt} = Ax(t) + Bu(t)$, with $x(t) \in \mathbb{R}^d$ and $u(t) \in \mathbb{R}$, is fed by a process u generated by a random source. The matrices A and B and the structure of the random generator of u are unknown. The system is initially at rest ($x(0) = 0$) and we can observe the state $x(T)$ at a final time T. Suppose that the system is operated $N = 1000$ times, where each time the input process is generated independently of the other experiments, and that 1000 final states $x^{(1)}(T), \cdots, x^{(1000)}(T)$ are recorded and the smallest subspace θ_{1000}^* of $\mathbb{R}^d$ containing all final states is computed. If θ_{1000}^* turns out to be a proper subspace of $\mathbb{R}^d$, we may think that the system is not completely reachable or that the source generating u is not sufficiently exciting. If the system is not completely reachable, future inputs u will generate final states $x(T)$ that do not explore the whole state space $\mathbb{R}^d$. In any case, irrespective of whether the system is reachable or not, we can apply the theory of this paper with a given β and claim that $x(T) \in \theta_{1000}^*$ holds with probability at least $1 - \varepsilon(\dim \theta_{1000}^*)$. For example, for $d = 300$ and $\beta = 10^{-6}$, if $\dim \theta_{1000}^* = 7$, then the claim is that $x(T) \in \theta_{1000}^*$ with probability at least 94.1%.

C. Unseen symbols of an alphabet

Let Δ be a possibly infinite, but countable, alphabet, equipped with a discrete probability. Let Θ be the set of all the finite subsets of Δ, and for any $\delta \in \Delta$ let $\Theta_\delta = \{\theta \in \Theta : \ \delta \in \theta\}$. Let moreover $f(\theta) = |\theta|$, the cardinality of θ. Given an independent random extraction $(\delta^{(1)}, \cdots, \delta^{(N)})$, the scenario-based problem is written as

$$\theta_N^* = \arg\min_{\theta \in \{\text{finite subset of } \Delta\}} |\theta|$$
$$\text{subject to } \delta^{(i)} \in \theta \text{ for all } i = 1, \cdots, N.$$

It prescribes to find the smallest subset of the alphabet that contains all the observed symbols and its unique solution is of course $\theta_N^* = \{\delta^{(1)}, \cdots, \delta^{(N)}\}$. [9] An irreducible support sub-sample of this problem is a sub-sample of $(\delta^{(1)}, \cdots, \delta^{(N)})$ containing all the elements appearing in $\{\delta^{(1)}, \cdots, \delta^{(N)}\}$ exactly once. Its length s_N^* is the number of distinct symbols observed.

The interpretation of Theorem 1 in this case is that the probability of the set of all unseen symbols is with confidence $1-\beta$ less than or equal to $\varepsilon(\text{number of already seen symbols})$.

[8] This optimization problem can be cast within the framework of Example 1 in Section I by taking $f(\theta) = -\theta$. Similarly, in the examples of Sections B, C and D we make reference to the optimization program in Example 1. Subsection E, instead, presents a decision problem that cannot be formulated in the form of Example 1.

[9] This is the set containing all the sampled symbols where a symbol that has been sampled twice or more times only appears once in the set.

This example has practical relevance in many problems in communication and other, more exotic, fields like e.g. bounding the probability of finding a new species of insect, given that s_N^* species have been observed after capturing N insects in a closed ecosystem under study.

We ran a simulation with a Poisson distribution with $\lambda = 3$ over a list of symbols, and randomly extracted $N = 1000$ symbols. The number of distinct symbols in the extraction was equal to 11. By an application of Theorem 1 with $\beta = 10^{-6}$ and using the $\varepsilon(\cdot)$ in (7) we obtain $\varepsilon(11) = 7.69\%$, which is interpreted as an upper bound to the probability of seeing a new symbol at the next extraction.

A final remark is that if one moves up from considering a countable alphabet to an uncountable one so that each symbol in the alphabet has probability zero of being drawn, then each new extraction will not coincide with a previously extracted symbol with probability 1. Hence, the violation will be equal to 1 no matter how large N is. In this case, $s_N^* = N$ and applying Theorem 1 coherently gives $\varepsilon(N) = 1$.

D. Largest substring

Let $\Delta = \Sigma^*$ be the set of all strings of finite, but otherwise arbitrary, length from a given alphabet Σ (including the empty string), equipped with a discrete probability. Let $\Theta = \Sigma^*$, Θ_δ be the set of all the substrings of δ, and $f(\theta) = -\text{length}(\theta)$. Given an independent sample of strings $(\delta^{(1)}, \cdots, \delta^{(N)})$, consider the following scenario-based problem

$$\theta_N^* = \arg\max_{\theta \in \Sigma^*} \text{length}(\theta)$$
$$\text{such that } \theta \text{ is a substring of } \delta^{(i)} \text{ for all } i = 1, \cdots, N,$$

whose solution is the largest substring common to all the strings $\delta^{(1)}, \cdots, \delta^{(N)}$. A solution always exists (possibly, it is the empty string since e.g. `ABC` and `XYZ` do not have non-empty substrings in common), but it is not necessarily unique (e.g., `ABCDEFXYZ` and `ABCUVWXYZ` have both `ABC` and `XYZ` as largest substrings). Suppose then that a lexicographical order is employed as a tie-break rule.

Theorem 1 establishes that the probability of extracting a string which does not contain θ_N^* as a substring is with confidence $1 - \beta$ less than or equal to $\varepsilon(s_N^*)$, where s_N^* is the smallest number of strings from $\delta^{(1)}, \cdots, \delta^{(N)}$ having θ_N^* as the largest substring.

As an example of practical application of this setup one can consider text analysis. Various texts of similar nature (e.g. emails, reviews) are analyzed and their common substring is determined. If for example 500 texts are analyzed and they have the largest substring θ_{500}^* in common with a minimal support sub-sample of length 11 (i.e., any group of 10 or less texts have in common a longer substring), then, by choosing $\beta = 10^{-4}$, we can claim that the probability that a future text of the same kind will contain s_{500}^* is at least $1 - \varepsilon(11) = 87.3\%$.

E. Ball coverings

For any $c \in \mathbb{R}^2$ and $r > 0$, consider the closed ball $B(c, r) = \{p \in \mathbb{R}^2 : \|p - c\| \leq r\}$. Given a finite set of points $P = \{p_1, \cdots, p_N\}$ in $\mathbb{R}^2$ and a fixed radius $r > 0$, a *centered r-ball covering* of P is a finite collection of balls $B_j = B(c_j, r)$, $j = 1, \cdots, n$, such that each ball is centered at a point in P (i.e., c_j is equal to p_i for some i), and such that $P \subset \cup_{j=1}^n B_j$. See Figure 6.

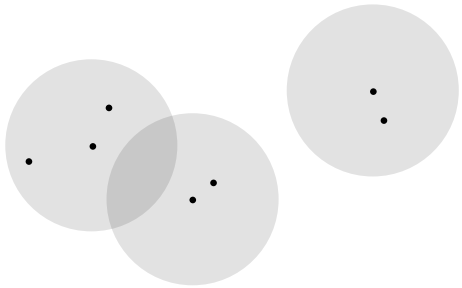


Figure 6. Centered r-ball covering.

Let now $\Delta = \mathbb{R}^2$ equipped with a probability P and let $(\delta^{(1)}, \cdots, \delta^{(N)})$ be an independent sample from Δ. For a fixed $r > 0$, consider the following problem:

find θ_N^*, which is a minimal centered r-ball covering of $\{\delta^{(1)}, \cdots, \delta^{(N)}\}$,

where *minimal* means that the number of balls of the covering is the minimum possible.[10] Since $\{B(\delta^{(i)}, r)\}_{i=1}^N$ is an admissible covering, a solution to the problem always exists. The solution, however, may not be unique (for instance, in Figure 6, the rightmost ball can be also centered in the other point contained in it to obtain another covering with the same number of balls). We decide to single out one solution by selecting the covering whose ball centers have the minimum mean distance from the origin. If P admits density, this tie-break rule isolates a single covering with probability 1.

A practical interpretation of the ball covering problem is the following. Suppose that a service provider must install n stations in order to serve N users. Each station must be maintained by a user (hence it must be located at the user's position), and every other user is served if his/her location is within a distance of r from at least one station. The overall goal is to minimize the number of stations, while the proposed tie-break rule minimizes the average distance from the provider's headquarters. Given a solution, one can find a support sub-sample and apply Theorem 1 to establish the probability of observing a new user who is not within a distance of r from the deployed stations. If for example, with 1500 users, one finds that the support sub-sample is 12, with $\beta = 10^{-6}$, one obtains $\varepsilon(12) = 5.8\%$, and the claim is that a new user is not served with probability less than 5.8%.

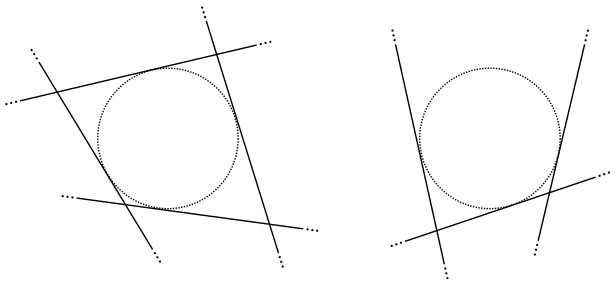
[10] Note that, due to the requirement that the balls must be centered at points taken from $(\delta^{(1)}, \cdots, \delta^{(N)})$, this problem cannot be formulated in the form of an optimization program as in Example 1 in Section I.

APPENDIX

In this appendix, we further elaborate on the discussion at the end of Section I and show that the results of [11] cannot be recovered in the context of this paper. This is done by exhibiting an example where the irreducible support sub-sample has always length 2, but the cumulative probability distribution of the violation is not dominated by a $\mathrm{Beta}(2, N-1)$ distribution as it would be if equations (2) and (3) taken from [11] were valid.

Let $\Delta = \{\delta \in \mathbb{R}^2 : \ \|\delta\| = 1\}$, equipped with the uniform probability P over the unitary circumference. Let $\Theta = \mathbb{R}^2$. For any $\delta \in \Delta$, consider the line T_δ tangent to the circumference at δ, and let Θ_δ be the closed half-plane with boundary T_δ that contains the origin (and hence the whole circumference). Let $(\delta^{(1)}, \cdots, \delta^{(N)})$, with $N \geq 2$, be an independent sample of points/tangents from P, and consider the following problem:

among all the points of intersection of two tangent lines, find the intersection θ_N^* that satisfies all the constraints $\theta \in \Theta_{\delta^{(i)}}$, $i = 1, \cdots, N$, and that has maximum distance from the origin.



(a) (b)

Figure 7. The feasible set: (a) polyhedron; (b) unbounded polytope.

In a typical situation, the intersection of the sets $\Theta_{\delta^{(i)}}$ is a polyhedron (see Figure 7(a)), in which case the solution of the problem is the feasible point furthest away from the center of the circumference. It may happen, however, that all the points $\delta^{(1)}, \cdots, \delta^{(N)}$ lie on the same half-circumference, so that the intersection of the sets $\Theta_{\delta^{(i)}}$ is an unbounded polytope (see Figure 7(b)). In this second case, the previous interpretation for θ_N^* is not valid any more, and for this reason this problem cannot be reformulated as an optimization program in the form of Example 1 in Section I.

A peculiarity of this problem is that $s_N^* = 2$ with probability 1. As a matter of fact, it is immediate to recognize that there is a unique irreducible support sub-sample, given by the two observations $(\delta^{(i_1)}, \delta^{(i_2)})$ corresponding to the two tangent lines passing through the solution θ_N^*.

Then, one may be tempted to believe that equations (2) and (3) with $d = 2$ hold true for the problem at hand. After all, the only assumption required in [11] within the context of scenario convex optimization to prove (2) and (3) with $d = 2$ is that $s_N^* \leq 2$. This result is however wrong as shown in Figure 8, where, for $N = 10$, $\mathsf{P}^N\left\{\mathsf{V}(\theta_N^*) > \varepsilon\right\}$ for the problem at hand is plotted[11] and compared with $\sum_{i=0}^{1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i}$ for $N = 10$, which is the dominating distribution in (2) and (3). It can be seen from the figure that, for a given ε, the

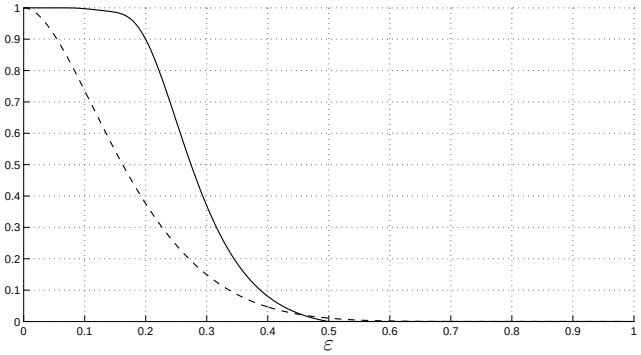


Figure 8. $\mathsf{P}^N\left\{\mathsf{V}(\theta_N^*) > \varepsilon\right\}$ for the problem at hand (solid line) vs. $\sum_{i=0}^{1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i}$ (dotted line) for $N = 10$.

probability that $\mathsf{V}(\theta_N^*) > \varepsilon$ is for the problem at hand larger than that given by (2) and (3). Hence, within the general setup of this paper, results as strong as those in [11] cannot be obtained. The very reason for this is that in the present example one condition is missing which is instead always satisfied in convex optimization (and, indeed, even in optimization without convexity conditions): what fails to be true is that adding a satisfied constraint may result here in a change of the solution (in Figure 7(b) this is e.g. the case if a tangent with slope high enough is added at the top of the circle), while this is instead not possible when θ_N^* is the solution to an optimization program.

REFERENCES

[1] T. Alamo, R. Tempo, and E.F. Camacho. Randomized strategies for probabilistic solutions of uncertain feasibility and optimization problems. *IEEE Transactions on Automatic Control*, 54(11):2545–2559, 2009.

[2] T. Alamo, R. Tempo, A. Luque, and D. R. Ramirez. Randomized methods for design of uncertain systems: sample complexity and sequential algorithms. *Automatica*, 51:160–172, 2015.

[3] G. Averkov and R. Weismantel. Transversal numbers over subsets of linear spaces. *Advances in Geometry*, 12:19–28, 2012.

[4] A. Ben-Tal and A. Nemirovski. On safe tractable approximations of chance-constrained linear matrix inequalities. *Mathematics of Operations Research*, 34(1):1–25, 2009.

[5] G.C. Calafiore and M.C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.

[6] G.C. Calafiore and M.C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

[7] G.C. Calafiore, D. Lyons, and L. Fagiano. On mixed-integer random convex programs. In *Proceedings of the 51st IEEE Conference on Decision and Control*, Maui, USA, 2012.

[8] M. C. Campi and S. Garatti. Wait-and-judge scenario optimization. *Mathematical Programming*, 167(1):155–189, 2018.

[9] M.C. Campi, C.G. Calafiore, and S. Garatti. Interval predictor models: identification and reliability. *Automatica*, 45(2):382–392, 2009.

[10] M.C. Campi and A. Carè. Random convex programs with L1-regularization: sparsity and generalization. *SIAM Journal on Control and Optimization*, 51(5):3532–3557, 2013.

[11] M.C. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008.

[11] $\mathsf{P}^N\left\{\mathsf{V}(\theta_N^*) > \varepsilon\right\}$ was obtained via Monte Carlo simulations over 10^7 runs.

[12] M.C. Campi and S. Garatti. A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. *Journal on Optimization Theory and Application*, 148(2):257–280, 2011.

[13] M.C. Campi, S. Garatti, and M. Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149 – 157, 2009.

[14] M.C. Campi, S. Garatti, and F.A. Ramponi. Non-convex scenario optimization with application to system identification. In *Proceedings of the 54th IEEE Conference on Decision and Control*, Osaka, Japan, 2015.

[15] A. Carè, S. Garatti, and M.C. Campi. Fast – fast algorithm for the scenario technique. *Operations Research*, 62(3):662–671, 2014.

[16] A. Carè, S. Garatti, and M.C. Campi. Scenario min-max optimization and the risk of empirical costs. *SIAM Journal on Optimization*, 25(4):2061–2080, 2015.

[17] M. Chamanbaz, F. Dabbene, R. Tempo, V. Venkataramanan, and Q. Wang. Sequential randomized algorithms for convex optimization in the presence of uncertainty. *IEEE Transactions on Automatic Control*, 2015. (published online).

[18] L.G. Crespo, D.P. Giesy, and S.P. Kenny. Interval predictor models with a formal characterization of uncertainty and reliability. In *Proceedings of the 53rd IEEE Conference on Decision and Control (CDC)*, pages 5991–5996, Los Angeles, CA, USA, 2014.

[19] D. Dentcheva. Optimization models with probabilistic constraints. In G. Calafiore and F. Dabbene, editors, *Probabilistic and Randomized Methods for Design under Uncertainty*, London, 2006. Springer-Verlag.

[20] P.M. Esfahani, T. Sutter, and J. Lygeros. Performance bounds for the scenario approach and an extension to a class of non-convex programs. *IEEE Transactions on Automatic Control*, 60(1):46–58, 2015.

[21] F. Dabbene G. Calafiore and R. Tempo. A survey of randomized algorithms for control synthesis and performance verification. *Journal of Complexity*, 23:301–316, 2007.

[22] S. Garatti and M.C. Campi. Modulating robustness in control design: principles and algorithms. *IEEE Control Systems Magazine*, 33(2):36–51, 2013.

[23] S. Grammatico, X. Zhang, K. Margellos, P.J. Goulart, and J. Lygeros. A scenario approach for non-convex control design. *IEEE Transactions on Automatic Control*, 61(2):334–345, 2016.

[24] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008.

[25] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

[26] J. Lofberg. Yalmip: a toolbox for modeling and optimization in matlab. In *Proceedings of the CACSD Conference*, pages 284–289, Taipei, Taiwan, 2004.

[27] J. Luedtke and S. Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19:674–699, 2008.

[28] K. Margellos, P.J. Goulart, and J. Lygeros. On the road between robust optimization and the scenario approach for chance constrained optimization problems. *IEEE Transactions on Automatic Control*, 59(8):2258–2263, 2014.

[29] K. Margellos, M. Prandini, and J. Lygeros. On the connection between compression learning and scenario based optimization. *IEEE Transactions on Automatic Control*, 60(10):2716–2721, 2015.

[30] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2006.

[31] I. R. Petersen and R. Tempo. Robust control of uncertain systems: classical results and recent developments. *Automatica*, 50:1315–1335, 2014.

[32] A. Prèkopa. *Stochastic programming*. Kluwer, Boston, MT, USA, 1995.

[33] G. Schildbach, L. Fagiano, and M. Morari. Randomized solutions to convex programs with multiple chance constraints. *SIAM Journal on Optimization*, 23(4):2479–2501, 2013.

[34] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on stochastic programming: modeling and theory*. MPS-SIAM, Philadelphia, USA, 2009.

[35] A.N. Shiryaev. *Probability*. Springer, New York, NY, USA, 1996.

[36] R. Tempo, G. Calafiore, and F. Dabbene. *Randomized Algorithms for Analysis and Control of Uncertain Systems – 2nd Edition*. Springer, London, UK, 2013.

[37] R. Tempo and H. Ishii. Monte Carlo and Las Vegas randomized algorithms for systems and control: An introduction. *European Journal of Control*, 13:189–203, 2007.

[38] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1996.

[39] P. Vayanos, D. Kuhn, and B. Rustem. A constraint sampling approach for multistage robust optimization. *Automatica*, 48(3):459–471, 2012.

[40] M. Vidyasagar. *A Theory of Learning and Generalization*. Springer-Verlag, London, 1997.

[41] M. Vidyasagar. Statistical learning theory and randomized algorithms for control. *IEEE Contr. Syst. Mag.*, 18:69–85, December 1998.

[42] M. Vidyasagar. Randomized algorithms for robust controller synthesis using statistical learning theory. *Automatica*, 37(10):1515–1528, October 2001.

[43] X. Zhang, S. Grammatico, G. Schildbach, P.J. Goulart, and J. Lygeros. On the sample size of random convex programs with structured dependence on the uncertainty. *Automatica*, 60:182–188, 2016.

Marco Claudio Campi is Professor of Automatic Control at the University of Brescia, Italy. In 1988, he received the Doctor degree in electronic engineering from the Politecnico di Milano, Milano, Italy. From 1988 to 1989, he was a Research Scientist at the Department of Electrical Engineering of the Politecnico di Milano. From 1989 to 1992, he worked as a Research Fellow at the Centro di Teoria dei Sistemi of the National Research Council (CNR) in Milano and, in 1992, he joined the University of Brescia, Brescia, Italy. He has held visiting and teaching appointments at the Australian National University, Canberra, Australia; the University of Illinois at Urbana-Champaign, USA; the Centre for Artificial Intelligence and Robotics, Bangalore, India; the University of Melbourne, Australia; the Kyoto University, Japan, Texas A&M University, USA and at the NASA, Langley Research Center, USA. Since 2011, Marco Campi is the chair of the Technical Committee IFAC on Modeling, Identification and Signal Processing (MISP). He has been in various capacities on the Editorial Board of Automatica, Systems and Control Letters and the European Journal of Control. Marco Campi is a recipient of the "Giorgio Quazza" prize, and, in 2008, he received the IEEE CSS George S. Axelby outstanding paper award for the article "The Scenario Approach to Robust Control Design". He has delivered plenary and semi-plenary addresses at major conferences including SYSID, MTNS, and CDC. Marco Campi is a distinguished lecturer of IEEE CSS, a Fellow of IEEE, a member of IFAC, and a member of SIDRA. The research interests of Marco Campi include: inductive methods, randomized algorithms, robust control, system identification, and learning theory.

Simone Garatti is Associate Professor at the Dipartimento di Elettronica ed Informazione of the Politecnico di Milano, Milan, Italy. He received the Laurea degree and the Ph.D. in Information Technology Engineering in 2000 and 2004, respectively, both from the Politecnico di Milano. From 2005 to 2015 he was assistant professor at the same university. In 2003, he was visiting scholar at the Lund University of Technology, Lund, Sweden, in 2006 at the University of California San Diego (UCSD), San Diego, CA, USA, and in 2007 at the Massachusetts Institute of Technology and the Northeastern University, Boston, MA, USA. He is member of the IEEE Technical Committee on Computational Aspects of Control System Design and of the IFAC Technical Committee on Modeling, Identification and Signal Processing. His research interests include data-based and stochastic optimization for problems in systems and control, system identification, model quality assessment, and uncertainty quantification.

Federico Alessandro Ramponi has been Assistant Professor at the Dept. of information engineering of the University of Brescia, Italy, since 2011. He obtained the Laurea degree and the Ph.D. in Information Engineering in 2004 and 2009 respectively, both from the University of Padova, Italy. From 2009 to 2011 he has been a post-doctoral fellow at the Automatic Control Lab, ETH Zurich, Switzerland, working in collaboration with J. Lygeros's research group. His current research interests include model identification, stochastic optimization, and generalizations, extensions, and applications of the Scenario Approach.