2019 18th European Control Conference (ECC)
Napoli, Italy, June 25-28, 2019

Complexity-based modulation of the data-set in scenario optimization

Simone Garatti and Marco C. Campi

Abstract— **The scenario approach is a broad methodology for data-driven optimization that has found numerous applications in systems and control design. It consists in making a decision that is optimal with respect to a given criterion, while also being consistent with a sample of observations that are called the "scenarios". More precisely, each scenario corresponds to a constraint and the solution is sought in the domain of feasibility of all scenario constraints. The level of robustness of the scenario solution is quantified by the "risk", which is the probability that the scenario solution is not consistent with a new, out-of-sample, scenario. Recent studies have unveiled a profound link between the risk and the complexity of the solution (defined as the minimum amount of scenarios that is needed to reconstruct the solution). In this work, we leverage these results to introduce a new learning scheme where the size of the scenario sample is iteratively learned during optimization as a function of the complexity of the current solution. This new scheme implies a better exploitation of the information, so that one achieves a prescribed level of risk while saving many data as compared to standard scenario schemes. This paper presents the theoretical study that proves this result and illustrates it through a numerical example.**

I. Introduction

Scenario optimization is a data-driven paradigm for decision-making in the presence of uncertainty. The prototype scenario optimization problem is as follows:

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

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 instances of convex constraint sets from a family $\{\mathcal{X}_\delta\}$ parameterized by the uncertainty parameter δ. Parameter δ is modeled as a random element from a probability space $(\Delta, \mathcal{D}, \mathbb{P})$ and δ_i, $i = 1, \dots, N$, is an independent random sample of δ values. The solution to (1) is called "scenario solution" and is denoted by x^*_N.

The idea underlying (1) is that $(\Delta, \mathcal{D}, \mathbb{P})$, which represents the mechanism through which uncertainty is generated, is unknown to the user. The only available source of knowledge on uncertainty is given by $\delta_1, \delta_2, \dots, \delta_N$, which are referred to as the "scenarios" and have to be thought of as data retrieved from experiments. The scenario solution x^*_N is the minimizer of the cost function $c(x)$ within the domain of x values that are feasible for the scenarios at hand.

The paradigm introduced by (1) is very general and it encompasses problems from various fields. In fact, since its introduction in the seminal paper [6] the scenario approach has obtained increasing recognition and, in systems and control, instances of (1) have found application in various methods for control system design [7], [14], [21], [26], [22], [2], system identification [30], [9], [29], [19], and machine learning [8], [10], [24], [17]. Moreover, the theoretical development of the approach has benefitted from many contributions along various directions, [1], [12], [27], [15], [23], [31], [16].

When applied in practice, the scenario solution will face new instances of uncertainty, different from those used in (1), and a fundamental question for its use is how guaranteed x^*_N is in relation to the satisfaction of the constraints $x \in \mathcal{X}_\delta$ for out-of-sample instances of $\delta \in \Delta$. The following notion of risk formalizes this idea.

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

It is important to remark that the risk cannot be directly computed because it depends on the probability $\mathbb{P}$, which is unknown. The studies in [6], [7], [11] have pioneered a theory that allows one to link the sample size N to $V(x^*_N)$. Specifically, the risk of the scenario solution $V(x^*_N)$ is a random variable defined over the product[1] probability space $(\Delta^N, \mathcal{D}^N, \mathbb{P}^N)$ because of the dependence of x^*_N on the random sample $\delta_1, \dots, \delta_N$ and the result of [11] states that $V(x^*_N)$ is bigger than ϵ with a probability which is always upper-bounded by a Beta distribution according to the following formula:

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

Bound (2) is valid for any scenario optimization problem in the form of (1) and for any $\mathbb{P}$; interestingly enough, this result 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 the class of "fully-supported" problems (see Definition 2.3 in [11]).

The result in (2) provides a quantitative tool to properly assess the size N of the sample $\delta_1, \delta_2, \dots, \delta_N$ so as to

S. Garatti is with the Dip. di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, piazza L. da Vinci 32, 20133 Milan, Italia. simone.garatti@polimi.it

M.C. Campi is with the Dip. di Ingegneria dell'Informazione, Università di Brescia, via Branze 38, 25123 Brescia, Italia marco.campi@unibs.it

The work of M.C. Campi was supported by the H&W program of the University of Brescia under the project "Classificazione della fibrillazione ventricolare a supporto della decisione terapeutica - CLAFITE".

[1] It is a product probability space because of the independence of scenarios.

guarantee that the risk x_N^* is no bigger than a user-chosen value ϵ with a suitable confidence level $1-\beta$. To this purpose, it is enough to choose the smallest N such that

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

The above assessment of N comes with a drawback though: it is based on a worst-case evaluation (only tight for fully-supported problems) and, as such, can be conservative in various cases. Indeed, optimization problems encountered in practice are often not fully-supported and some post-analysis reveals that the actual risk is much smaller than the prescribed threshold ϵ, indicating that less scenarios would have been sufficient to achieve the required level of violation $V(x_N^*) \leq \epsilon$. This is unfortunate for two reasons:

i. the bigger N, the more the computational effort to solve problem (1);
ii. more importantly, when scenarios are data obtained from experiments, they can be valuable and expensive resources. Hence, over-sizing N has a cost in terms of the effort that is required to obtain the scenarios.

A. Paper contribution

The objective of this paper is that of introducing a new scheme to obtain guaranteed solutions (i.e. solutions with a prescribed level of risk) for (1) with fewer scenarios. Given the above discussion on fully-supported problems, this result cannot be achieved by solving (1) in one-shot. The idea is to instead resort to a sequential scheme where the solution is iteratively computed with a sample size that is progressively increased through iterations, until a suitable termination condition is satisfied. Hence, the number of required scenarios is not fixed beforehand and it becomes a random variable. The sequential scheme marks a positive result whenever this random variable distributes on values smaller than the N value prescribed by (3), for which the termination condition is key to achieve the goal.

The idea of sequential algorithms is not new and it appeared already in relation to randomized methods, see [20], [3], [4], [28], [25] and references therein. In the context of scenario optimization, this idea has been used in the recent contributions [18] and [5]. However, in these works the focus is on point (i) above and the intent is that of reducing the computational burden. Correspondingly, the termination condition in the sequential schemes of [18] and [5] is based on a validation rule, i.e., at each iteration one checks whether the current solution satisfies all (or a preset part of) the constraints associated to new scenarios that are not used in the optimization phase. The underlying idea is that validating a solution is computationally inexpensive and repeatedly computing a solution with a small amount of scenarios is less computationally demanding than computing the scenario solution in one-shot with all the N scenarios in place. Thus, the schemes of [18] and [5] can, and often do, offer a big computational saving. On the other hand, if these algorithms are analyzed from the perspective of point (ii), one may wonder why a valuable resource such as a scenario should not be employed for design purposes and it is instead just used for validation. Moreover, upon counting how many scenarios are used in total, the algorithms of [18] and [5] can be very demanding, because they can go through many validation phases resulting in a total number of scenarios that is even bigger than the N prescribed by (3).

In this paper, we specifically focus on point (ii) above and we leverage the recent results of [13] to propose a sequential algorithm whose termination condition is not based on validation, but rather on a notion of complexity of the scenario solution. In this way, at each iteration, the available scenarios are all used to optimize, without any waste of resources. Moreover, a mathematical analysis is also developed to properly size the number of scenarios that are required at each iteration in order to minimize the usage of scenarios while ensuring that the risk of the solution is below threshold ϵ with high confidence $1-\beta$. In the end, the total number of scenarios that are used is typically smaller than the N prescribed by (3), and in some cases it is much smaller. Hence, the proposed algorithm offers an attractive way to find guaranteed solutions with a (possibly large) saving of data with respect to the one-shot computation of x_N^* in (1).

B. Structure of the paper

The new sequential algorithm is presented in Section II, while its theoretical analysis is given in Section III. The proofs are in Section IV, while a numerical example in Section V concludes the paper.

II. Incremental scenario optimization

We start by introducing an assumption of existence and uniqueness of the scenario solution.

Assumption 1 (existence and uniqueness): For every N and for every sample δ_i, $i = 1, \ldots, N$, program (1) admits 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^*. □

The approach for breaking the tie in Assumption 1 is the same as in [6]. An example of a tie-break function is the norm of x, $t_1(x) = \|x\|$, which alone always suffices to break the tie. Another example is the lexicographic rule, which consists in minimizing the components of x in succession, i.e. $t_1(x) = x_1$, $t_2(x) = x_2$, ..., $t_d(x) = x_d$.

We now briefly revisit the result of [13], which provides the fundamental framework to build the new sequential algorithm. Paper [13] has established that in all scenario problems there exists a profound link between risk and complexity, where the notion of complexity is the following.

Definition 2 (support constraint and complexity): 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) changes the solution x_N^*. The

complexity s_N^* of the scenario optimization problem (1) is the number of its support constraints. □

The complexity s_N^* depends on the random sample $\delta_1, \delta_2, \ldots, \delta_N$, and hence it is a random variable defined over $(\Delta^N, \mathcal{D}^N, \mathbb{P}^N)$ that takes value in the set $\{0, 1, \ldots, d\}$. It is indeed a well known fact that the complexity cannot be larger than d in a convex setup, [6], and fully-supported problems are those problems for which $s_N^* = d$ with probability 1. Problems that are not fully-supported instead show a varied behavior of s_N^*. Note that, differently from $V(x_N^*)$, s_N^* is accessible, which means that it can be computed according to its definition for a given sample $\delta_1, \delta_2, \ldots, \delta_N$.

In a nutshell, the result of [13] says that the random variables $V(x_N^*)$ and s_N^* have always a strong probabilistic dependence, irrespective of the problem at hand and of the probability $\mathbb{P}$. This result has an important conceptual implication: validation with new scenarios is not necessary to assess the risk of the scenario solution; rather, $V(x_N^*)$ can be judged from s_N^*, and the smaller s_N^*, the smaller $V(x_N^*)$. Going back to the goal of devising a suitable sequential algorithm whose termination condition does not depend on validation, the link between risk and complexity discussed above suggests the following sequential Algorithm 1 (in the algorithm, the sizes $N_0, N_1, \ldots, N_d$ are left undefined and their choice is discussed later in the light of the theory).

Algorithm 1 (Incremental scenario optimization): input := an increasing sequence of integers $N_0, N_1, \ldots, N_d$; output := x^*.

0. Set $j := 0$ and $N_{-1} := 0$.
1. Collect scenarios $\delta_{N_{j-1}+1}$, $\delta_{N_{j-1}+2}$ and add them to the sample of scenarios.
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. □

In Algorithm 1 the size of the scenario sample is progressively increased through iterations (that is, at the first iteration just N_0 scenarios are used; if a second iteration is needed, new scenarios are added so as to have N_1 scenarios in all; and so on and so forth). The termination test is based on an evaluation of the current solution complexity and, as iterations progress, the condition becomes less and less strict since more scenarios become available. The algorithm has been named "Incremental" because all the scenarios that are used in one iteration are then re-used in subsequent iterations, with the addition of new scenarios. Moreover, at each iteration, all the available scenarios are used to optimize.

The fundamental issue that we need to explore now is how $N_0, N_1, \ldots, N_d$ must be selected so that upon termination the algorithm returns a solution x^* whose risk is below the threshold ϵ with high confidence $1 - \beta$. The result in the next section provides values for $N_0, N_1, \ldots, N_d$ as a function of the chosen ϵ and β and rigorously characterizes the advantages of the proposed incremental approach. The technically complex proof of this result is postponed to Section IV to improve readability.

III. Sizing of $N_0, N_1, \ldots, N_d$

The analysis is conducted under the following non-degeneracy assumption introduced in [13].

Assumption 2 (non-degeneracy): For every N, the solution x_N^* to program (1) coincides with probability 1 with the solution that is obtained after eliminating all the constraints that are not of support. □

Assumption 2 is a mild condition that excludes that the constraints accumulate anomalously at the solution. See [13] for more comments on degeneracy.

Towards the goal of sizing $N_0, N_1, \ldots, N_d$, start by defining $\bar{M}_0 = 1$ and

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

for $j = 1, \ldots, d$. Note that $\sum_{\ell=0}^{j-1} \binom{N}{\ell} \epsilon^\ell (1-\epsilon)^{N-\ell}$ is the probability that $V(x_N^*) > \epsilon$ for a fully-supported problem in dimension j (i.e. with $x \in \mathbb{R}^j$), see equation (2). Note also that given any fully-supported problem in dimension j, this problem can be embedded in an augmented d-dimensional optimization domain by adding $d-j$ dummy variables so that the number of support constraints remains equal to j with probability 1 and the probability that $V(x_N^*) > \epsilon$ remains $\sum_{\ell=0}^{j-1} \binom{N}{\ell} \epsilon^\ell (1-\epsilon)^{N-\ell}$ as before. When Algorithm 1 is used for one of these augmented fully-supported problems, the algorithm will systematically stop at the j-th step and, given the interpretation of $\bar{M}_j$ defined above, it must be that $N_j \geq \bar{M}_j$ in order to obtain $V(x^*) \leq \epsilon$ with confidence $1 - \beta$. This set $\bar{M}_j$ as a lower bound to the value taken by N_j.

Next, the following theorem provides the sought quantification of $N_0, N_1, \ldots, N_d$ and is the main achievement of this section.

Theorem 1: Given a confidence parameter $\beta \in (0, 1)$ and a risk parameter $\epsilon \in (0, 1)$, let x^* be the solution returned by Algorithm 1 where N_j, $j = 0, 1, \ldots, d$, is given by:[2]

$$N_j = \min \Big\{ N : \ N \geq \bar{M}_j \quad \text{and} \qquad (5)$$
$$\binom{N}{j} (1-\epsilon)^{N-j} \leq \frac{\beta}{(d+1)(\bar{M}_j+1)} \sum_{m=j}^{\bar{M}_j} \binom{m}{j} (1-\epsilon)^{m-j} \Big\}.$$

Then, under Assumptions 1 and 2, it holds that $\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \leq \beta$. □

Proof: see Section IV.

Figures 1-4 profiles the N_j's obtained from Theorem 1 for various values of ϵ and d and for $\beta = 10^{-6}$ (blue dots).

[2]It is worth noticing that the definition of N_j is always well-posed since $\binom{N}{j}(1-\epsilon)^{N-j} \to 0$ as $N \to +\infty$. An effective way to compute N_j is via bisection.

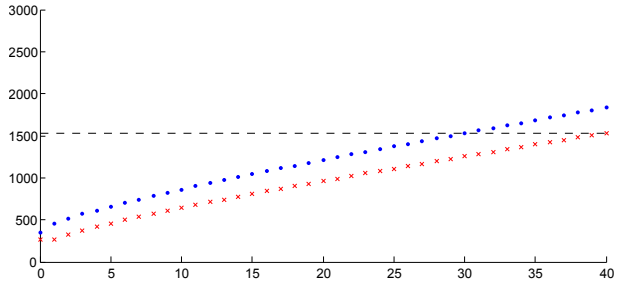

Fig. 1. N_j (blue dots) vs $\bar{M}_j$ (red crosses) and N (dashed black line) - $d = 40$, $\epsilon = 0.05$, and $\beta = 10^{-6}$

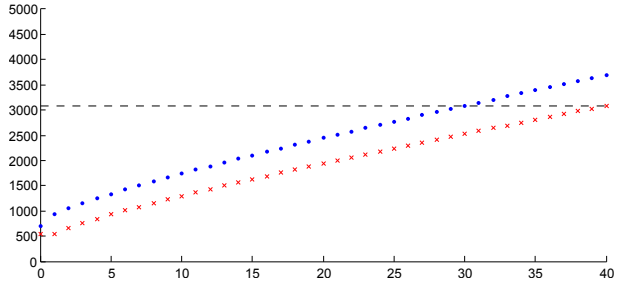

Fig. 2. N_j (blue dots) vs $\bar{M}_j$ (red crosses) and N (dashed black line) - $d = 40$, $\epsilon = 0.025$, and $\beta = 10^{-6}$

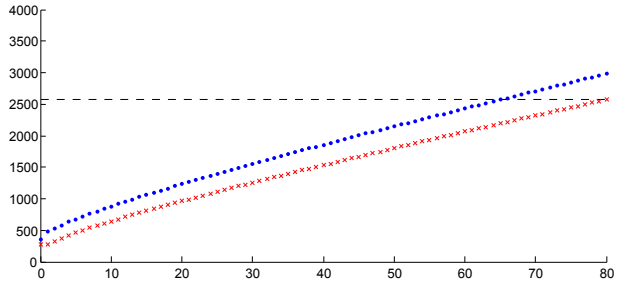

Fig. 3. N_j (blue dots) vs $\bar{M}_j$ (red crosses) and N (dashed black line) - $d = 80$, $\epsilon = 0.05$, and $\beta = 10^{-6}$

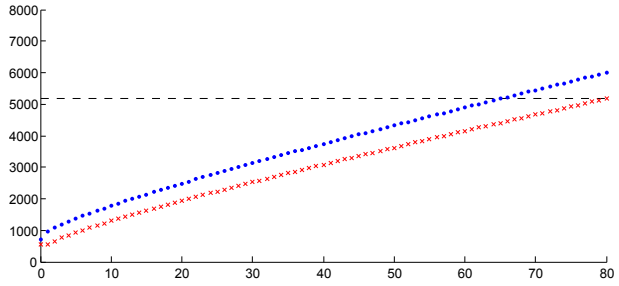

Fig. 4. N_j (blue dots) vs $\bar{M}_j$ (red crosses) and N (dashed black line) - $d = 80$, $\epsilon = 0.025$, and $\beta = 10^{-6}$

The plots also display through a black dashed line the N computed from (3), which is the size required for the one-shot solution of the scenario program (1) to attain the same risk level ϵ with confidence $\beta = 10^{-6}$. As it appears, N_j is smaller than N for most values of j, while it is moderately bigger than N for j values that are close to d. In large-scale problems where it is likely to observe a number of support constraints that is significantly smaller than d, the algorithm comes to termination at early stages when $N_j \ll N$ and a large saving of scenarios is obtained.

Further, it is interesting to compare the N_j of Theorem 2 with the insurmountable lower limits given by $\bar{M}_j$, which are also profiled in Figures 1-4 for easy comparison (red crosses). One can see that the N_j and $\bar{M}_j$ curves are close to each other, showing the tightness of the result. The comparison between N_j and $\bar{M}_j$ also suggests an additional interesting observation: knowing in advance that the number of support constraints is always equal to j provides a relatively moderate advantage as compared to running the algorithm and waiting that it stops when the termination condition is reached. This result embodies the essence of the so called "wait-and-judge" paradigm of [13]: exploiting the information contained in the scenarios $\delta_1, \delta_2, \ldots, \delta_N$ by a-posteriori assessing the number of support constraints in a given program levels the advantage that comes from an a-priori knowledge on the complexity.

IV. Proof of Theorem 1

Given the structure of Algorithm 1, it is clear that $V(x^*) > \epsilon$ corresponds to the occurrence for some j of the fact that $s^*_{N_\ell} > \ell$ for $\ell < j$, $s^*_{N_j} \leq j$ (so that $x^* = x^*_{N_j}$), and $V(x^*_{N_j}) > \epsilon$. From this, we have

$$\begin{aligned}
&\mathbb{P}^{N_d}\{V(x^*) > \epsilon\} \\
&\quad = \sum_{j=0}^{d} \mathbb{P}^{N_j}\{s^*_{N_0} > 0 \wedge \ldots \wedge s^*_{N_{j-1}} > j-1 \\
&\qquad\qquad \wedge\ s^*_{N_j} \leq j \wedge V(x^*_{N_j}) > \epsilon\} \\
&\quad \leq \sum_{j=0}^{d} \mathbb{P}^{N_j}\{s^*_{N_j} \leq j \wedge V(x^*_{N_j}) > \epsilon\} \\
&\quad = \sum_{j=0}^{d} \sum_{k=0}^{j} \mathbb{P}^{N_j}\{s^*_{N_j} = k \wedge V(x^*_{N_j}) > \epsilon\} \qquad (6)
\end{aligned}$$

Define $\bar{\epsilon}(k) = \epsilon$ for $k \leq j$ and $\bar{\epsilon}(k) = 1$ for $k > j$. Then, the inner sum in the last expression of (6) can also be written as

$$\begin{aligned}
&\sum_{k=0}^{j} \mathbb{P}^{N_j}\{s^*_{N_j} = k \wedge V(x^*_{N_j}) > \epsilon\} \\
&\quad = \sum_{k=0}^{d} \mathbb{P}^{N_j}\{s^*_{N_j} = k \wedge V(x^*_{N_j}) > \bar{\epsilon}(k)\} \\
&\quad = \mathbb{P}^{N_j}\{V(x^*_{N_j}) > \bar{\epsilon}(s^*_{N_j})\}. \qquad (7)
\end{aligned}$$

By an application of Theorem 1 in [13], it now holds that

$$\mathbb{P}^{N_j}\{V(x^*_{N_j}) > \bar{\epsilon}(s^*_{N_j})\} \leq \gamma^*_j, \qquad (8)$$

where γ^*_j is obtained as the solution of the following variational problem ($\mathsf{C}^d[0,1]$ denotes the set of continuous functions with continuous derivative up to order d and

$\frac{\mathrm{d}^0}{\mathrm{d}t^0}\xi(t)$ has to be intended as $\xi(t)$):

$$\gamma_j^* = \inf_{\xi(\cdot)\in \mathsf{C}^d[0,1]} \quad \xi(1) \tag{9}$$
$$\text{subject to:} \quad \frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\xi(t) \geq \binom{N_j}{k} t^{N_j-k}\cdot \mathbf{1}_{[0,1-\epsilon)}(t),$$
$$t\in[0,1], \quad k=0,1,\ldots,j,$$
$$\frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\xi(t) \geq 0,$$
$$t\in[0,1], \quad k=j+1,\ldots,d.$$

In what follows we will show that for each $j=0,1,\ldots,d$, $\gamma_j^* \leq \frac{\beta}{d+1}$, so that, using (8) in (7) and, in turn, (7) in (6), one obtains

$$\mathbb{P}^{N_d}\{V(x^*)>\epsilon\} \leq \sum_{j=0}^{d}\gamma_j^* \leq \sum_{j=0}^{d}\frac{\beta}{d+1} = \beta, \tag{10}$$

which is the statement of the theorem.

Fix a j in $0,1,\ldots,d$ and consider problem (9). Let $\bar{\xi}(t) = \frac{\beta}{(d+1)(\bar{M}_j+1)}\sum_{m=0}^{\bar{M}_j} t^m$. We want to show that $\bar{\xi}(t)$ is feasible for (9) so that it must be that

$$\gamma_j^* \leq \bar{\xi}(1) = \frac{\beta}{(d+1)(\bar{M}_j+1)}\sum_{m=0}^{\bar{M}_j} 1 = \frac{\beta}{d+1}, \tag{11}$$

which is the sought conclusion. To this purpose, note that

$$\frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\bar{\xi}(t) = \frac{\beta}{(d+1)(\bar{M}_j+1)}\sum_{m=k}^{\bar{M}_j}\binom{m}{k}t^{m-k} \tag{12}$$

is non-negative in $[0,1]$ for all k so that the constraints in (9) for $k=j+1,\ldots,d$ are satisfied. Consider now the constraint for $k=j$: it can be verified by computing for which t the inequality

$$\frac{\beta}{(d+1)(\bar{M}_j+1)}\sum_{m=j}^{\bar{M}_j}\binom{m}{j}t^{m-j} \geq \binom{N_j}{j}t^{N_j-j} \tag{13}$$

is true (note that the left-hand side of (13) is equal to $\frac{1}{j!}\frac{\mathrm{d}^j}{\mathrm{d}t^j}\bar{\xi}(t)$, see (12)). Inequality (13) can be also written as

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

where, recalling that $N_j \geq \bar{M}_j$ by (5), the left-hand side is clearly a strictly decreasing function that tends to $+\infty$ for $t\to 0$ and that tends to 0 for $t\to\infty$. This yields that the inequality (13) is always satisfied over an interval of the type $[0,\tau]$. Since

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

by (5), the interval $[0,\tau]$ includes $[0,1-\epsilon)$, which means that the constraint in (9) for $k=j$ is satisfied.

Having proved the feasibility for $k=j$, the feasibility for all the remaining constraints corresponding to $k<j$ easily follows by induction, because if the constraint for a given $k\leq j$ is satisfied, then we have that

$$\begin{aligned}
&\frac{1}{(k-1)!}\frac{\mathrm{d}^{k-1}}{\mathrm{d}t^{k-1}}\bar{\xi}(t)\\
&= k\cdot\left[\frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\bar{\xi}(0) + \int_0^t \frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\bar{\xi}(\tau)\mathrm{d}\tau\right]\\
&\geq [\tfrac{1}{k!}\tfrac{\mathrm{d}^k}{\mathrm{d}t^k}\bar{\xi}(0)\geq 0]\\
&\geq k\cdot\int_0^t \frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\bar{\xi}(\tau)\mathrm{d}\tau\\
&\geq [\text{the constraint in (9) for } k \text{ holds true}]\\
&\geq k\cdot\int_0^t \binom{N_j}{k}\tau^{N_j-k}\cdot\mathbf{1}_{[0,1-\epsilon)}(\tau)\mathrm{d}\tau\\
&\geq \binom{N_j}{k-1}t^{N_j-k+1}\cdot\mathbf{1}_{[0,1-\epsilon)}(t),
\end{aligned}$$

that is, the constraint for $k-1$ is satisfied as well.

Thus, in conclusion, it remains proven that $\bar{\xi}(t)$ is feasible for (9), which by (11) and (10) concludes the proof. ■

V. Numerical example

Suppose that a number of points p_i are independently sampled from $\mathbb{R}^{50}$ according to the following mechanism: $p_i = q_i + c_i$ where the q_i is a vector drawn from a 50-dimensional Gaussian distribution with zero mean and identity covariance matrix, while c_i is a 50-dimensional vector whose components are all identical with value equal to 0 with probability 95% and a value taken from a Gaussian distribution $G(0,4)$ with probability 5%. Consider then the problem of translating the negative orthant in $\mathbb{R}^{50}$ (i.e., the domain where all components are negative or zero) so that the translated orthant contains all the given points while the translation shift is minimized. This amounts to solving the scenario optimization problem

$$\min_{x\in\mathbb{R}^{50}} \quad \sum_{j=1}^{50} x_j$$
$$\text{subject to:} \quad x_j \geq p_{i,j}, \quad i=1,\ldots,N, \tag{14}$$

where $\delta = p$ and j denotes component. Repeated experiments reveal that, for various value of N, the scenario optimization problem (14) has a complexity that is subject to large variability, spanning with high probability almost the whole range of admissible values.

In this context, the risk corresponds to the probability that a next extracted point p is not within the computed orthant, and, in order to guarantee that the risk is below $\epsilon = 5\%$ with high confidence $\beta = 10^{-6}$, an application of (3) with $d=50$ gives $N=1801$.

In order to reduce the number of scenarios, Algorithm 1 was applied in repeated simulation trials: we ran the algorithm 1000 times, and each time we recorded the total number of scenarios used at the termination of the algorithm. The resulting histogram for the numbers of scenarios used is depicted in Figure 5, where the number $N=1801$ prescribed by the standard one-shot application of scenario optimization

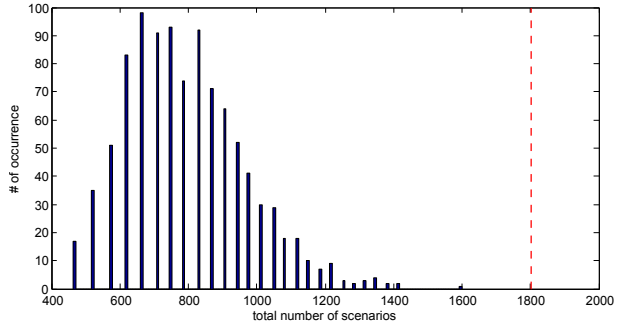

Fig. 5. Histogram of the number of used scenarios in 1000 runs of Algorithm 1

is also shown through a dashed red line. On average, the number of scenarios used by Algorithm 1 is 802.

VI. Conclusions

We have introduced a new scheme for scenario optimization that is able to sequentially learn the sample size required to obtain prescribed levels of risk. The proposed scheme does not rely on validation; rather, all data are used to optimize, while the risk is evaluated by leveraging recent results that link the risk to the scenario solution complexity. A solid theoretical analysis has been presented to obtain a calibration of the algorithm that has been shown to be nearly optimal. The proposed approach allows for a (often large) saving of scenarios as compared to non sequential schemes.

References

[1] 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*, 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. Bayraksan and D.P. Morton. A sequential sampling procedure for stochastic programming. *Operations Research*, 59(4):898–913, 2011.

[4] G. Bayraksan and P. Pierre-Louis. Fixed-width sequential stopping rules for a class of stochastic programs. *SIAM Journal on Optimization*, 22(4):1518–1548, 2012.

[5] G. Calafiore. Repetitive scenario design. *IEEE Transactions on Automatic Control*, 62(3):1125–1137, 2017.

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

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

[8] M.C. Campi. Classification with guaranteed probability of error. *Machine Learning*, 80:63–84, 2010.

[9] M.C. Campi, 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 l_1-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.

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

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

[14] 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.

[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] A. Caré, F.A. Ramponi, and M.C. Campi. A new classification algorithm with guaranteed sensitivity and specificity for medical applications. *IEEE Control Systems Letters*, 2(3):393–398, 2018.

[18] 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*, 61(9):2565–2571, 2016.

[19] L.G. Crespo, S.P. Kenny, and D.P. Giesy. Random predictor models for rigorous uncertainty quantification. *International Journal for Uncertainty Quantification*, 5(5):469–489, 2015.

[20] 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.

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

[22] 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.

[23] 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.

[24] 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*, 60(10):2716–2721, 2015.

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

[26] 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*, 50(12):3009–3018, 2014.

[27] 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.

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

[29] 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.

[30] 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.

[31] 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, 2015.