Complexity is an effective observable to tune early stopping in scenario optimization

S. Garatti, *Member, IEEE*, A. Carè, *Member, IEEE*, and Marco C. Campi, *Fellow, IEEE*

***Abstract*— Scenario optimization is a broad scheme for data-driven decision-making in which experimental observations act as constraints on the feasible domain for the optimization variables. The probability with which the solution is not feasible for a new, out-of-sample, observation is called the "risk". Recent studies have unveiled the profound link that exists between the risk and a properly defined notion of "complexity" of the scenario solution. In the present work, we leverage these results to introduce a new scheme where the size of the sample of scenarios is iteratively tuned to the current complexity of the solution so as to eventually hit a desired level of risk. This new scheme implies a substantial saving of data as compared to previous approaches. The paper presents the new method, offers a full theoretical study and illustrates it on a control problem.**

***Index Terms*— randomized methods, optimization under uncertainties, scenario optimization.**

I. Introduction

SCENARIO OPTIMIZATION is a data-driven paradigm for decision-making in the presence of uncertainty whose prototype problem is written as:

$$\begin{aligned} \min_{x\in\mathcal{X}} \quad & c(x) \\ \text{subject to:} \quad & x \in \bigcap_{i=1,\dots,N} \mathcal{X}_{\delta_i}, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^d$ is a vector of optimization variables, $c(x)$ is a convex cost function, $\mathcal{X} \subseteq \mathbb{R}^d$ is a convex set, and $\mathcal{X}_{\delta_i}$ are convex constraint sets drawn from a family $\{\mathcal{X}_\delta\}$. The uncertainty parameter δ is modeled as a random outcome from a probability space $(\Delta, \mathcal{D}, \mathbb{P})$ and δ_i, $i = 1, \dots, N$, is a random sample of independent draws from $(\Delta, \mathcal{D}, \mathbb{P})$. The solution to (1), possibly after the use of a rule to break ties, is denoted by x^*_N.

The idea behind (1) is that, in a given application, the mechanism $(\Delta, \mathcal{D}, \mathbb{P})$ by which uncertainty takes place is unknown, or only partially known, and one uses a sample of observations $\delta_1, \delta_2, \dots, \delta_N$, the "scenarios", to build a decision that is feasible for the situations that have been observed (which justifies the constraints $x \in \bigcap_{i=1,\dots,N} \mathcal{X}_{\delta_i}$ in (1)) while trying to optimize the cost function $c(x)$. This setup is quite broad and encompasses problems of various nature, ranging from systems theory and signal processing to health care and economics. In control, the scenario approach has thrived in various directions and has found application to controller design [1]–[7], identification [8]–[16], and learning [17]–[21], also fostered by theoretical advances established in many contributions [22]–[34].

After it has been designed, a scenario solution will unlikely be applied to an instance of uncertainty that has been previously seen and hence incorporated in (1). Therefore, it is essential to develop a generalization theory that enables one to draw conclusions on the probability of not satisfying new constraints, i.e., the probability that $x^*_N \notin \mathcal{X}_\delta$ for an out-of-sample $\delta \in \Delta$. Of course one would desire that such a theory provides tight results (so that it is useful in practice) while assuming as little prior knowledge as possible on the mechanism by which the δ's are generated (so that the theory is broadly applicable).

To set the stage of study, we start by formalizing the notion of risk.

Definition 1 (risk): The *risk* of a given $x \in \mathcal{X}$ is defined as $V(x) = \mathbb{P}\{\delta \in \Delta : x \notin \mathcal{X}_\delta\}$. ⋆

In the context of (1), we are interested in $V(x^*_N)$, the risk of the scenario decision x^*_N. While this quantity is not directly measurable because $\mathbb{P}$ is normally not known or only partly known, nonetheless a breakthrough result established in [35] allows one to bound the probability with which $V(x^*_N)$ is bigger than any given threshold ϵ according to formula[1]

$$\mathbb{P}^N\{V(x^*_N) > \epsilon\} \leq \sum_{i=0}^{d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}. \tag{2}$$

Equation (2) has to be read that, no matter what $\mathbb{P}$ is, the left-hand side is always upper bounded by the right-hand side, a quantity that does not depend on $\mathbb{P}$. Moreover, the result in (2) is not improvable since it is exact (i.e., $\mathbb{P}^N\{V(x^*_N) > \epsilon\} = \sum_{i=0}^{d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$) for a whole class of problems, named "fully-supported" in Definition 2.3 of [35].

Result (2) provides a quantitative tool to *a priori* determine the size N of the sample $\delta_1, \delta_2, \dots, \delta_N$ so that the solution

S. Garatti is with the Dipartimento di Elettronica, Informazione e Bioingegneria - Politecnico di Milano, piazza Leonardo da Vinci 32, 20133 Milano, Italia (e-mail: simone.garatti@polimi.it).

A. Carè and M.C. Campi are with the Dipartimento di Ingegneria dell'Informazione - Università di Brescia, via Branze 38, 25123 Brescia, Italia (e-mail: [algo.care,marco.campi]@unibs.it).

[1]Note that $V(x^*_N)$ is a random variable defined over the product probability space $(\Delta^N, \mathcal{D}^N, \mathbb{P}^N)$ that hosts $(\delta_1, \delta_2, \dots, \delta_N)$ where $\mathbb{P}^N$ is a product probability because of independence.

x_N^* has a risk no bigger than ϵ with (high) confidence level $1-\beta$. To this purpose one chooses the smallest N such that the following relation holds:

$$\sum_{i=0}^{d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq \beta. \tag{3}$$

For fully-supported problems, the value of N selected from (3) makes $V(x_N^*)$ concentrate near ϵ, see [35]. On the other hand, optimization problems encountered in practice are often not fully-supported, in which case using (3) overestimates the value of N that is necessary to secure the sought level of risk. Under these circumstances, using equation (3) is unfortunate in two respects:

(i) large samples result in a high computational burden for solving problem (1);
(ii) scenarios are often data collected from experiments (possibly, expensive or time-consuming) and are a limited and costly resource.

While item (i) strongly relates to technology and is lessened by the ever increasing computational power of electronic devices, item (ii) is more intrinsic, with significant implications in many branches of science and engineering. Examples are found in testing the clinical conditions of patients, collecting the rate-of-returns of various financial assets, or performing crash-tests in critical or unsafe operating conditions.

A. The contribution of this paper

In this paper, we focus on the key issue posed in point (ii) above and introduce a sequential scheme in which the sample size is progressively increased through iterations until a suitable halting condition is satisfied. As a result, the number of scenarios is not fixed in advance, but becomes a random variable whose value is adapted to the situation at hand by a suitable stopping rule. We show that the algorithm comes to an early stop in conditions where this suffices to warrant the desired level of risk, so providing a saving of scenarios.[2] While the stopping rule tunes the scenario scheme to the current situation, it is very important to remark that the theory of this paper is distribution-free, pretty much alike the theory in [35], i.e., it holds independently of $\mathbb{P}$: knowledge on when to stop iterating is gained through experience without resorting to any *a priori* assumption on the underlying mechanism by which scenarios are generated.

The idea of sequential randomized algorithms is not new, see e.g. [36]–[40] and the references therein, and it has been applied to scenario optimization in the contributions [41] and [42]. It is important to note, however, that the focus of these two works is point (i) above, that is, the computational complexity of the algorithm. Correspondingly, the halting condition in the schemes of [41] and [42] is based on validation: at each iteration one checks whether the current solution satisfies all (or a preset part of) the constraints associated to new scenarios that were not used to optimize. This is based on the idea that validating is computationally inexpensive and repeatedly computing the solution with a small amount of scenarios is also less computationally demanding than computing the scenario solution in one-shot. Thus, the schemes of [41] and [42] can, and often do, offer a significant computational saving. On the other hand, in the wake of (ii) one may be reluctant to use scenarios for validation and, indeed, upon counting how many scenarios are used in total, one sees that the algorithms of [41] and [42] are prone to be consuming because they can go through many validation steps. Hence, in the end, they use more scenarios than the number N prescribed by (3).

Focusing a bit closer on the sequential scheme proposed in this paper, we notice that its halting condition *is not based on validation, instead it exploits a deep connection between two concepts, risk and complexity*, that was first noticed in [43]. As a consequence, the available scenarios are all used for design purposes, so eliminating any usage of resources to validate the solution.

The present paper revisits and substantially develops an idea that was first presented by two of its authors in the conference paper [44]. Specifically, a preliminary version of the material presented in Section II and part of the material in Section III of this paper can be found in [44], while all other parts are entirely new.

B. Structure of the paper

In the next section, we introduce our sequential scheme, which culminates in Algorithm 1. Algorithm 1 articulates a general procedure but it lacks to specify the sample size that is used at each iteration to achieve the desired risk upon termination. Then, the paper proceeds along two parallel paths. Section III provides a first way to set the sample size: this approach is simple to understand and, importantly, simple to implement, but it is not fully optimized. An alternative approach that offers more scenario saving is then presented in Section IV. The practical use of the method is illustrated in Section V on a control problem in mechanics. Conclusions are drawn in Section VI.

II. The sequential scheme

First, we revise the results in [43], which provides the ground for the study in this paper, followed by a formal presentation of the new sequential scheme.

A. Some useful results from [43]

The following assumptions and definitions, borrowed from [43], are in order.

Assumption 1 (existence and uniqueness): For every N and for every sample $\delta_1, \delta_2, \ldots, \delta_N$, problem (1) admits a solution. If more than one solution exists, one solution is singled out by the application of a convex tie-break rule, which breaks the tie by minimizing an additional convex function $t_1(x)$, and, possibly, other convex functions $t_2(x)$, $t_3(x)$, ...if the tie still occurs. The so-obtained solution is denoted by x_N^*. ⋆

[2]While our sequential scheme is effective in reducing the number of scenarios (point (ii) above), there is no claim that it always offers a computational advantage.

The approach for breaking the tie in Assumption 1 is the same as that in [45]. An example of a tie-break function is the norm of x, $t_1(x) = \|x\|$. Another example is the lexicographic rule, which consists in minimizing the components of x in succession, i.e., $t_1(x) = x_1$, $t_2(x) = x_2$, etc.

Definition 2 (support constraint): A constraint $x \in \mathcal{X}_{\delta_i}$ of the scenario optimization problem (1) is called a *support constraint* if its removal (while all other constraints are maintained) yields a new solution, different from x^*_N. ⋆

Assumption 2 (non-degeneracy): For every N, the solution x^*_N to problem (1) coincides with probability 1 (with respect to the sample $\delta_1, \delta_2, \ldots, \delta_N$) with the solution that is obtained after eliminating all the constraints that are not of support. ⋆

Assumption 2 just rules out situations where the boundary of distinct constraints accumulate anomalously, and the reader is referred to [43] for more discussion on this point.

Definition 3 (complexity): The complexity s^*_N of the scenario optimization problem (1) is the number of its support constraints. ⋆

The terminology "complexity" was first introduced in [46]. It hints at the fact that the solution can be reconstructed from s^*_N scenarios and, hence, s^*_N represents a "complexity of representation" of the solution.

The complexity s^*_N depends on the random sample $\delta_1, \delta_2, \ldots, \delta_N$ and is therefore a random variable over $(\Delta^N, \mathcal{D}^N, \mathbb{P}^N)$. From results in [45], one knows that s^*_N never exceeds d, the number of optimization variables, and fully-supported problems give $s^*_N = d$ with probability 1, see [35]. Notice also that, unlike $V(x^*_N)$, s^*_N is an accessible quantity.[3]

Under the present assumptions, Theorem 2 of [43] ensures that, with probability $1-\beta$ (β is normally chosen to be a very small value, such as 10^{-6}, so that a probability of $1-\beta$ can be interpreted as "practical certainty"), it holds that

$$V(x^*_N) \leq \bar{\epsilon}(s^*_N),$$

where $\bar{\epsilon}(s^*_N)$ is equal to the empirical risk $\frac{s^*_N}{N}$ plus a small margin that bears a logarithmic dependence on β (this is the reason why the margin keeps small, and is therefore useful in practice, even when β is very small). The reader is referred to [43] for an exact expression of the function $\bar{\epsilon}(k)$; see also [47] for an asymptotic analysis on how the margin $\bar{\epsilon}(k) - \frac{k}{N}$ decays with N. Based on these quantitative results one can conclude that the two random variables $V(x^*_N)$, the risk, and s^*_N, the complexity, have a universal and strong kinship, so much so that small values of $\frac{s^*_N}{N}$ correspond to small values of $V(x^*_N)$. As we shall see, this result is the fundamental stepping stone this paper moves from.

B. Incremental scenario optimization

The result outlined in the previous section makes it possible to judge the risk without resorting to validation steps. Accordingly, the Algorithm 1 presented in this section aims at computing a solution x^* with guaranteed level of risk (i.e., such that $V(x^*) \leq 1-\epsilon$ with high confidence $1-\beta$) by considering a set of scenarios that is gradually expanded while the current level of risk is estimated from the complexity of the solution to decide when to halt the procedure. More specifically, one starts with a small number of scenarios N_0, computes the solution, and checks whether the complexity is equal to 0 (while this may sound odd, it is possible that the solution ends up on the boundary of $\mathcal{X}$ in such a way that there are no support constraints). In this case, the algorithm comes to termination and the solution is returned. Otherwise, new scenarios are added to the existing ones so as to reach a total number of N_1. A new solution is then computed and again its complexity is used to assess whether the solution meets the prescribed level of risk. Since the total number of scenarios has been increased, the less restrictive condition that $s^*_{N_1} \leq 1$ suffices to secure the desired level of risk. The procedure continues similarly by adding scenarios and verifying a progressively less restrictive condition until the condition is satisfied. This, at worst, happens at the d-th iteration owing to the theoretical result that the complexity is always less than d.

Algorithm 1 (incremental scenario optimization):

input = $N_0, N_1, \ldots, N_d$; output = x^*

0. Set $j := 0$ and $N_{-1} := 0$.
1. Collect a sample of independent scenarios $\delta_{N_{j-1}+1}, \delta_{N_{j-1}+2}, \ldots, \delta_{N_j}$, which is also independent of scenarios collected in previous steps.
2. Compute

$$x^*_{N_j} = \arg \min_{x \in \mathcal{X} \subseteq \mathbb{R}^d} \quad c(x) \qquad (4)$$
$$\text{subject to:} \quad x \in \mathcal{X}_{\delta_i},\ i = 1, 2, \ldots, N_j.$$

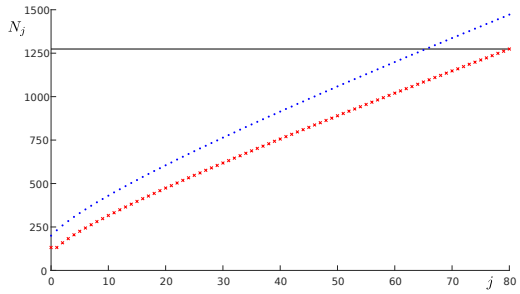
   Compute the complexity $s^*_{N_j}$ of problem (4).
3. IF $s^*_{N_j} \leq j$ THEN halt the algorithm and RETURN $x^* := x^*_{N_j}$;
   ELSE set $j := j+1$ and GOTO step 1. ⋆

While this algorithm incorporates the fundamental idea of estimating $V(x^*_{N_j})$ from $s^*_{N_j}$, it misses to indicate how $N_0, N_1, \ldots, N_d$ should be selected to guarantee the desired level of risk upon termination. Finding suitable values for $N_0, N_1, \ldots, N_d$ is the main goal of the next sections.

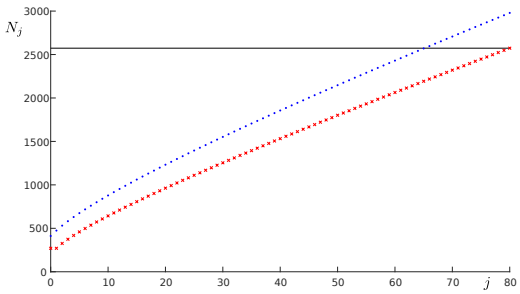
The algorithm has been named "incremental" because all scenarios used up at a given step are also maintained later down the procedure, so avoiding any waste of scenarios. Moreover, at each iteration all the scenarios are used for optimizing, with no toll paid for validation.

The actual saving of scenarios with respect to the value of N that can be selected *a priori* using equation (3) depends on the specific problem at hand. As is clear, in a fully-supported problem the algorithm systematically halts at the d-th iteration, resulting in no saving of scenarios. This is intrinsic since fully-supported problems do require the number of scenarios

[3] According to its definition, computing s^*_N requires solving N optimization problems, each of which obtained by removing one of the constraints from (1). However, support constraints must be active and, hence, the search can be restricted to active constraints only.



(a) $d = 80$, $\epsilon = 0.1$, and $\beta = 10^{-6}$.



(b) $d = 80$, $\epsilon = 0.05$, and $\beta = 10^{-6}$.

Fig. 1: N_j obtained from Theorem 1 for two values of ϵ when $d = 80$ and $\beta = 10^{-6}$ (blue dots) vs $\bar{M}_j$ (red crosses); the horizontal black solid line is the value of N computed according to (3). Note that the y-axes in the two plots have different scales.

computed from (3) to secure the desired level of risk. On the other hand, in many other problems encountered in practice the algorithm terminates at an earlier stage and offers a substantial saving of scenarios as compared to the one-shot approach.

In the next Section III, we provide a first result for computing $N_0, N_1, \ldots, N_d$ that can be established by a direct use of previous achievements from [43]. This is done in the interest of providing a first easy-to-use approach that already captures all salient aspects of the method. While this first approach is able to furnish evaluations that are satisfactory in many applications, an even tighter result can be obtained at the price of deeper theoretical investigations (and a more cumbersome implementation) as presented in the following Section IV.

III. Selection of $\mathbf{N_0, N_1, \ldots, N_d}$

To put our results in place, our discussion starts with a lemma (Lemma 1) that specifies lower limits (indicated below by the symbol $\bar{M}_j$) for N_j: owing to fundamental theoretical limits, any choice of N_j smaller than $\bar{M}_j$ cannot attain the desired result that the risk of the decision x^* returned by Algorithm 1 is no more than ϵ with confidence $1 - \beta$.

For a better understanding of the results provided below, note that N_d is the largest number of scenarios used in the algorithm so that all random variables involved in the problem can be seen as being defined over $(\Delta^{N_d}, \mathcal{D}^{N_d}, \mathbb{P}^{N_d})$: this justifies using $\mathbb{P}^{N_d}$ to compute the probability of $V(x^*) > \epsilon$.

Lemma 1: Given a "confidence parameter" $\beta \in (0, 1)$ and a "reliability parameter" $\epsilon \in (0, 1)$, let $N_0, N_1, \ldots, N_d$ be an instance of the input to Algorithm 1. Under Assumptions 1 and 2, if it holds that $\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \leq \beta$ for all $\mathbb{P}$, then it must hold that $N_j \geq \bar{M}_j$, where, for $j = 1, \ldots, d$,

$$\bar{M}_j = \min\left\{N \geq j : \sum_{i=0}^{j-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq \beta\right\}, \quad (5)$$

while $\bar{M}_0 = \bar{M}_1$. ⋆

The proof of Lemma 1 can be found in Appendix A.

Having established the lower bounds $\bar{M}_j$, $j = 0, 1, \ldots, d$, we are now in a position to state the following theorem, which provides a valid selection of $N_0, N_1, \ldots, N_d$. As we shall see after the theorem, the so-obtained values for N_j do not take any large margin from $\bar{M}_j$.

Theorem 1: Given a "confidence parameter" $\beta \in (0, 1)$ and a "reliability parameter" $\epsilon \in (0, 1)$, for any $j = 0, 1, \ldots, d$ let[4]

$$N_j = \min\left\{N \geq \bar{M}_j : \quad (6)\right.$$
$$\left.\frac{\beta}{(d+1)(\bar{M}_j+1)} \sum_{m=j}^{\bar{M}_j} \binom{m}{j} (1-\epsilon)^{m-j} \geq \binom{N}{j} (1-\epsilon)^{N-j}\right\}.$$

Under Assumptions 1 and 2, if these $N_0, N_1, \ldots, N_d$ are used as input in Algorithm 1, then it holds that $\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \leq \beta$. ⋆

The proof of Theorem 1 is given in Appendix B.

Figure 1 profiles N_j as given by (6) for two values of ϵ when $d = 80$ and $\beta = 10^{-6}$ (blue dots). In the same figure, the solid line is the value N computed from (3), which is required for the one-shot scenario problem (1) to guarantee the same level of risk as for the incremental scheme. N_j marks a gain over N for most values of j, while it is moderately bigger than N for values of j close to d (see also Section III-C for a way of improving the evaluation of N_j near d). Lower bounds $\bar{M}_j$ are also profiled as red crosses.

Remark 1: If one has a problem in dimension j, then one knows in advance that the complexity of the solution is no more than j and the value given by the red cross in Figure 1 corresponding to the abscissa j can be applied for an *a priori* determination of the number of scenarios. On the other hand, the number of scenarios given by the blue dot corresponding to j is sufficient if the corresponding solution is *a posteriori* seen to have complexity no more than j. The relatively small mismatch between the red cross and the blue dot (this mismatch is further reduced according to later evaluations provided in Section IV) goes down to the essence of the approach presented here: exploiting information collected during the operation of the method levels the possible advantage that comes from addressing simpler problems. ⋆

[4]The definition of N_j is always well-posed because $\binom{N}{j}(1-\epsilon)^{N-j} \to 0$ as $N \to \infty$.

A. Computational aspects

The value of N_j given by (6) can be easily computed by bisection according to the following procedure.

First, check whether it is true that

$$\binom{\bar{M}_j}{j}(1-\epsilon)^{\bar{M}_j-j} \leq \frac{\beta}{(d+1)(\bar{M}_j+1)} \sum_{m=j}^{\bar{M}_j} \binom{m}{j}(1-\epsilon)^{m-j}. \tag{7}$$

If yes, then set $N_j = \bar{M}_j$; otherwise, find N_j in (6) by running a bisection procedure with $\bar{M}_j$ and a high enough value for N as initial extremes.

In what follows, we provide a justification for this procedure.

First, when $j = 0$, $\binom{N}{0}(1-\epsilon)^{N-0} = (1-\epsilon)^N$ is decreasing towards 0 and takes value 1 for $N = 0$, while $\frac{\beta}{(d+1)(\bar{M}_0+1)} \sum_{m=0}^{\bar{M}_0} \binom{m}{0}(1-\epsilon)^{m-0} \leq \frac{\beta}{(d+1)(\bar{M}_0+1)} \sum_{m=0}^{\bar{M}_0}(1-0)^m = \frac{\beta}{d+1} < 1$. Hence, after checking through (7) whether $\bar{M}_0$ is already the sought solution, N_0 can be computed via bisection with $\bar{M}_0$ and a large enough value as initial extremes.

When instead $j = 1, \ldots, d$, it is easy to see that $\binom{N}{j}(1-\epsilon)^{N-j}$ as a function of N, for $N \geq j$ is first increasing and then decreasing; moreover, $\binom{N}{j}(1-\epsilon)^{N-j} \to 0$ as $N \to \infty$. Denote by $N_j^{\max}$ the value of N attaining the maximum of $\binom{N}{j}(1-\epsilon)^{N-j}$. Two cases may then arise.

CASE 1] Suppose first that $\bar{M}_j \geq N_j^{\max}$ so that $\binom{N}{j}(1-\epsilon)^{N-j}$ is decreasing for $N \geq \bar{M}_j$. Then, if (7) is satisfied, then the whole sequence $\binom{N}{j}(1-\epsilon)^{N-j}$ for $N \geq \bar{M}_j$ lies below the threshold $\frac{\beta}{(d+1)(\bar{M}_j+1)} \sum_{m=j}^{\bar{M}_j} \binom{m}{j}(1-\epsilon)^{m-j}$ and the sought value of N_j coincides with $\bar{M}_j$. Otherwise, if

$$\binom{\bar{M}_j}{j}(1-\epsilon)^{\bar{M}_j-j} > \frac{\beta}{(d+1)(\bar{M}_j+1)} \sum_{m=j}^{\bar{M}_j} \binom{m}{j}(1-\epsilon)^{m-j},$$

then the sequence $\binom{N}{j}(1-\epsilon)^{N-j}$ for $N \geq \bar{M}_j$ is initially, and for a finite number of values of N, above and then, for all the remaining values of N, below the threshold $\frac{\beta}{(d+1)(\bar{M}_j+1)} \sum_{m=j}^{\bar{M}_j} \binom{m}{j}(1-\epsilon)^{m-j}$. Hence, the sought N_j can be obtained by running the bisection algorithm with $\bar{M}_j$ and a high enough value for N as initial extremes.

CASE 2] Suppose now that $\bar{M}_j < N_j^{\max}$. In this case, it holds that

$$\begin{aligned}
&\frac{\beta}{(d+1)(\bar{M}_j+1)} \sum_{m=j}^{\bar{M}_j} \binom{m}{j}(1-\epsilon)^{m-j} \\
&\leq \frac{\beta}{(d+1)(\bar{M}_j+1)} \sum_{m=j}^{\bar{M}_j} \binom{\bar{M}_j}{j}(1-\epsilon)^{\bar{M}_j-j} \\
&\quad \text{(because } \binom{m}{j}(1-\epsilon)^{m-j} \text{ is increasing for } m \leq \bar{M}_j) \\
&\leq \binom{\bar{M}_j}{j}(1-\epsilon)^{\bar{M}_j-j} \\
&\quad \text{(because } \beta/(d+1) < 1).
\end{aligned}$$

This means that the sequence $\binom{N}{j}(1-\epsilon)^{N-j}$ has its initial value at $N = \bar{M}_j$ above the threshold $\frac{\beta}{(d+1)(\bar{M}_j+1)} \sum_{m=j}^{\bar{M}_j} \binom{m}{j}(1-\epsilon)^{m-j}$. Since $\binom{N}{j}(1-\epsilon)^{N-j}$ is first increasing and then decreasing to 0 as $N \to \infty$, this means that the sequence $\binom{N}{j}(1-\epsilon)^{N-j}$ for $N \geq \bar{M}_j$ is again initially above and then below the threshold $\frac{\beta}{(d+1)(\bar{M}_j+1)} \sum_{m=j}^{\bar{M}_j} \binom{m}{j}(1-\epsilon)^{m-j}$ and the sought N_j can still be obtained by bisection with $\bar{M}_j$ and a high enough value for N as initial extremes.

B. An explicit upper bound for N_j

The following theorem, whose proof can be found in Appendix C, provides some extra insight on the values N_j given in Theorem 1.

Theorem 2: The values N_j computed according to (6) satisfy the following inequality:

$$N_j \leq \frac{2}{\epsilon}\left[j \ln\left(\frac{2}{\epsilon}\right) + \ln\left(\frac{1}{\alpha}\right)\right] + 1, \tag{8}$$

where $\alpha = \min\{\beta, h\}$, with $h := \frac{\beta}{(d+1)(\bar{M}_j+1)} \sum_{m=j}^{\bar{M}_j} \binom{m}{j}(1-\epsilon)^{m-j}$ (note that h is the left-hand side of the inequality in (6)). ⋆

We hasten to remark that the right-hand side of (8) should not be used in place of the N_j computed according to (6) because this would result in a conservative evaluation. Instead, the importance of equation (8) is firstly that it reveals that N_j is small for small values of j, with little sensitivity to β and d (the only term depending on β and d is α, which appears under the sign of logarithm). Secondly, it allows one to make a comparison with the following upper bound derived in [6] for the value of N obtained from (3):

$$N \leq \frac{2}{\epsilon}\left[d + \ln\left(\frac{1}{\beta}\right)\right]. \tag{9}$$

Comparing the right-hand side of (8) with that of (9), one recognizes the major fact that the number d of optimization variables in (9) is replaced in formula (8) by j, the iteration step. This fact is at the basis of the conspicuous saving that is obtained in the case of early termination.

C. An alternative bound with little penalty on high complexities

Looking at Figure 1, one can notice a somehow significant increase of scenarios as compared to the *a priori* bound represented by the solid line for values of j larger than 70 (corresponding to "high complexity"), and may wonder whether this can be avoided. A simple way to do so is to simultaneously guarantee a sample size adapted to the complexity, while also keeping the original *a priori* sample size, and then pick the most favorable evaluation between the two depending on the circumstance. To achieve an exact confidence β, this approach requires to split the confidence by means of a "trade-off" parameter $c \in (0,1)$ that compromises between the two bounds: a confidence $c\beta$ is assigned to the complexity-dependent bound, while the confidence $(1-c)\beta$ is assigned to the *a priori* bound. Along this approach, one obtains the following theorem.

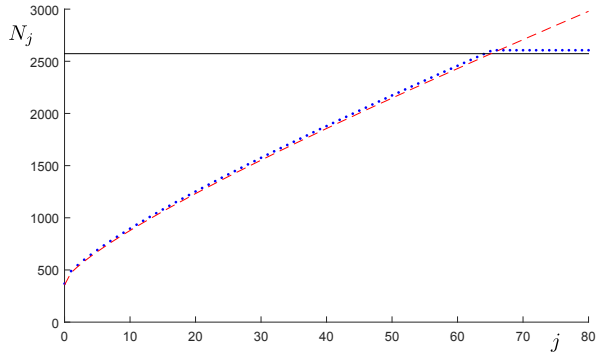


Fig. 2: N_j in this section (blue dots) vs. N_j computed according to (6) (dashed red line); the solid line is the *a priori* bound computed according to (3); $d = 80$, $\epsilon = 0.05$, and $\beta = 10^{-6}$.

Theorem 3: Given a "confidence parameter" $\beta \in (0,1)$, a "reliability parameter" $\epsilon \in (0,1)$, and a "trade-off" parameter $c \in (0,1)$, let

$$\bar{N} = \min\left\{N : \sum_{i=1}^{d-1}\binom{N}{i}\epsilon^i(1-\epsilon)^{N-i} \le (1-c)\beta\right\};$$

further, for any $j = 0, 1, \ldots, d$ let

$$\tilde{N}_j = \min\left\{N \ge \bar{M}_j : \frac{c\beta}{(d+1)(\bar{M}_j+1)}\sum_{m=j}^{\bar{M}_j}\binom{m}{j}(1-\epsilon)^{m-j} \ge \binom{N}{j}(1-\epsilon)^{N-j}\right\}.$$

Define

$$N_j = \min\{\tilde{N}_j, \bar{N}\}.$$

Under Assumptions 1 and 2, if these $N_0, N_1, \ldots, N_d$ are used as input in Algorithm 1, then it holds that $\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \le \beta$. ⋆

The proof of Theorem 3 is given in Appendix D.

We illustrate the importance of the theorem by choosing $c = \frac{1}{2}$. With this choice, the obtained N_j takes a small margin over the old N_j for values of j where $N_j = \tilde{N}_j$ (this is because β has been downsized to $\beta/2$) to the advantage of getting close to the *a priori* bound for large values of the complexity j. Figure 2 considers again the same situation as case (b) in Figure 1: the beneficial effects of the approach of this section are clearly visible for large complexity values ($j > 70$).

IV. A REFINED APPROACH TO SELECT $N_0, N_1, \ldots, N_d$

In this section we present a way to compute $N_0, N_1, \ldots, N_d$ that outdoes the result of Section III at the price of a more complicated procedure, as outlined below in Algorithm 2. One may particularly want to resort to the algorithm in this section when the gap between N_j as computed from (6) and the lower bound $\bar{M}_j$ in (5) turns out to be large, so leaving room for improvement. In Algorithm 2, $[\cdot]_+$ denotes positive part, i.e., $[x]_+ = x$ when $x \ge 0$ and $[x]_+ = 0$ otherwise.

Algorithm 2:

input = $\beta \in (0,1)$, $\epsilon \in (0,1)$; output = $N_0, N_1, \ldots, N_d$.

0. Set $\lambda_{d,d} := \frac{\beta}{\bar{M}_d+1}$ and compute
$$N'_d = \min\left\{N \ge \bar{M}_d : \lambda_{d,d}\sum_{m=d}^{\bar{M}_d}\binom{m}{d}(1-\epsilon)^{m-d} \ge \binom{N}{d}(1-\epsilon)^{N-d}\right\}.$$
1. FOR $k = d-1, d-2, \ldots, 0$ execute the following steps 1.1, 1.2, 1.3.
   1.1. Set $\lambda_{k,d} := \frac{\beta}{\bar{M}_d+1}$.
   1.2. FOR $j = d, d-1, \ldots, k+1$ execute the following steps 1.2.1, 1.2.2, 1.2.3.
   1.2.1. Set $\mu_{k,j} := \frac{\left[\binom{N'_j}{k}(1-\epsilon)^{N'_j-k} - \lambda_{k,j}\sum_{m=\bar{M}_{j-1}+1}^{\bar{M}_j}\binom{m}{k}(1-\epsilon)^{m-k}\right]_+}{\lambda_{k,j}\sum_{m=k}^{\bar{M}_{j-1}}\binom{m}{k}(1-\epsilon)^{m-k}}$.
   1.2.2. IF $\mu_{k,j} \ge 1$ THEN halt the algorithm and RETURN an error code.
   1.2.3. Set $\lambda_{k,j-1} := (1-\mu_{k,j})\lambda_{k,j}$.
   1.3. Compute
$$N'_k = \min\left\{N \ge \bar{M}_k : \lambda_{k,k}\sum_{m=k}^{\bar{M}_k}\binom{m}{k}(1-\epsilon)^{m-k} \ge \binom{N}{k}(1-\epsilon)^{N-k}\right\}.$$
2. Set $N_0 := N'_0$. FOR $k = 1, 2, \ldots, d$ IF $N'_k < N_{k-1}$ THEN set $N_k := N_{k-1}$ ELSE set $N_k := N'_k$.
3. RETURN $N_0, N_1, \ldots, N_d$. ⋆

The somewhat complicated structure of Algorithm 2 finds its motivation in the proof of Theorem 4 and the reader who wants to understand the motivation of the various steps is referred to the proof of this theorem for details.

Remark 2: Step 1.2.2 implements a termination with error. The reason for issuing an error is that, when $\mu_{k,j} \ge 1$, step 1.2.3 gives a negative value for $\lambda_{k,j-1}$ and this makes it impossible to satisfy equation (33) or equation (34) in the proof of Theorem 4. While in all trials that we have performed the algorithm has never terminated with an error, it is at present an open question whether condition $\mu_{k,j} \ge 1$ can indeed happen. Importantly, if it happens, Algorithm 2 does not help find *improved* values for $N_0, N_1, \ldots, N_d$, but one can still resort to equation (6) for a valid determination of $N_0, N_1, \ldots, N_d$.

Remark 3 (Numerical computation of N'_k in step 1.3):

Upon entering step 1.3 it holds that $\mu_{k,j} < 1$ for all $j = k, \ldots, d$, so that, given steps 1.1 and 1.2.3, one has $\lambda_{k,k} \leq \frac{\beta}{\bar{M}_d+1}$. Then, one can repeat, *mutatis mutandis*, the argument given in Section III-A to show that N'_k in step 1.3 can be always computed by a standard bisection algorithm with $\bar{M}_k$ and a higher enough value for N as extremes. We also note here for subsequent use that the same argument given in Section III-A gives that the sequence $\binom{N}{k}(1-\epsilon)^{N-k}$ for $N \geq N'_k$ is decreasing. Hence,

$$\binom{N_k}{k}(1-\epsilon)^{N_k-k} \leq \binom{N'_k}{k}(1-\epsilon)^{N'_k-k}, \qquad (10)$$

since $N_k \geq N'_k$ by definition (step 2). ⋆

The following theorem formally states the properties of the solution returned by Algorithm 1 when $N_0, N_1, \ldots, N_d$ are obtained from Algorithm 2.

Theorem 4: Given a "confidence parameter" $\beta \in (0,1)$ and a reliability parameter $\epsilon \in (0,1)$, let N_j, $j = 0, 1, \ldots, d$, be the values returned by Algorithm 2, assuming that it comes to termination without error. Under Assumptions 1 and 2, if these $N_0, N_1, \ldots, N_d$ are used as input in Algorithm 1, then it holds that $\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \leq \beta$. ⋆

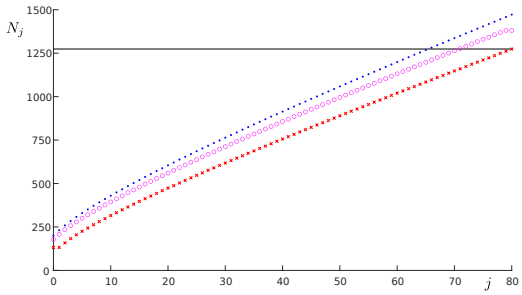
The proof of Theorem 4 is given in Appendix E.

Figure 3 depicts the N_j's returned by Algorithm 2 for the same values of d, ϵ and β considered in Figure 1. For the sake of comparison, in the plots we have also reported the values of N_j computed according to (6), besides the lower bounds $\bar{M}_j$ and the value of N computed from (3). The margin taken by $N_0, N_1, \ldots, N_d$ given by Algorithm 2 with respect to the unsurmountable lower bound given by the $\bar{M}_j$'s is about two thirds the margin taken by $N_0, N_1, \ldots, N_d$ computed from (6). It has also to be noted that the same reasoning as in Section III-C can be adopted in this context to reduce the sample complexity for values of j close to d.

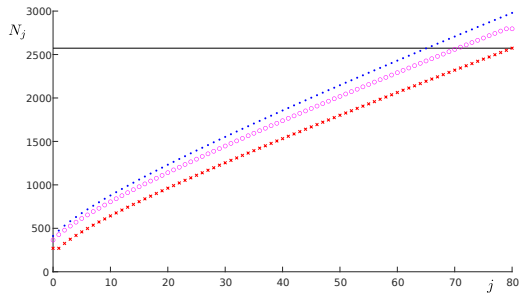
V. Control of a four-mass, four-spring system

We consider a finite-horizon optimal control problem taken from [48], which in turn was inspired by an example in [49]. The time horizon is 5s and the optimal control problem can be thought of as one step of an MPC - Model Predictive Control - procedure. The study is conducted *in silico*, which allows us to compute the actual risk of the solution (because all probabilities are known) and hence confirm the validity of the theorems; moreover, we shall be able to provide multiple runs to empirically compute the distribution of the number of employed scenarios.

The mechanical system in Figure 4 is composed by four masses and four springs. The system state is an 8-dimensional vector $\xi = [d_1\ d_2\ d_3\ d_4\ \dot{d}_1\ \dot{d}_2\ \dot{d}_3\ \dot{d}_4]^T$, where d_1, d_2, d_3 and d_4 are the displacements from the nominal positions $\bar{l}_1$, $\bar{l}_2$, $\bar{l}_3$ and $\bar{l}_4$ of the masses at equilibrium for zero input, and the dot symbol denotes time-derivative. The control input is $u = [u_1, u_2, u_3]^T$, where u_1, u_2 and u_3 are forces acting on the masses as shown in Figure 4. All masses and stiffness constants are set to value 1.



(a) $d = 80$, $\epsilon = 0.1$, and $\beta = 10^{-6}$.



(b) $d = 80$, $\epsilon = 0.05$, and $\beta = 10^{-6}$.

Fig. 3: N_j computed according to Algorithm 2 (magenta circles) vs N_j computed according to (6) (blue dots). The figure also represents $\bar{M}_j$ (red crosses) and the value of N computed according to (3) (solid black line). Note that the y-axes in the two plots have different scales.

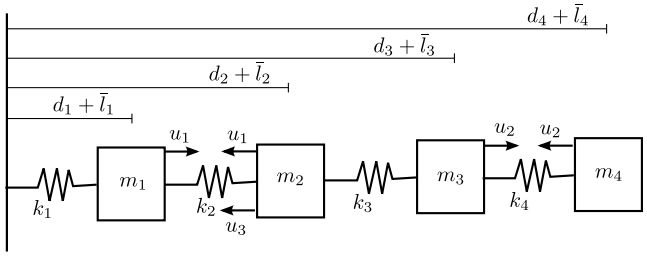


Fig. 4: Scheme of the mechanical system ($\bar{l}_1, \bar{l}_2, \bar{l}_3, \bar{l}_4$ are the nominal positions of the masses).

The control action is actuated by a ZOH - zero-order hold - and it is constant over the sampling time of 1s. The corresponding discrete-time dynamics is

$$\xi_{t+1} = A\xi_t + Bu_t + Dw_t,$$

where

$$A = \begin{bmatrix} 0.19 & 0.35 & 0.03 & 0.00 & 0.71 & 0.14 & 0.01 & 0.00 \\ 0.35 & 0.22 & 0.35 & 0.04 & 0.14 & 0.71 & 0.14 & 0.01 \\ 0.03 & 0.35 & 0.23 & 0.39 & 0.01 & 0.14 & 0.71 & 0.14 \\ 0.00 & 0.04 & 0.39 & 0.58 & 0.00 & 0.01 & 0.14 & 0.85 \\ -1.28 & 0.44 & 0.12 & 0.01 & 0.19 & 0.35 & 0.03 & 0.00 \\ 0.44 & -1.15 & 0.45 & 0.13 & 0.35 & 0.22 & 0.35 & 0.04 \\ 0.12 & 0.45 & -1.15 & 0.57 & 0.03 & 0.35 & 0.23 & 0.39 \\ 0.01 & 0.13 & 0.57 & -0.71 & 0.00 & 0.04 & 0.39 & 0.58 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.39 & 0.00 & -0.04 \\ -0.39 & 0.04 & -0.42 \\ -0.04 & 0.39 & -0.04 \\ -0.00 & -0.42 & -0.00 \\ 0.57 & 0.01 & -0.14 \\ -0.58 & 0.13 & -0.71 \\ -0.13 & 0.57 & -0.14 \\ -0.01 & -0.71 & -0.01 \end{bmatrix},$$

and Dw_t models an additional stochastic disturbance on the system with w_t a bivariate process and

$$D = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T,$$

which means that the external disturbance affects the fourth mass only. See [48] for more details on this model.

By reconstructing the disturbance from the state according to

$$w_t = D^\dagger(\xi_{t+1} - A\xi_t - Bu_t),$$

where $D^\dagger$ is the pseudo-inverse of D, the following parametrization (first proposed in [50]) of the control action as an affine function of the disturbance is adopted

$$u_t = \gamma_t + \sum_{\tau=0}^{t-1} \theta_{t,\tau} w_\tau, \tag{11}$$

where $\gamma_t \in \mathbb{R}^3$ and $\theta_{t,\tau} \in \mathbb{R}^{3\times 2}$ are optimization variables. Parametrization (11) has the advantage of making u_t and ξ_t linear functions of γ_t and $\theta_{t,\tau}$.

The system is initially at rest, that is, $\xi_0 = 0$. Assuming that N realizations of the disturbance[5] over the time horizon 0 to 4 are available, say $w_0^{(i)}, w_1^{(i)}, \dots, w_4^{(i)}$, $i = 1, \dots, N$, the control parameters γ_t and $\theta_{t,\tau}$ are designed according to the following scenario problem:

$$\min_{\substack{h_S, h_C, \gamma_t, \theta_{t,\tau} \\ t=0,\dots,4, \\ \tau=0,\dots,t-1}} h_S + h_C + 0.1\sqrt{\sum_{t=0}^{4} \|\gamma_t\|^2} + 0.1\sqrt{\sum_{t=0}^{4}\sum_{\tau=0}^{t-1} \|\theta_{t,\tau}\|^2}$$

$$\text{subject to: } \begin{cases} \begin{cases} \xi_0^{(i)} = 0 \\ \xi_{t+1}^{(i)} = A\xi_t^{(i)} + Bu_t^{(i)} + Dw_t^{(i)} \\ u_t^{(i)} = \gamma_t + \sum_{\tau=0}^{t-1} \theta_{t,\tau} w_\tau^{(i)}, \\ t = 0,1,2,3,4 \end{cases} \\ \max_{t=1,\dots,5} \|\xi_t^{(i)}\|_\infty \le h_S \\ \max_{t=1,\dots,5} \|C\xi_t^{(i)}\|_\infty \le h_C \\ \max_{t=1,\dots,5} \|u_{t-1}^{(i)}\|_\infty \le 1, \\ i = 1, \dots, N \end{cases} \tag{12}$$

where

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and $\xi_t^{(i)}$ and $u_{t-1}^{(i)}$ are the state and the control input corresponding to the i-th realization of the disturbance $w_0^{(i)}, w_1^{(i)}, \dots, w_4^{(i)}$. The interpretation of (12) is as follows: h_S is an upper bound on the displacement of the masses from their nominal position and h_C is an upper bound on the spring deformations (note that $C\xi_t^{(i)} = [d_{1,t}^{(i)} \quad d_{2,t}^{(i)} - d_{1,t}^{(i)} \quad d_{3,t}^{(i)} - d_{2,t}^{(i)} \quad d_{4,t}^{(i)} - d_{3,t}^{(i)}]^T$ are indeed spring deformations); hence, one wants to keep the masses close to their nominal position and avoid an over-deformation of the springs; the last constraint in (12) sets a saturation limit on the control action; finally, the terms under square root are regularization terms that help keep the coefficients of the control parametrization small.

Note that (12) has the same structure as the problem in (1) with $\delta_i = (w_0^{(i)}, w_1^{(i)}, \dots, w_4^{(i)})$ and the optimization variable x formed by the two parameters h_S, h_C and the coefficients that define the feedback control law, i.e., $\gamma_t, \theta_{t,\tau}$, $t = 0, \dots, 4$, $\tau = 0, \dots, t-1$. Since ξ_t and u_t depend linearly on γ_t and $\theta_{t,\tau}$, problem (12) is convex (in fact, a second-order cone program) and can be easily solved by standard software.

The total number of optimization variables that are present in (12) is $d = 77$ and, selecting $\epsilon = 5\%$ and $\beta = 10^{-6}$, an application of (3) gives $N = 2498$; this is the number of scenarios that are needed for the one-shot solution of (12) to have a risk of violating the constraints for a new realization of the disturbance below 5% with high confidence $1 - 10^{-6}$. In order to reduce the number of scenarios, Algorithm 1 was applied to problem (12) with $N_0, N_1, \dots, N_{77}$ given by Algorithm 2; since Algorithm 1 has a stochastic termination rule, for a better appreciation of the performances this test was repeated 300 times. In all trials, the solution had a risk below 5%, as expected owing to the high confidence enforced. Upon termination, the number of scenarios employed by Algorithm 1 was on average equal to 1107 and, in all runs, less than 1295. Figure 5 depicts the complete histogram of the numbers of scenarios used in the various runs, where the red line indicates the sample size needed for the one-shot approach.

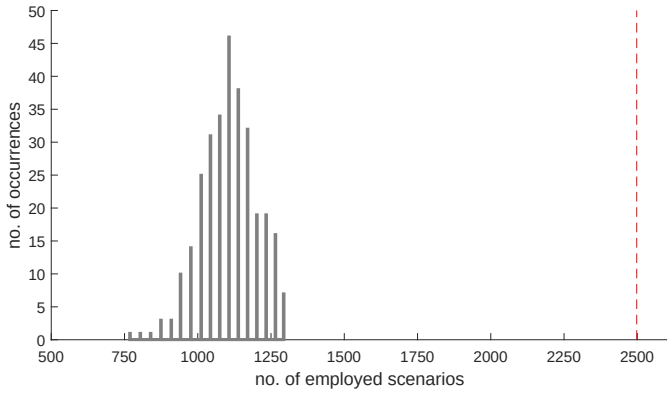


Fig. 5: Histogram of the number of scenarios used in 300 runs of Algorithm 1.

VI. Conclusions

Making decisions in an uncertain world always implies some level of risk and any scientific theory of decision-making must incorporate assessments of the probability with which shortfalls or undesired events can occur. In recent years, the scenario approach has thrived as a general methodology

[5]For reproducibility, we inform the reader that w_t is taken as a bi-variate white Gaussian noise with zero mean and covariance matrix $I_{2\times 2}$; however, the method developed in this paper does not use this information.

to address this issue where rigorous evaluations of the risk can be formulated *a posteriori* based on known elements of the decision process. The present paper moves a new fundamental step within the scenario framework by reversing this perspective: one declares a desired level of risk and the algorithms here developed drive the observational effort to meet the indicated risk level while using a minimum amount of resources. To this end, the *complexity* of the solution is used as a dependable beacon that provides indications on the incurred risk without resorting to extra validation data sets. While deep in nature, this result also furnishes tools to save data in all practical problems where data are a limited and costly resource.

APPENDIX

A. Proof of Lemma 1

To prove that any feasible choice of N_j must satisfy the condition $N_j \geq \bar{M}_j$, we start by considering a fully-supported problem in dimension j ($j \leq d$) with decision variable $x \in \mathcal{X} \subseteq \mathbb{R}^j$, cost function $\bar{c}(x)$ and scenario constraints $x \in \bar{\mathcal{X}}_{\delta_i}$, $i = 1, \ldots, N$. For any problem of this type, the risk $V(x^*_N)$ exceeds ϵ with probability $\sum_{i=0}^{j-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$ (refer to (2)); consequently, $\bar{M}_j$, by its very definition, is the minimum possible number of scenarios that is required for the fully-supported problem in dimension j to secure that the risk of its solution is no more than ϵ with confidence $1-\beta$. Next, we "embed" the fully-supported problem into an augmented d-dimensional problem defined as follows: the decision variables are $(x_1, \ldots, x_j, x_{j+1}, \ldots, x_d) \in \mathcal{X} \times [0, \infty)^{d-j}$ ($\mathcal{X}$ is as before), the cost function is defined as $\bar{c}(x_1, \ldots, x_j) + \sum_{\ell=1}^{d-j} x_{j+\ell}$ ($\bar{c}(\cdot)$ is as before), and the scenario constraints are as before, i.e., they do not constrain the new $d-j$ variables $x_{j+1}, x_{j+2}, \ldots, x_d$, which, therefore, at the optimum are set to zero to avoid an increase in the cost value. Owing to the "dummy" nature of the new variables $x_{j+1}, x_{j+2}, \ldots, x_d$, it is easy to see that: the number of support constraints of the augmented problem remains equal to j with probability one; the probability that the risk exceeds ϵ equals $\sum_{i=0}^{j-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$ as before; hence, $\bar{M}_j$ is still the minimum possible number of scenarios guaranteeing that the risk of the solution is no more than ϵ with confidence $1-\beta$. It is then clear that when the incremental scenario algorithm is applied to one of these augmented fully-supported problems, it will stop with probability one at the j-th iteration and it must be that $N_j \geq \bar{M}_j$, $j = 1, \ldots, d$, in order to obtain $V(x^*) \leq \epsilon$ with confidence $1-\beta$.

We finally show the very last assertion in the lemma which is that N_0 cannot be smaller than $\bar{M}_0 = \bar{M}_1 = \min\{N \geq 1 : (1-\epsilon)^N \leq \beta\}$. To this purpose consider the following scenario problem (as before, $x_2, \ldots, x_d$ are dummy variables and they are introduced to embed the problem in a generic dimension d):

$$\begin{aligned} \min_{x_1 \geq 0, \ldots, x_d \geq 0} \quad & \sum_{\ell=1}^{d} x_\ell \\ \text{subject to:} \quad & x_1 \geq \delta_i, \quad i = 1, \ldots, N, \end{aligned}$$

where δ is uniformly distributed over the interval $[\epsilon + \vartheta - 1, \epsilon + \vartheta]$, $0 < \vartheta \leq 1 - \epsilon$, and suppose we run Algorithm 1 for this problem. We then have (the symbol $\wedge$ denotes logical conjunction)

$$\begin{aligned} &\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \\ &\quad \geq \mathbb{P}^{N_0}\{V(x^*_{N_0}) > \epsilon \ \wedge s^*_{N_0} = 0\} \\ &\quad = \mathbb{P}^{N_0}\{V(x^*_{N_0}) > \epsilon \ \wedge x^*_{N_0} = 0\} \\ &\qquad \text{(because, with probability 1,} \\ &\qquad\ s^*_{N_0} = 0 \text{ if and only if } x^*_{N_0} = 0) \\ &\quad = \mathbb{P}^{N_0}\{x^*_{N_0} = 0\} \\ &\qquad \text{(because } V(0) = \mathbb{P}\{\delta > 0\} = \epsilon + \vartheta > \epsilon) \\ &\quad = \mathbb{P}^{N_0}\{\delta_i \leq 0, \ i = 1, \ldots, N_0\} \\ &\quad = (1 - \epsilon - \vartheta)^{N_0}. \end{aligned}$$

Hence, in order to have $\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \leq \beta$, it must be that $(1-\epsilon-\vartheta)^{N_0} \leq \beta$ and, since this must hold for all $\vartheta > 0$, we conclude that $N_0 \geq \bar{M}_0$.

Thus, altogether, it remains proven that $\bar{M}_j$ lower bounds any feasible value of N_j for every $j = 0, 1, \ldots, d$. ■

B. Proof of Theorem 1

Since Algorithm 1 stops and returns $x^* = x^*_{N_j}$ the first time that $s^*_{N_j} \leq j$, condition $V(x^*) > \epsilon$ occurs if and only if, for some j, it holds that: $s^*_{N_\ell} > \ell$ for $\ell < j$; $s^*_{N_j} \leq j$; and $V(x^*_{N_j}) > \epsilon$. Thus, we have ($\wedge$ denotes the logical conjunction):

$$\begin{aligned} &\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \\ &= \mathbb{P}^{N_0}\{s^*_{N_0} \leq 0 \wedge V(x^*_{N_0}) > \epsilon\} \\ &+ \mathbb{P}^{N_1}\{s^*_{N_0} > 0 \wedge s^*_{N_1} \leq 1 \wedge V(x^*_{N_1}) > \epsilon\} \\ &+ \ldots \\ &+ \mathbb{P}^{N_j}\{s^*_{N_0} > 0 \wedge \ldots \wedge s^*_{N_{j-1}} > j-1 \wedge s^*_{N_j} \leq j \wedge V(x^*_{N_j}) > \epsilon\} \\ &+ \ldots \\ &+ \mathbb{P}^{N_d}\{s^*_{N_0} > 0 \wedge \ldots \wedge s^*_{N_{d-1}} > d-1 \wedge s^*_{N_d} \leq d \wedge V(x^*_{N_d}) > \epsilon\} \\ &\leq \sum_{j=0}^{d} \mathbb{P}^{N_j}\{s^*_{N_j} \leq j \wedge V(x^*_{N_j}) > \epsilon\} \\ &\qquad \text{(where condition } s^*_{N_0} > 0 \wedge \ldots \wedge s^*_{N_{j-1}} > j-1 \\ &\qquad\ \text{has been suppressed)} \\ &= \sum_{j=0}^{d} \sum_{k=0}^{j} \mathbb{P}^{N_j}\{s^*_{N_j} = k \wedge V(x^*_{N_j}) > \epsilon\}. \end{aligned} \tag{13}$$

Since we want next to apply Theorem 1 in [43], we are well advised to operate a change of notation as follows: let $\bar{\epsilon}(k) = \epsilon$ for $k \leq j$ and $\bar{\epsilon}(k) = 1$ for $k > j$ and re-write the inner sum in the last expression as follows

$$\begin{aligned} &\sum_{k=0}^{j} \mathbb{P}^{N_j}\{s^*_{N_j} = k \wedge V(x^*_{N_j}) > \epsilon\} \\ &\qquad = \sum_{k=0}^{d} \mathbb{P}^{N_j}\{s^*_{N_j} = k \wedge V(x^*_{N_j}) > \bar{\epsilon}(k)\} \\ &\qquad = \mathbb{P}^{N_j}\{V(x^*_{N_j}) > \bar{\epsilon}(s^*_{N_j})\}. \end{aligned} \tag{14}$$

Now, applying Theorem 1 in [43] to the right-hand side of (14) yields

$$\mathbb{P}^{N_j}\{V(x^*_{N_j}) > \bar{\epsilon}(s^*_{N_j})\} \leq \gamma^*_j, \tag{15}$$

where γ^*_j is obtained as the solution of the following variational problem ($\mathsf{C}^d[0,1]$ is the space of continuous functions with continuous derivatives up to order d and $\frac{\mathrm{d}^0}{\mathrm{d}t^0}\xi(t)$ has to be interpreted as $\xi(t)$; moreover, $\mathbf{1}_S$ is the indicator function: $\mathbf{1}_S(t) = 1$ if $t \in S$ and $\mathbf{1}_S(t) = 0$ otherwise):

$$\gamma^*_j = \inf_{\xi(\cdot)\in\mathsf{C}^d[0,1]} \xi(1) \tag{16}$$

$$\text{subject to: } \frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\xi(t) \geq \binom{N_j}{k} t^{N_j-k}\cdot \mathbf{1}_{[0,1-\epsilon)}(t), t\in[0,1], \quad k=0,1,\ldots,j,$$

$$\frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\xi(t) \geq 0, \quad t\in[0,1], \quad k=j+1,\ldots,d.$$

Our goal is to show that, for each $j = 0,1,\ldots,d$, $\gamma^*_j \leq \frac{\beta}{d+1}$, so that, using (15) in (14) and, in turn, (14) in (13), one obtains

$$\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \leq \sum_{j=0}^{d}\gamma^*_j \leq \sum_{j=0}^{d}\frac{\beta}{d+1} = \beta, \tag{17}$$

which is the statement of the theorem.

Fix a value of $j \in \{0,1,\ldots,d\}$ and consider problem (16). We shall show that function $\bar{\xi}(t) = \frac{\beta}{(d+1)(\bar{M}_j+1)}\sum_{m=0}^{\bar{M}_j} t^m$ is feasible for (16); since

$$\bar{\xi}(1) = \frac{\beta}{(d+1)(\bar{M}_j+1)}\sum_{m=0}^{\bar{M}_j} 1 = \frac{\beta}{d+1}, \tag{18}$$

this indeed implies that the optimal value γ^*_j is no larger than $\frac{\beta}{d+1}$, which is the sought conclusion.

To show that $\bar{\xi}(t)$ is feasible, note first that

$$\frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\bar{\xi}(t) = \frac{\beta}{(d+1)(\bar{M}_j+1)}\sum_{m=k}^{\bar{M}_j}\binom{m}{k}t^{m-k},$$

which is non-negative in $[0,1]$ for all k, showing that all of the constraints in (16) for $k = j+1,\ldots,d$ are satisfied.

Next consider the constraint for $k = j$, whose validity can be checked by evaluating for which t the inequality

$$\frac{1}{j!}\frac{\mathrm{d}^j}{\mathrm{d}t^j}\bar{\xi}(t) = \frac{\beta}{(d+1)(\bar{M}_j+1)}\sum_{m=j}^{\bar{M}_j}\binom{m}{j}t^{m-j} \geq \binom{N_j}{j}t^{N_j-j} \tag{19}$$

holds true. Given the definition of N_j in (6), we have that

$$\frac{\beta}{(d+1)(\bar{M}_j+1)}\sum_{m=j}^{\bar{M}_j}\binom{m}{j}(1-\epsilon)^{m-j} \geq \binom{N_j}{j}(1-\epsilon)^{N_j-j},$$

and, applying Lemma 2 in Appendix F with $\alpha_m = \frac{\beta}{(d+1)(\bar{M}_j+1)}$ for all m, we obtain that the inequality (19) is verified over the interval $[0,1-\epsilon)$. That is, the constraint for $k = j$ in (16) is satisfied.

The validity of the constraints in (16) for $k = j-1, j-2,\ldots,0$ is now checked by induction. Recalling the definition of $\bar{\xi}(t)$, we have that

$$\begin{aligned}
&\frac{1}{(k-1)!}\frac{\mathrm{d}^{k-1}}{\mathrm{d}t^{k-1}}\bar{\xi}(t)\\
&\quad= k\cdot\left[\frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\bar{\xi}(0) + \int_0^t \frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\bar{\xi}(\tau)\mathrm{d}\tau\right]\\
&\quad\geq k\cdot\int_0^t \frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\bar{\xi}(\tau)\mathrm{d}\tau\\
&\qquad\text{(where term } \frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\bar{\xi}(0)\text{, which is non-negative,}\\
&\qquad\text{has been dropped)}\\
&\quad\geq k\cdot\int_0^t \binom{N_j}{k}\tau^{N_j-k}\cdot\mathbf{1}_{[0,1-\epsilon)}(\tau)\mathrm{d}\tau\\
&\qquad\text{(where we have used the inductive}\\
&\qquad\text{assumption valid for } k)\\
&\quad\geq \binom{N_j}{k-1}t^{N_j-k+1}\cdot\mathbf{1}_{[0,1-\epsilon)}(t),
\end{aligned}$$

which closes the induction step from k to $k-1$.

This concludes the proof. ■

C. Proof of Theorem 2

Let $\bar{N}$ be any integer such that

$$\bar{N} \geq \frac{2}{\epsilon}\left[j\ln\left(\frac{2}{\epsilon}\right) + \ln\left(\frac{1}{\alpha}\right)\right], \tag{20}$$

which is equivalent to

$$\alpha \geq \left(\frac{2}{\epsilon}\right)^j \mathrm{e}^{-\frac{\epsilon}{2}\bar{N}}. \tag{21}$$

Note that

$$\begin{aligned}
&\left(\frac{2}{\epsilon}\right)^j \mathrm{e}^{-\frac{\epsilon}{2}\bar{N}}\\
&\quad\geq \left(\frac{2}{\epsilon}\right)^j\left(1-\frac{\epsilon}{2}\right)^{\bar{N}}\\
&\qquad\text{(because } 1-\frac{\epsilon}{2} \text{ is the tangent at } \epsilon=0 \text{ of } \mathrm{e}^{-\frac{\epsilon}{2}}\text{)}\\
&\quad= \left(\frac{2}{\epsilon}\right)^j\sum_{r=0}^{\bar{N}}\binom{\bar{N}}{r}\left(\frac{\epsilon}{2}\right)^r(1-\epsilon)^{\bar{N}-r}\\
&\quad\geq \binom{\bar{N}}{j}(1-\epsilon)^{\bar{N}-j},
\end{aligned}$$

which, combined with (21), gives

$$\alpha \geq \binom{\bar{N}}{j}(1-\epsilon)^{\bar{N}-j}.$$

Recalling that $h \geq \alpha$, one then obtains $h \geq \binom{\bar{N}}{j}(1-\epsilon)^{\bar{N}-j}$, which proves that $\bar{N}$ satisfies the inequality in (6). Further, $\beta \geq \alpha$ used in (20) gives

$$\bar{N} \geq \frac{2}{\epsilon}\left[j\ln\left(\frac{2}{\epsilon}\right) + \ln\left(\frac{1}{\beta}\right)\right] \geq \frac{2}{\epsilon}\left[j + \ln\left(\frac{1}{\beta}\right)\right],$$

which, in view of the result in [6], suffices for the condition $\sum_{i=0}^{j-1}\binom{\bar{N}}{i}\epsilon^i(1-\epsilon)^{\bar{N}-i} \leq \beta$ that appears in the right-hand side

of (5) to hold, so that $\bar{N} \geq M_j$.
In conclusion, any $\bar{N}$ satisfying (20) is in the set defined in the right-hand side of (6) and this proves (8). ■

D. Proof of Theorem 3

We preliminarily observe that the $\tilde{N}_j$ are computed from (6) with $c\beta$ in place of β, while $\bar{N}$ is obtained by downsizing β to $(1-c)\beta$ in (3). Let $\tilde{x}^*$ be the solution returned by Algorithm 1 with $\tilde{N}_j$ in place of N_j and denote by $\bar{x}^*$ the solution to (1) with $N = \bar{N}$. When Algorithm 1 is run with $N_j = \min\{\tilde{N}_j, \bar{N}\}$, we have that x^* is either equal to $\tilde{x}^*$ or $\bar{x}^*$. Hence,

$$V(x^*) > \epsilon \implies V(\tilde{x}^*) > \epsilon \ \text{ or } \ V(\bar{x}^*) > \epsilon.$$

This gives

$$\begin{aligned} \mathbb{P}^{N_d}\{V(x^*) > \epsilon\} &\leq \mathbb{P}^{N_d}\{V(\tilde{x}^*) > \epsilon \ \text{ or } \ V(\bar{x}^*) > \epsilon\} \\ &\leq \mathbb{P}^{N_d}\{V(\tilde{x}^*) > \epsilon\} + \mathbb{P}^{N_d}\{V(\bar{x}^*) > \epsilon\} \\ &\leq c\beta + (1-c)\beta = \beta. \end{aligned}$$

■

E. Proof of Theorem 4

Proceeding as in the first part of the proof of Theorem 1 one obtains

$$\begin{aligned} &\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \\ &\quad \leq \sum_{j=0}^{d}\sum_{k=0}^{j} \mathbb{P}^{N_j}\{s^*_{N_j} = k \wedge V(x^*_{N_j}) > \epsilon\}, \end{aligned} \tag{22}$$

which is the same as (13).

To help the reader, we first outline informally the reasoning in the proof. Similarly to the proof of Theorem 1, we want to ensure that the sum (22) is smaller than β for the chosen values of N_j; here, however, we will take into account the mutual dependencies between the $(d+1)(j+1)$ terms that appear in the sum, with the aim of obtaining a more refined evaluation of (22) than that of Theorem 1. Thus, the terms in (22) are preliminarily expressed by means of some common generalized distribution functions, which serve the role of a "representation basis" that must satisfy certain structural constraints. A tight bound to (22) is then obtained by maximizing the value of (22) subject to these constraints. While this is the conceptual backbone of the proof, the largest part of it (from equation (28) onward) consists of tackling the ensuing maximization problem by means of relaxation and dualization – see equations (28) and (29) – and then showing by a direct calculation that (22) is smaller than β.

To continue the proof, note that, for any j and k, the event $S_{j,k} := \{s^*_{N_j} = k \wedge V(x^*_{N_j}) > \epsilon\}$ can be expressed as the union of sets $A_{j,(i_1,\ldots,i_k)} := \{s^*_{N_j} = k \wedge V(x^*_{N_j}) > \epsilon \wedge$ the support constraints have indexes $(i_1,\ldots,i_k)\}$ for $(i_1,\ldots,i_k)$ ranging over all possible combinations of k indexes from $\{1,\ldots,N_j\}$.[6] Since the sets $A_{j,(i_1,\ldots,i_k)}$ are disjoint and have all equal probability because of the i.i.d. assumption, we get

$$\mathbb{P}^{N_j}\{S_{j,k}\} = \binom{N_j}{k}\mathbb{P}^{N_j}\{A_{j,(1,\ldots,k)}\}. \tag{23}$$

[6]For $k=0$, $(i_1,\ldots,i_k)$ becomes the empty list and $A_{j,(i_1,\ldots,i_k)} = S_{j,k}$.

Next, for $k = 0,1,\ldots,d$, let x^*_k be the solution to (1) when only the first k constraints are in place (the remaining $N_j - k$ are instead removed) and let s^*_k be the corresponding number of support constraints. It is an intuitive fact that $\mathbb{P}^{N_j}\{A_{j,(1,\ldots,k)}\} = \mathbb{P}^{N_j}\{s^*_k = k \wedge V(x^*_k) > \epsilon \wedge$ the constraints with indexes $(k+1,\ldots,N_j)$ are satisfied by $x^*_k\}$ (a formal proof of this fact is given in Section 5.1.1 of [43]). Since the conditional probability that the constraints with indexes $k+1, k+2, \ldots, N_j$ are satisfied under the condition that $V(x^*_k) = v$ is $(1-v)^{N_j-k}$, by introducing the functions

$$F_k(v) = \mathbb{P}^k\{V(x^*_k) \leq v \ \wedge \ s^*_k = k\},$$

($F_k(v)$ are generalized distribution functions, see [43]) we can write

$$\mathbb{P}^{N_j}\{A_{j,(1,\ldots,k)}\} = \int_{(\epsilon,1]} (1-v)^{N_j-k}\mathrm{d}F_k(v). \tag{24}$$

Putting (24), (23) and (22) together, we now have

$$\begin{aligned} &\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \\ &\quad \leq \sum_{j=0}^{d}\sum_{k=0}^{j}\binom{N_j}{k}\int_{(\epsilon,1]}(1-v)^{N_j-k}\mathrm{d}F_k(v) \\ &\quad = \sum_{k=0}^{d}\sum_{j=k}^{d}\binom{N_j}{k}\int_{(\epsilon,1]}(1-v)^{N_j-k}\mathrm{d}F_k(v) \\ &\qquad \text{(where we have changed the order in which} \\ &\qquad \text{summations are performed)} \\ &\quad = \sum_{k=0}^{d}\int_{(\epsilon,1]}\sum_{j=k}^{d}\binom{N_j}{k}(1-v)^{N_j-k}\mathrm{d}F_k(v). \end{aligned} \tag{25}$$

The generalized distribution functions $F_k(v)$ depend on the problem at hand. Nevertheless, for any problem, they must obey for any $m = 0,1,2,\ldots$ the following equation

$$\sum_{k=0}^{\min\{m,d\}}\binom{m}{k}\int_{[0,1]}(1-v)^{m-k}\mathrm{d}F_k(v) = 1. \tag{26}$$

It is indeed easy to show that the left-hand side of (26) is equal to $\mathbb{P}^m\{V(x^*_m) \in [0,1]\}$, which is the probability of an event that is always satisfied (see [43, Equation (18)] for further details).

In view of the universal validity of (26), when establishing upper bounds to (25) we can work under the condition that equations (26) hold for all m. In more formal terms, we write

$$\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \leq \gamma,$$

where

$$\gamma = \sup_{F_0,F_1,\ldots,F_d} \sum_{k=0}^{d}\int_{(\epsilon,1]}\sum_{j=k}^{d}\binom{N_j}{k}(1-v)^{N_j-k}\mathrm{d}F_k(v) \tag{27}$$

$$\text{subject to: } \sum_{k=0}^{\min\{m,d\}}\binom{m}{k}\int_{[0,1]}(1-v)^{m-k}\mathrm{d}F_k(v) = 1, \quad m = 0,1,2,\ldots.$$

To proceed, we truncate the infinitely many constraints in (27) to the first $\bar{M}_d$ and let

$$\gamma_{\bar{M}_d} = \sup_{F_0,F_1,\ldots,F_d} \sum_{k=0}^{d} \int_{(\epsilon,1]} \sum_{j=k}^{d} \binom{N_j}{k} (1-v)^{N_j-k} \mathrm{d}F_k(v) \quad (28)$$

$$\text{subject to:} \quad \sum_{k=0}^{\min\{m,d\}} \binom{m}{k} \int_{[0,1]} (1-v)^{m-k} \mathrm{d}F_k(v) = 1, \quad m = 0, 1, \ldots, \bar{M}_d.$$

Since (28) is less constrained than (27), it holds that $\gamma \leq \gamma_{\bar{M}_d}$, $\forall \bar{M}_d$.

Standard manipulations further show that dual of Problem (28) is

$$\gamma^*_{\bar{M}_d} = \inf_{\xi(\cdot)\in \mathsf{P}_{\bar{M}_d}} \quad \xi(1) \quad (29)$$

$$\text{subject to: } \frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\xi(t) \geq \sum_{j=k}^{d} \binom{N_j}{k} t^{N_j-k} \cdot \mathbf{1}_{[0,1-\epsilon)}(t), t \in [0,1], \quad k = 0, 1, \ldots, d,$$

where $\frac{\mathrm{d}^0}{\mathrm{d}t^0}\xi(t)$ has to be meant as $\xi(t)$ and $\mathsf{P}_{\bar{M}_d}$ is the class of polynomials of order $\bar{M}_d$. Moreover, thanks to weak duality the optimal value $\gamma^*_{\bar{M}_d}$ of the dual problem is no smaller than the optimal value of the primal, which yields

$$\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \leq \gamma^*_{\bar{M}_d}.$$

Next, we want to upper-bound $\gamma^*_{\bar{M}_d}$. Consider the function $\bar{\xi}(t) = \frac{\beta}{\bar{M}_d+1} \sum_{m=0}^{\bar{M}_d} t^m$. If we succeed in showing that $\bar{\xi}(t)$ is feasible for problem (29), this concludes the proof because we then have

$$\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \leq \gamma^*_{\bar{M}_d} \leq \bar{\xi}(1) = \frac{\beta}{\bar{M}_d+1} \sum_{m=0}^{\bar{M}_d} 1^m = \beta. \quad (30)$$

To show the feasibility of $\bar{\xi}(t)$, note first that

$$\frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\bar{\xi}(t) = \frac{\beta}{\bar{M}_d+1} \sum_{m=k}^{\bar{M}_d} \binom{m}{k} t^{m-k}. \quad (31)$$

Let k be any number in $\{0, 1, ..., d\}$. Given the definition of $\lambda_{k,j-1}$ in step 1.2.3 of Algorithm 2, it holds that $\lambda_{k,j-1} + \mu_{k,j} \cdot \lambda_{k,j} = \lambda_{k,j}$. Moreover, $\lambda_{k,d} = \frac{\beta}{\bar{M}_d+1}$ (see step 1.1). This allows one to rewrite $\frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\bar{\xi}(t)$ in (31) as follows:

$$\begin{aligned}
&\frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\bar{\xi}(t) \\
&= \lambda_{k,d} \sum_{m=k}^{\bar{M}_d} \binom{m}{k} t^{m-k} \\
&= \lambda_{k,d} \sum_{m=k}^{\bar{M}_{d-1}} \binom{m}{k} t^{m-k} + \lambda_{k,d} \sum_{m=\bar{M}_{d-1}+1}^{\bar{M}_d} \binom{m}{k} t^{m-k} \\
&= (\lambda_{k,d-1} + \mu_{k,d} \cdot \lambda_{k,d}) \sum_{m=k}^{\bar{M}_{d-1}} \binom{m}{k} t^{m-k} \\
&\quad + \lambda_{k,d} \sum_{m=\bar{M}_{d-1}+1}^{\bar{M}_d} \binom{m}{k} t^{m-k} \\
&= \lambda_{k,d-1} \sum_{m=k}^{\bar{M}_{d-1}} \binom{m}{k} t^{m-k} \\
&\quad + \left[\mu_{k,d} \cdot \lambda_{k,d} \sum_{m=k}^{\bar{M}_{d-1}} \binom{m}{k} t^{m-k} + \lambda_{k,d} \sum_{m=\bar{M}_{d-1}+1}^{\bar{M}_d} \binom{m}{k} t^{m-k} \right] \\
&= (\lambda_{k,d-2} + \mu_{k,d-1} \cdot \lambda_{k,d-1}) \sum_{m=k}^{\bar{M}_{d-2}} \binom{m}{k} t^{m-k} \\
&\quad + \lambda_{k,d-1} \sum_{m=\bar{M}_{d-2}+1}^{\bar{M}_{d-1}} \binom{m}{k} t^{m-k} \\
&\quad + \left[\mu_{k,d} \cdot \lambda_{k,d} \sum_{m=k}^{\bar{M}_{d-1}} \binom{m}{k} t^{m-k} + \lambda_{k,d} \sum_{m=\bar{M}_{d-1}+1}^{\bar{M}_d} \binom{m}{k} t^{m-k} \right] \\
&= \cdots \\
&= \lambda_{k,k} \sum_{m=k}^{\bar{M}_k} \binom{m}{k} t^{m-k} \\
&\quad + \sum_{j=k+1}^{d} \left[\mu_{k,j} \cdot \lambda_{k,j} \sum_{m=k}^{\bar{M}_{j-1}} \binom{m}{k} t^{m-k} + \lambda_{k,j} \sum_{m=\bar{M}_{j-1}+1}^{\bar{M}_j} \binom{m}{k} t^{m-k} \right].
\end{aligned} \quad (32)$$

Given the definition of N'_k in step 1.3 of Algorithm 2, the definition of N_k in step 2, and considering that $\binom{N_k}{k}(1-\epsilon)^{N_k-k} \leq \binom{N'_k}{k}(1-\epsilon)^{N'_k-k}$ (see equation (10)), it holds that $N_k \geq \bar{M}_k$ and that

$$\lambda_{k,k} \sum_{m=k}^{\bar{M}_k} \binom{m}{k} (1-\epsilon)^{m-k} \geq \binom{N_k}{k} (1-\epsilon)^{N_k-k}.$$

Hence, an application of Lemma 2 with $\alpha_m = \lambda_{k,k}$ for all m gives that

$$\lambda_{k,k} \sum_{m=k}^{\bar{M}_k} \binom{m}{k} t^{m-k} \geq \binom{N_k}{k} t^{N_k-k} \quad (33)$$

for $0 \leq t \leq 1-\epsilon$. Similarly, for $j = k+1, \ldots, d$, by the very definition of $\mu_{k,j}$, it holds that

$$\begin{aligned}
&\mu_{k,j} \cdot \lambda_{k,j} \sum_{m=k}^{\bar{M}_{j-1}} \binom{m}{k} (1-\epsilon)^{m-k} + \lambda_{k,j} \sum_{m=\bar{M}_{j-1}+1}^{\bar{M}_j} \binom{m}{k} (1-\epsilon)^{m-k} \\
&\geq \binom{N'_j}{k} (1-\epsilon)^{N'_j-k},
\end{aligned} \quad (34)$$

which, along with the fact that $\binom{N_j}{k}(1-\epsilon)^{N_j-k} \leq \binom{N'_j}{k}(1-\epsilon)^{N'_j-k}$ (see equation (10)), gives that

$$\begin{aligned}
&\mu_{k,j} \cdot \lambda_{k,j} \sum_{m=k}^{\bar{M}_{j-1}} \binom{m}{k} (1-\epsilon)^{m-k} + \lambda_{k,j} \sum_{m=\bar{M}_{j-1}+1}^{\bar{M}_j} \binom{m}{k} (1-\epsilon)^{m-k} \\
&\geq \binom{N_j}{k} (1-\epsilon)^{N_j-k}.
\end{aligned}$$

Since $N_j \geq N'_j \geq \bar{M}_j$, applying Lemma 2 with $\alpha_m = \mu_{k,j} \cdot \lambda_{k,j}$ for $m = k, \ldots, \bar{M}_{j-1}$ and $\alpha_m = \lambda_{k,j}$ for $m = \bar{M}_{j-1} +$

$1, \ldots, \bar{M}_j$ yields that

$$\mu_{k,j} \cdot \lambda_{k,j} \sum_{m=k}^{\bar{M}_{j-1}} \binom{m}{k} t^{m-k} + \lambda_{k,j} \sum_{m=\bar{M}_{j-1}+1}^{\bar{M}_j} \binom{m}{k} t^{m-k} \geq \binom{N_j}{k} t^{N_j-k} \tag{35}$$

for $0 \leq t \leq 1-\epsilon$. Thus, using (33) and (35) to bound one by one the terms in the right-hand side of (32) finally gives

$$\frac{1}{k!} \frac{\mathrm{d}^k}{\mathrm{d}t^k} \bar{\xi}(t) \geq \sum_{j=k}^{d} \binom{N_j}{k} t^{N_j-k}$$

for $0 \leq t \leq 1-\epsilon$, $k = 0, 1, \ldots, d$. This means that $\bar{\xi}(t)$ is feasible for (29), which proves the theorem in the light of (30). ■

F. Auxiliary lemma

Lemma 2: For given integers $j \geq 0$, $\bar{M} \geq j$ and $\bar{N} \geq \bar{M}$, for given $\bar{t} \in [0, +\infty)$ and for given coefficients $\alpha_m \in [0, 1)$, $m = j, j+1, \ldots, \bar{M}$, suppose that

$$\sum_{m=j}^{\bar{M}} \alpha_m \binom{m}{j} \bar{t}^{m-j} \geq \binom{\bar{N}}{j} \bar{t}^{\bar{N}-j}. \tag{36}$$

Then, it holds that

$$\sum_{m=j}^{\bar{M}} \alpha_m \binom{m}{j} t^{m-j} \geq \binom{\bar{N}}{j} t^{\bar{N}-j} \tag{37}$$

for all $t \in [0, \bar{t}]$.

Proof: In order for (36) to hold, it cannot be that $j = \bar{M} = \bar{N}$. Hence, the ratio

$$\frac{\sum_{m=j}^{\bar{M}} \alpha_m \binom{m}{j} t^{m-j}}{\binom{\bar{N}}{j} t^{\bar{N}-j}} = \sum_{m=j}^{\bar{M}} \alpha_m \frac{\binom{m}{j}}{\binom{\bar{N}}{j}} \frac{1}{t^{\bar{N}-m}}$$

is a strictly decreasing function that tends to $+\infty$ for $t \to 0$ and that, for $t \to \infty$, tends either to $\alpha_{\bar{M}} < 1$ (if $\bar{M} = \bar{N}$) or to 0 (if $\bar{M} < \bar{N}$). This yields that inequality (37) is satisfied over an interval of the type $[0, t']$, which for sure includes $[0, \bar{t}]$ because of (36). ■

REFERENCES

[1] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, 2006.

[2] M. C. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design," *Annual Reviews in Control*, vol. 33, no. 2, pp. 149 – 157, 2009.

[3] S. Garatti and M. C. Campi, "Modulating robustness in control design: principles and algorithms," *IEEE Control Systems Magazine*, vol. 33, no. 2, pp. 36–51, 2013.

[4] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, "The scenario approach for Stochastic Model Predictive Control with bounds on closed-loop constraint violations," *Automatica*, vol. 50, no. 12, pp. 3009–3018, 2014.

[5] 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*, vol. 61, no. 2, pp. 334–345, 2016.

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

[7] H. A. Nasir, A. Carè, and E. Weyer, "A scenario-based stochastic MPC approach for problems with normal and rare operations with an application to rivers," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 4, pp. 1397–1410, 2018.

[8] J. S. Welsh and C. R. Rojas, "A scenario based approach to robust experiment design," in *Proceedings of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, 2009.

[9] M. Campi, G. Calafiore, and S. Garatti, "Interval predictor models: identification and reliability," *Automatica*, vol. 45, no. 2, pp. 382–392, 2009.

[10] J. S. Welsh and H. Kong, "Robust experiment design through randomisation with chance constraints," in *Proceedings of the 18th IFAC World Congress*, Milan, Italy, 2011.

[11] E. Patelli, M. Broggi, S. Tolo, and J. Sadeghi, "Cossan software: A multidisciplinary and collaborative software for uncertainty quantification," in *Proceedings of the 2nd ECCOMAS thematic conference on uncertainty quantification in computational sciences and engineering, UNCECOMP 2017, Rhodes Island, Greece*, 2013.

[12] L. G. Crespo, S. P. Kenny, and D. P. Giesy, "Random predictor models for rigorous uncertainty quantification," *International Journal for Uncertainty Quantification*, vol. 5, no. 5, pp. 469–489, 2015.

[13] L. G. Crespo, S. P. Kenny, D. P. Giesy, R. B. Norman, and S. Blattnig, "Application of interval predictor models to space radiation shielding," in *Proceedings of the 18th AIAA Non-Deterministic Approaches Conference*, San Diego, CA, USA, 2016.

[14] S. Garatti, M. C. Campi, and A. Carè, "On a class of interval predictor models with universal reliability," *Automatica*, vol. 110, p. 108542, 2019.

[15] L. G. Crespo, D. P. Giesy, and S. P. Kenny, "A scenario optimization approach to system identification with reliability guarantees," in *Proceedings of the Americal Control Conference*, 2019, pp. 2100–2106.

[16] C. Wang, C. Shang, F. Yang, D. Huang, and B. Yu, "Robust interval prediction model identification with a posteriori reliability guarantee," in *Proceedings of the 21th IFAC World Congress*, Berlin, Germany, 2020.

[17] M. C. Campi, "Classification with guaranteed probability of error," *Machine Learning*, vol. 80, pp. 63–84, 2010.

[18] M. C. Campi and A. Carè, "Random convex programs with l_1-regularization: sparsity and generalization," *SIAM Journal on Control and Optimization*, vol. 51, no. 5, pp. 3532–3557, 2013.

[19] K. Margellos, M. Prandini, and J. Lygeros, "On the connection between compression learning and scenario based single-stage and cascading optimization problems," *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2716–2721, 2015.

[20] A. Carè, F. A. Ramponi, and M. C. Campi, "A new classification algorithm with guaranteed sensitivity and specificity for medical applications," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 393–398, 2018.

[21] B. G. Anderson and S. Sojoudi, "Data-driven assessment of deep neural networks with random input uncertainty," 2020, https://arxiv.org/abs/2010.01171.

[22] T. Alamo, R. Tempo, and E. F. Camacho, "A randomized strategy for probabilistic solutions of uncertain feasibility and optimization problems," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2545–2559, 2009.

[23] M. C. Campi and S. Garatti, "A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality," *Journal of Optimization Theory and Applications*, vol. 148, no. 2, pp. 257–280, 2011.

[24] G. Schildbach, L. Fagiano, and M. Morari, "Randomized solutions to convex programs with multiple chance constraints," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2479–2501, 2013.

[25] A. Carè, S. Garatti, and M. C. Campi, "FAST - Fast Algorithm for the Scenario Technique," *Operations Research*, vol. 62, no. 3, pp. 662–671, 2014.

[26] 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*, vol. 59, no. 8, pp. 2258–2263, 2014.

[27] 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*, vol. 60, pp. 182–188, 2015.

[28] P. Mohajerin 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*, vol. 60, no. 1, pp. 46–58, 2015.

[29] A. Carè, S. Garatti, and M. C. Campi, "Scenario min-max optimization and the risk of empirical costs," *SIAM Journal on Optimization*, vol. 25, no. 4, pp. 2061–2080, 2015.

[30] F. A. Ramponi, "Consistency of the scenario approach," *SIAM Journal on Optimization*, vol. 28, no. 1, pp. 135–162, 2018.

[31] C. Shang and F. You, "A posteriori probabilistic bounds of convex scenario programs with validation tests," *IEEE Transactions on Automatic Control*, vol. 66, no. 9, pp. 4015–4028, 2021.

[32] M. Picallo and F. Dörfler, "Sieving out unnecessary constraints in scenario optimization with an application to power systems," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 6100–6105.

[33] L. Romao, K. Margellos, and A. Papachristodoulou, "On the exact feasibility of convex scenario programs with discarded constraints," *arXiv preprint arXiv:2003.13433*, 2020.

[34] M. Assif, D. Chatterjee, and R. Banavar, "Scenario approach for minmax optimization with emphasis on the nonconvex case: Positive results and caveats," *SIAM Journal on Optimization*, vol. 30, no. 2, pp. 1119–1143, 2020.

[35] M. Campi and S. Garatti, "The exact feasibility of randomized solutions of uncertain convex programs," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1211–1230, 2008.

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

[37] G. Bayraksan and D. P. Morton, "A sequential sampling procedure for stochastic programming," *Operations Research*, vol. 59, no. 4, pp. 898–913, 2011.

[38] G. Bayraksan and P. Pierre-Louis, "Fixed-width sequential stopping rules for a class of stochastic programs," *SIAM Journal on Optimization*, vol. 22, no. 4, pp. 1518–1548, 2012.

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

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

[41] M. Chamanbaz, F. Dabbene, R. Tempo, V. Venkataramanan, and Q.-G. Wang, "Sequential randomized algorithms for convex optimization in the presence of uncertainty," *IEEE Transactions on Automatic Control*, vol. 61, no. 9, pp. 2565–2571, 2016.

[42] G. Calafiore, "Repetitive scenario design," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1125–1137, 2017.

[43] M. C. Campi and S. Garatti, "Wait-and-judge scenario optimization," *Mathematical Programming*, vol. 167, no. 1, pp. 155–189, 2018.

[44] S. Garatti and M. C. Campi, "Complexity-based modulation of the data-set in scenario optimization," in *2019 18th European Control Conference (ECC)*, 2019, pp. 1386–1391.

[45] G. Calafiore and M. Campi, "Uncertain convex programs: randomized solutions and confidence levels," *Mathematical Programming*, vol. 102, no. 1, pp. 25–46, 2005.

[46] S. Garatti and M. C. Campi, "Risk and complexity in scenario optimization," *Mathematical Programming*, vol. 191, pp. 243–279, 2022.

[47] M. C. Campi and S. Garatti, "A theory of the risk for optimization with relaxation and its application to support vector machines," *Journal of Machine Learning Research*, vol. 22, no. 288, pp. 1–38, 2021.

[48] M. C. Campi, S. Garatti, and M. Prandini, "Scenario optimization for MPC," in *Handbook of Model Predictive Control*, S. Raković and W. Levine, Eds. Cham, Switzerland: Birkhäuser, 2019, pp. 445–463.

[49] E. Cinquemani, M. Agarwal, D. Chatterjee, and J. Lygeros, "Convexity and convex approximations of discrete-time stochastic control problems with constraints," *Automatica*, vol. 47, no. 9, pp. 2082–2087, 2011.

[50] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski, "Optimization over state feedback policies for robust control with constraints," *Automatica*, vol. 42, no. 4, pp. 523–533, 2006.

Simone Garatti (M'13) 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. In 2003, he held a visiting position at the Lund University of Technology, in 2006 at the University of California San Diego (UCSD), in 2007 at the Massachusetts Institute of Technology (MIT), and in 2019 at the University of Oxford. In 2021 Simone Garatti was keynote speaker at the IEEE 3rd Conference on Norbert Wiener in the 21st Century. From 2013 to 2019 he was member of the EUCA Conference Editorial Board, while he is currently associate editor of the International Journal of Adaptive Control and Signal Processing and member of the IEEE-CSS Conference Editorial Board. He is also member of the IEEE Technical Committees on Computational Aspects of Control System Design and on System Identification and Adaptive Control, and of the IFAC Technical Committee on Modeling, Identification and Signal Processing. His research interests include data-driven optimization and decision-making, stochastic optimization for problems in systems and control, system identification, model quality assessment, and uncertainty quantification.

Algo Carè (M'19) received the Ph.D. degree in informatics and automation engineering in 2013 from the University of Brescia, Italy, where he is currently a Research Fellow with the Department of Information Engineering. After his Ph.D., he spent two years at the University of Melbourne, VIC, Australia, as a Research Fellow in system identification with the Department of Electrical and Electronic Engineering. In 2016, he was a recipient of a two-year ERCIM Fellowship that he spent at the Institute for Computer Science and Control (SZTAKI), Hungarian Academy of Sciences (MTA), Budapest, Hungary, and at the Multiscale Dynamics Group, National Research Institute for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands. He received the triennial Stochastic Programming Student Paper Prize by the Stochastic Programming Society for the period 2013–2016. He is member of the IFAC Technical Committee on Modeling, Identification and Signal Processing, of the IEEE Technical Committee on Systems Identification and Adaptive Control, and of the EUCA Conference Editorial Board. His current research interests include data-driven decision methods, system identification, and learning theory.

Marco Claudio Campi (F'12) is Professor of Automatic Control at the *University of 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; the *Texas A&M University*, USA; the *NASA Langley Research Center*, Hampton, Virginia. USA. Marco Campi was till recently the chair of the *Technical Committee IFAC on Modeling, Identification and Signal Processing (MISP)*, and has been in various capacities on the Editorial Board of *Automatica*, *Systems and Control Letters* and the *European Journal of Control*. He 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. Currently he is a *distinguished lecturer of the Control Systems Society*. Marco Campi is a Fellow of IEEE, a Fellow of IFAC. The research interests of Marco Campi include: *data-driven decision making*, *learning*, *system identification*,*stochastic systems*.