2025 IEEE 64th Conference on Decision and Control (CDC)
December 10-12, 2025. Rio de Janeiro, Brazil

Online Complexity Estimation for Repetitive Scenario Design

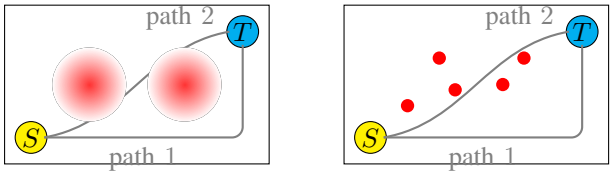
Guillaume O. Berger[1], and Raphaël M. Jungers[1]

***Abstract*— We consider the problem of repetitive scenario design where one has to solve repeatedly a scenario design problem and can adjust the sample size (number of scenarios) to obtain a desired level of risk (constraint violation probability). We propose an approach to learn on the fly the optimal sample size based on observed data consisting in previous scenario solutions and their risk level. Our approach consists in learning a function that represents the pdf (probability density function) of the risk as a function of the sample size. Once this function is known, retrieving the optimal sample size is straightforward. We prove the soundness and convergence of our approach to obtain the optimal sample size for the class of fixed-complexity scenario problems, which generalizes fully-supported convex scenario programs that have been studied extensively in the scenario optimization literature. We also demonstrate the practical efficiency of our approach on a series of challenging repetitive scenario design problems, including non-fixed-complexity problems, nonconvex constraints and time-varying distributions.**

I. INTRODUCTION

This paper is concerned with the problem of using the optimal amount of data in repetitive scenario design. Scenario design is a powerful tool for designing "optimal" solutions while guaranteeing that some random constraint $g(x) \leq 0$, where $g : \mathsf{X} \to \mathbb{R}$ is picked randomly, is satisfied with high probability; see, e.g., [1]–[15]. The principle of scenario design is to replace the random constraint by N i.i.d. samples (called *scenarios*) of it, namely $g_i(x) \leq 0$ for $i = 1, \dots, N$. The final solution is then selected—e.g., by optimizing some preference criterion $J(x)$—among all solutions that satisfy the sampled constraints. To fix ideas, let us consider the example of optimal path planning in an uncertain environment consisting of randomly positioned obstacles (see Fig. 1). In this context, the constraint $g(x) \leq 0$ is to avoid the obstacles and its probability distribution is given by the probability distribution of the position of the obstacles (see Fig. 1a). The scenario design approach consists in drawing N sampled positions of the obstacles, and finding a path that avoids the obstacles in all—or a predefined fraction—of the sampled positions (see Fig. 1b).

Under some conditions on the problem, and if N is large enough, probabilistic guarantees on the constraint violation probability (called the *risk*) of the solution returned by the scenario design algorithm can be given by the theory of scenario design [1], [3], [4], [9], [16], [17]. However, a large sample size N can severely affect the cost of finding a feasible solution to the sampled problem. For instance, if the constraints are nonconvex (as in Fig. 1 for instance), the cost of solving the sampled problem can be exponential in N. Moreover, in cases where one can tolerate some probability of failure, sampling too many constraints can lead to overly conservative solutions [4]. Therefore, finding the smallest N that guarantees an upper bound on the risk is of paramount importance for practical applications.



(a) Model-based problem: pdf of the position of the obstacles.

(b) Scenario problem: sampled positions of the obstacles.

Fig. 1: Comparison between the model-based problem (a) and the scenario problem (b). The scenario design algorithm will choose path 2 over path 1, because both paths avoid the obstacles in all sampled positions (b) but path 1 is shorter. We see (a) that path 1 avoids the obstacles with probability 1, while path 2 avoids the obstacles with probability $1 - \epsilon$, with $\epsilon > 0$. Under some conditions on the problem, and if enough samples are used, this ϵ can be arbitrarily small, with high confidence.

We address this question in the context of *repetitive* scenario design. Repetitive scenario design is when a similar or slowly-varying scenario design task is performed repeatedly. In other words, it is an online version of scenario design that accounts for continued updates in the data and the environment, assuming small magnitude of the updates. Online path planning is a good example, where path planning (Fig. 1) is performed repeatedly to account for changes in the position of the obstacles and the initial condition of the system. Because the task is repeated, and under sufficiently slow variations, we can use the information gathered from previous computations to optimize the sample size N_t in future steps, so that ultimately the sample size is optimal, i.e., $N_t \to N_\star(t)$, where $N_\star(t)$ is the optimal sample size at time t. Nevertheless, this must be done in a cautious way because we cannot exceed the risk tolerance too often during the task. This precludes naive strategies like "augmenting N_t if we exceed the risk tolerance at step $t-1$, and decreasing N_t if we meet the risk tolerance at step $t-1$".

Instead, we propose an approach consisting in learning a function $f_\theta(v, N)$—with arguments $v \in [0, 1]$ and $N \in \mathbb{N}$, and parameter θ—that aims to approximate for each $N \in \mathbb{N}$ the probability density function (pdf) of the risk of the solution returned by the scenario design algorithm with N

[1]Both authors are with ICTEAM, UCLouvain, Louvain-la-Neuve, Belgium. GB is an FNRS Postdoctoral Researcher. RJ is an FNRS honorary Research Associate. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement no. 864017 (L2C), from the Horizon Europe programme under grant agreement no. 101177842 (Unimaas), and from the ARC (French Community of Belgium) (SIDDARTA). Emails: {guillaume.berger,raphael.jungers}@uclouvain.be.

i.i.d. samples. To fit θ, we use previously collected data, which consists of triples (N_s, x_s), for $s = 1, \dots, t$, where t is the current step, x_s is the solution computed at step s. The advantage of this approach is that, through θ, we can retrieve information about the risk pdf $f_\theta(\cdot, N)$, even for values of N that have never been used, and in this way infer the optimal sample size $N_\star$ without exceeding the risk tolerance ϵ too often (Section III). We demonstrate the correctness and convergence (to the optimal sample size) of our approach for the class of "fixed-complexity" scenario problems, when $f_\theta(\cdot, N)$ is the pdf of the *Beta distribution* with parameters θ and $N-\theta+1$ (Section IV). The fixed-complexity scenario problems—which generalize fully-supported random convex programs [2], [3]—serve in this paper as a paradigmatic class of slowly-varying scenario problems. We demonstrate the practical applicability and efficiency of our approach on a wide range of repetitive scenario design problems, including non-fixed-complexity scenario problems, nonconvex constraints, time-varying distribution, etc. (Section V).

Comparison with the literature

The seminal works [2], [3] on convex scenario optimization introduced the notion of fully-supported problems. For these problems, the pdf of the risk is known to be proportional to $v^{d-1}(1-v)^{N-d}$ (Beta distribution), where d is the dimension of the decision variable x. For general convex optimization problems, only an upper bound on the cdf of the risk is known [2], [3], [17], [18]. This upper bound can be quite conservative, especially when the number of "support constraints" is small compared to the dimension of the decision variable [13]. One solution to this problem is the "wait-and-judge" scenario approach [8], [19], where the "complexity of the set of sampled constraints" is computed after the samples are drawn in order to derive an upper bound on the cdf of the risk. Yet, computing the complexity of a set of constraints can be challenging (typically, exponential in N) and the resulting bound on the risk is still conservative [19]. Our approach, by contrast, tries to learn directly the pdf of the risk as a function of N, thereby removing most of the conservatism. For the class of fixed-complexity convex programs studied in Section III, the parameter θ represents the complexity of the problem and is assumed to be fixed (as an ideal case of slowly varying). In this case, the approach can be seen as an indirect way of learning this complexity, not requiring computations of the complexity of sample sets.

The recent papers [20], [21] also use repeated calls to a scenario design algorithm in order to solve a design problem with random constraints. However, their objective is different: they aim to solve the problem only once (not continually as us). For that, they increase progressively the number N of samples until a suitable solution is found. These approaches are not well suited for our online scenario design framework because there is no mechanism for decreasing N, so that N will be conservative with probability one in the long run. Furthermore, the approach in [21] requires to compute the complexity of sample sets, which is something that we want to avoid.

One limitation of our approach is that we need to evaluate the risk $\mathsf{P}[g(x_t) > 0]$ (cf. Section III-A) of the solution x_t at each step t. Computing the risk exactly can be costly in some applications. Alternatively, one can use sample-based approximation methods as in [22, Appendix A]; see also [20, Section II.A]. Clearly, using sample-based methods for evaluating the risk requires to use extra samples; this additional sample complexity, as well as a way to optimize it (e.g., by reusing previous samples or tuning the accuracy of the approximation), is not included in our analysis (we leave it for future work). The results of this paper remain nevertheless valuable also in contexts where risk is evaluated via sampling. For example, when the sampling cost is negligible compared to the computational cost of solving large-scale scenario design problems (e.g., nonconvex problems with complexity exponential in N), it becomes worthwhile to invest effort in accurate risk evaluation if doing so reduces the size of the scenario design problems to be solved.

All proofs are available in the extended version [22].

Notation: $\mathbb{N}$ is the set of nonnegative integers. For $n \in \mathbb{N}$, we let $[n] = \{1, \dots, n\}$. $\|\cdot\|_2$ is the Euclidean norm, and $\|\cdot\|_\infty$ is the L^∞-norm. The following functions will be useful:

- The *Beta function*, defined for all $a, b > 0$ by $B(a,b) = \int_0^1 v^{a-1}(1-v)^{b-1}\,\mathrm{d}v$.
- The *Gamma function*, defined for all $z > 0$ by $\Gamma(z) = \int_0^\infty t^{z-1}e^{-z}\,\mathrm{d}t$.
- The *Digamma function*, defined for all $z > 0$ by $\Psi(z) = \Gamma'(z)/\Gamma(z)$ [23, §6].

II. PROBLEM STATEMENT

We consider an optimization problem of the form

$$\min_{x \in \mathsf{X}} J(x, \xi) \quad \text{s.t.} \quad g(x, \xi) \le 0 \quad \forall\, g \in \mathcal{G}, \tag{1}$$

where $J : \mathsf{X} \times \Xi \to \mathbb{R}$, $\mathcal{G}$ is a set of functions from $\mathsf{X} \times \Xi$ to $\mathbb{R}$, and $\xi \in \Xi$ is an external parameter. (1) is called the *robust design problem* because the constraint $g(x, \xi) \le 0$ must be satisfied for *all* $g \in \mathcal{G}$. A relaxation—and sometimes more realistic version—of the robust problem is the *chance-constrained design problem*:

$$\min_{x \in \mathsf{X}} J(x, \xi) \quad \text{s.t.} \quad \mathsf{P}[g(x, \xi) > 0] \le \epsilon, \tag{2}$$

where P is a probability measure on $\mathcal{G}$, and $\epsilon \in [0, 1]$ is an upper bound on the *risk*, i.e., the probability of violating the constraint $g(x, \xi) \le 0$. The difference between (2) and (1) is that a feasible solution x of (2) is allowed to violate the constraint $g(x, \xi) \le 0$ for some values of $g \in \mathcal{G}$, provided the probability measure of these values does not exceed ϵ. (2) can be very challenging to solve, as it typically involves a nonconvex optimization problem [24]. A way to circumvent this is to consider instead the *scenario design problem*:

$$\min_{x \in \mathsf{X}} J(x, \xi) \quad \text{s.t.} \quad g_i(x, \xi) \le 0 \quad \forall\, i \in [N], \tag{3}$$

where for each $i \in [N]$, $g_i \in \mathcal{G}$. Clearly, (3) is a relaxation of (1) since only a subset of $\mathcal{G}$ is used. For this reason, a

feasible solution of (3) is not expected to be feasible for (1) in general. However, if N is large and the g_i's are sampled i.i.d. according to P, one can expect that the solution of (3) is feasible for (2). Note that when the g_i's are sampled at random, the solution of (3) is also a random variable (since it depends on the samples). Therefore, the property that the solution of (3) is feasible for (2) is a random property whose probability can be quantified.

To formalize the above, we introduce some notation. Given $\xi \in \Xi$ and $(g_1, \dots, g_N) \in \mathcal{G}^N$, we denote by $\mathsf{A}_\xi(g_1, \dots, g_N)$ the solution of (3) with $g_1, \dots, g_N$.[1] Given $x \in \mathsf{X}$ and $\xi \in \Xi$, we denote by $V_\mathsf{P}(x, \xi)$ its *risk*, i.e., the probability of violating the constraint $g(x, \xi) \leq 0$ with respect to P: $V_\mathsf{P}(x, \xi) = \mathsf{P}[g(x, \xi) > 0]$. When P and ξ are clear from the context, we write A and $V(x)$ for A_ξ and $V_\mathsf{P}(x, \xi)$. The probability that the solution of (3) is feasible for (2) is then given by

$$C_\xi(\epsilon, N) \coloneqq \mathsf{P}^N\big(\{\mathbf{g} \in \mathcal{G}^N : V_\mathsf{P}(\mathsf{A}_\xi(\mathbf{g}), \xi) \leq \epsilon\}\big), \qquad (4)$$

where $\mathbf{g}$ is a shorthand notation for $(g_1, \dots, g_N)$. Several lower bounds on $C_\xi(\epsilon, N)$ have been proposed in the literature. These bounds generally depend on an intrinsic quantity of the problem of interest, called its *complexity* [2], [3], [16], [17]. The definition of complexity varies from one approach to another. For non-degenerate convex scenario programs, the dimension of the decision variable can be used as a complexity measure [2], [3]. Other common complexity measures include the VC dimension, the Rademacher complexity and the compression size [16], [17]. Yet, sharp bounds on these quantities are generally elusive, resulting in overly conservative sample size requirements.

We address the problem of finding sharp upper bounds on $C_\xi(\epsilon, N)$ in the context of *repetitive scenario design*, that is, when (3) is solved repeatedly with ξ (or more precisely its effect on (3)) varies slowly. The goal is that ultimately N is close to the smallest value such that $C_\xi(\epsilon, N) \geq \beta$, where $\beta \in [0, 1]$ is a given *confidence parameter*. In other words, the problem we address is the following:

Problem 1: Let $\mathcal{G}$, P and A be as above. Given $\epsilon \in [0, 1]$ and $\beta \in [0, 1]$, let $N_\star(\xi) = \min\{N \in \mathbb{N} : C_\xi(\epsilon, N) \geq \beta\}$. Find a repetitive scenario design algorithm such that, with high probability, $N_t \approx N_\star(\xi_t)$ for all $t \in \mathbb{N}_{>0}$ large enough, where N_t is the sample size used at step t of the algorithm and ξ_t is the value of the external parameter at step t.

We stress out that a small change in N can have a large impact on the computational complexity of solving (3); for nonconvex problems with exact computations, the dependence in N can be exponential or worse.

We conclude this section with an example:

Example 1: Consider the problem of finding the shortest path between a source location ξ_S and a target location ξ_T while avoiding moving obstacles, as depicted in Fig. 2a. In the spirit of [25], [26], we parametrize the path by using H "viapoints", denoted by $x_1, \dots, x_H \in \mathbb{R}^2$, which are our decision variables. Each viapoint needs to avoid the obstacles and two consecutive viapoints cannot be more than δ units of distance apart. Problem 1 becomes

$$\begin{aligned} \min_{x_1,\dots,x_H \in \mathsf{B}} \quad & \|x_H - \xi_T\|_2 \\ \text{s.t.} \quad & \|x_t - x_{t-1}\|_2 \leq \delta, \quad t \in [H], \\ & g(x_t, \xi) \leq 0, \quad t \in [H], \quad g \in \mathcal{G}, \end{aligned} \qquad (5)$$

where $x_0 = \xi_S$, $\mathsf{B} = [0, 5] \times [0, 3]$, and for each $g \in \mathcal{G}$, $g(x, \xi)$ is the signed distance (positive in case of collision) between x and the obstacles in one specific position. In the example of Fig. 2a, each $g \in \mathcal{G}$ has the form

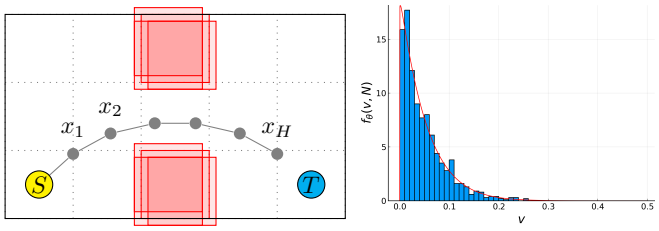
$$g(x, \xi; \tilde{x}_l, \tilde{x}_u) = \min\{\tfrac{1}{2} - \|x - \tilde{x}_l\|_\infty, \tfrac{1}{2} - \|x - \tilde{x}_u\|_\infty\},$$

where $\tilde{x}_l, \tilde{x}_u \in \mathbb{R}^2$ are the centers of the lower and upper obstacles, respectively. Note that in Fig. 1 there is no path from ξ_S to ξ_T that satisfies the constraint $g(x) \leq 0$ for all $\tilde{x}_l, \tilde{x}_u \in \mathbb{R}^2$. Hence, the robust problem (5) is typically infeasible. Assume that a probability distribution P on the value of $\tilde{x}_l$ and $\tilde{x}_u$, e.g., $[\tilde{x}_l; \tilde{x}_u] \sim \mathcal{N}(\mu, \Sigma^2)$, is given. In this case, the chance-constrained problem (2) becomes relevant and captures the fact that the path avoids the obstacles with high probability. Solving the chance-constrained problem, which is often intractable, can be approached by solving the scenario problem (3). In this case, it amounts to sample N values of $(\tilde{x}_l, \tilde{x}_u)$ and solve the optimization problem

$$\begin{aligned} \min_{x_1,\dots,x_H \in \mathsf{B}} \quad & \|x_H - \xi_T\|_2 \\ \text{s.t.} \quad & \|x_t - x_{t-1}\|_2 \leq \delta, \quad t \in [H], \\ & g(x_t, \xi; \tilde{x}_{l,i}, \tilde{x}_{u,i}) \leq 0, \quad t \in [H],\ i \in [N]. \end{aligned} \qquad (6)$$

The determination of an optimal sample size N is the central question of this paper. ◁

In the rest of this paper, we assume that ξ is fixed, and we address Problem 1 under this assumption. The idea is that the results obtained for ξ fixed will be useful in practice for problems with slowly-varying ξ (as we showcase in numerical experiments in Section V). A theoretical analysis of the case of slowly-varying ξ is left for future work.



(a) Problem illustration. (b) Histogram of risk.

Fig. 2: Path planning problem of Example 1. (a) The path is represented by a linked sequence of H viapoints (gray dots). Each viapoint must avoid the obstacles (red regions) and two consecutive viapoints cannot be more than δ units of distance apart. (b) Histogram (using $M = 1000$ experiments) of the risk of paths computed with $N = 20$ sampled positions of the obstacles (hence, a total of 20 000 sampled constraints), and $H = 100$ viapoints. The red curve is $f_\theta(\cdot, N)$ where $\theta = \theta_\star(\mathcal{D})$ and $\mathcal{D}$ is the data set obtained from the M experiments (cf. Section III).

[1] Without loss of generality, we assume that the solution exists and is unique; see, e.g., [3] for ways to handle problems with no solutions or non-unique solutions. Note that this definition of A is very general: it includes any scenario optimization problems, and any method to obtain an exact *or approximate* solution to it. Allowing approximate solutions is particularly relevant when dealing with nonconvex or NP-hard problems.

III. PROPOSED APPROACH

Our approach consists in learning a function $f_\theta(v, N)$ that approximates for each $N \in \mathbb{N}$ the pdf of $V(\mathsf{A}(g_1, \dots, g_N))$ when $g_1, \dots, g_N$ are sampled i.i.d. from P. In other words, we aim to find a value of θ such that $f_\theta(v, N) \approx f(v, N)$, where $f(v, N) \coloneqq \frac{\mathrm{d}}{\mathrm{d}v} \mathsf{P}^N\big(\{\mathsf{g} \in \mathcal{G}^N : V(\mathsf{A}(\mathsf{g})) \leq v\}\big)$. To do that, given a data set $\mathcal{D} = \{(v_j, N_j)\}_{j=1}^M \subseteq [0,1] \times \mathbb{N}$, we define the Maximum Likelihood Estimator (MLE):

$$\theta_\star(\mathcal{D}) = \arg\max_{\theta \in \Theta} \ell(\theta; \mathcal{D}), \tag{7}$$

where $\ell(\theta; \mathcal{D}) \coloneqq \frac{1}{M} \sum_{j=1}^M \log(f_\theta(v_j, N_j))$.

In the rest of this section, we first explain how we build the data set $\mathcal{D}$ in the context of repetitive scenario design. Then, we describe the parametrization that we consider for f_θ and explain how we compute $\theta_\star(\mathcal{D})$ for this parametrization. Finally, we explain how we estimate the optimal sample size N given a parameter θ satisfying $f_\theta(v, N) \approx f(v, N)$.

A. Data collection

In the context of repetitive scenario design, (3) is solved repeatedly with different sample sets. Let N_t be the sample size used in step t, and let x_t be the returned solution. For each $t \in \mathbb{N}_{>0}$, let $v_t = V(x_t)$. This gives at each step $t \in \mathbb{N}_{>0}$, the data set $\mathcal{D}_t = \{(v_s, N_s)\}_{s=1}^t$, from which the MLE $\theta_t = \theta_\star(\mathcal{D}_t)$ can be computed.

Remark 1: Given x, $V(x)$ can sometimes be easily computed; this is the case, e.g., for the problems in Sections V-A and V-C. Otherwise, it can always be easily approximated to high accuracy by using a Bernoulli test (e.g., the problems in Section V-B); see [22, Appendix A] for details.

B. Distribution shape and fitting

In this paper, we propose the following parametrization of f_θ: for $v \in (0, 1]$ and $N \in \mathbb{N}_{>\theta}$,

$$f_\theta(v, N) = \frac{v^{\theta-1}(1-v)^{N-\theta}}{B(\theta, N-\theta+1)}, \quad \theta \in \Theta \coloneqq \mathbb{R}_{>0}, \tag{8}$$

where B is the Beta function (cf. Section I). Note that for all $N \in \mathbb{N}_{>\theta}$, $f_\theta(\cdot, N)$ is the pdf of the distribution $\mathrm{Beta}(\theta, N-\theta+1)$ [27, §25]. The reasons for this choice of f_θ will be apparent in Theorem 1. Note that (8) requires that $v > 0$ and $N > \theta$. However, we want to allow that $\theta \geq N_j$ or $v_j = 0$ for some $(v_j, N_j) \in \mathcal{D}$. Therefore, we extend the domain of f_θ as follows: for $v \in [0, 1]$ and $N \in \mathbb{N}$,

$$f_\theta(v, N) = \begin{cases} 1 & \text{if } v = 0 \text{ or } N = 0, \\ \frac{v^{\theta-1}(1-v)^{N-\theta}}{B(\theta, N-\theta+1)} & \text{if } v \neq 0 \text{ and } N > \theta, \\ N v^{N-1} & \text{otherwise.} \end{cases} \tag{9}$$

We discuss below some properties of f_θ that are useful for optimizing the log-likelihood $\ell(\cdot; \mathcal{D})$:

Proposition 1: Let $v \in [0, 1]$ and $N \in \mathbb{N}$. The function $\theta \mapsto f_\theta(v, N)$ is upper semi-continuous on Θ.

Proposition 2: Let $v \in (0, 1)$ and $N \in \mathbb{N}_{>0}$. The function $\theta \mapsto \log(f_\theta(v, N))$ is smooth and strictly concave on $(0, N)$.

Corollary 1: Let $\mathcal{D} = \{(v_j, N_j)\}_{j=1}^M \subseteq [0, 1] \times \mathbb{N}$. It holds that $\ell(\cdot; \mathcal{D})$ is upper semi-continuous on Θ, and for every interval $I \subseteq \Theta \setminus \{N_j\}_{j=1}^M$, $\ell(\cdot; \mathcal{D})$ is either constant on I, or is smooth and strictly concave on I.

Corollary 1 implies that the maximum of $\ell(\theta; \mathcal{D})$ exists and is finite. It also implies that $\ell(\theta; \mathcal{D})$ can be maximized easily by considering separately each maximal interval I of $\Theta \setminus \{N_j\}_{j=1}^M$; indeed, on each I, $\ell(\theta; \mathcal{D})$ is either constant or smooth and concave. The maximum of a concave function defined on $\mathbb{R}$ can be computed very efficiently and reliably, e.g., using iterative methods. In our numerical experiments, we used Newton–Raphson's method with an educated guess for the initial iterate; this method proved very efficient in all experiments, eliminating the need for additional measures or adjustments.

Example 2: The MLE $\theta_\star(\mathcal{D})$ for the problem in Example 1 is represented in Fig. 2b. We collected $M = 1000$ data points by solving the scenario problem 1000 times with $N = 20$ and $H = 100$. ◁

C. Update of the sample size

Given a parameter value θ such that $f_\theta(v, N) \approx f(v, N)$, a risk tolerance $\epsilon \in [0, 1]$ and a confidence $\beta \in [0, 1]$, we estimate the optimal sample size $N_\star$ in Problem 1 as follows. By definition of f, it holds that $C(\epsilon, N) = \int_0^\epsilon f(v, N)\,\mathrm{d}v$. This gives the optimal sample size estimate

$$\bar{n}(\theta; \epsilon, \beta) = \min\left\{N \in \mathbb{N} : \int_0^\epsilon f_\theta(v, N)\,\mathrm{d}v \geq \beta\right\}.$$

Since the value of the integral is increasing with N, $\bar{n}(\theta; \epsilon, \beta)$ can be computed efficiently by using bisection.

The overall algorithm is presented in Algo. 1. Note that if $V(x)$ is estimated by using a Bernoulli test as explained in [22, Appendix A], then P needs not to be known precisely by the algorithm; a *generative model* (i.e., an oracle generating i.i.d. samples from P) is sufficient.

Remark 2: The parameter $N_{\max}$ in Algo. 1 can be arbitrarily large. Its purpose is only to simplify the analysis of the convergence of the algorithm (see Theorem 1). In future work, we will work on removing this parameter, even though in practice, $\{N_t\}_{t=1}^\infty$ will always be bounded due to hardware and software limitations.

Algorithm 1: Repetitive Scenario Design

Data: $\epsilon \in [0, 1]$, $\beta \in [0, 1]$, $N_1 \in \mathbb{N}$, $N_{\max} \in \mathbb{N}$.
$\mathcal{D}_0 \leftarrow \emptyset$ `// Data set`
for $t = 1, 2, \dots$ **do**
  Draw N_t i.i.d. samples $g_1, \dots, g_{N_t} \sim \mathsf{P}$
  Let $x_t \leftarrow \mathsf{A}(g_1, \dots, g_{N_t})$
  Let $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{(V_\mathsf{P}(x_t), N_t)\}$
  Let $\theta_t \leftarrow \theta_\star(\mathcal{D}_t)$ `// see (7)`
  Let $N_{t+1} \leftarrow \min\{\bar{n}(\theta_t; \epsilon, \beta), N_{\max}\}$

IV. ANALYSIS OF THE ALGORITHM

We demonstrate the convergence of Algo. 1 when applied to *fixed-complexity* scenario problems, defined below. This

notion generalizes the notion of fully-supported convex scenario programs, which have been extensively studied in the scenario optimization literature [2], [3], [12]–[14].[2]

Definition 1: Given $\mathcal{G}$, P, A and $d \in \mathbb{N}_{>0}$, we say that $(\mathcal{G}, \mathsf{P}, \mathsf{A})$ has *fixed complexity* d if for all $N \in \mathbb{N}_{\geq d}$, the following holds with probability one on $(g_1, \ldots, g_N) \in \mathcal{G}^N$: there is a unique subset $(g_{i_1}, \ldots, g_{i_d})$ with $1 \leq i_1 < \ldots < i_d \leq N$ such that $\mathsf{A}(g_1, \ldots, g_N) = \mathsf{A}(g_{i_1}, \ldots, g_{i_d})$.

Remark 3: When applied to problems with non-fixed ξ, the analog condition would be that for all $\xi \in \Xi$, $(\mathcal{G}, \mathsf{P}, \mathsf{A}_\xi)$ has fixed complexity d_ξ, where d_ξ changes slowly over time. A formal analysis is left for future work.

When applied to fixed-complexity problems, Algo. 1 converges to the optimal sample size, thereby providing a valid solution to Problem 1. This is formalized in the next theorem, which is the main theoretical result of this paper:

Theorem 1: Let $(\mathcal{G}, \mathsf{P}, \mathsf{A})$ have fixed complexity $d \in \mathbb{N}_{>0}$. Let $0 < \epsilon < \beta < 1$ and $N_\star = \min\{N \in \mathbb{N} : C(\epsilon, N) \geq \beta\}$. Consider the sequences $\{N_t\}_{t=1}^\infty$ and $\{\theta_t\}_{t=1}^\infty$ generated by Algo. 1. Assume that $N_\star \leq N_{\max}$. The following holds with probability one: (i) $\theta_t \to d$, and (ii) there is $t_0 \in \mathbb{N}_{>0}$ such that for all $t \in \mathbb{N}_{\geq t_0}$, $N_t \in [N_\star - 1, N_\star + 1]$.

The theoretical analysis is for the moment restricted to the class of fixed-complexity scenario problems. However, in the next section, we show that our approach performs also very well in practice on a series of challenging repetitive scenario design problems, including non-fixed-complexity problems, nonconvex constraints and time-varying distributions.

V. NUMERICAL EXPERIMENTS

In all experiments, we used $\epsilon = 0.1$ and $\beta = 0.9$.[3]

A. Fixed-complexity problems

We start with two fixed-complexity scenario problems. The first problem consists in solving

$$\min_{x \in \mathbb{R}} x \quad \text{s.t.} \quad x \geq u, \quad u \sim \mathcal{N}(1, 2). \tag{10}$$

This problem has fixed complexity $d = 1$. The evolution of N_t and θ_t over the first 1000 steps is represented in Fig. 3. We also represented the cumulative of $V(x_t)$. We notice that $\theta_t \to 1$, and $V(x_t) \leq \epsilon$ with frequency at least β.

Remark 4: Note that for all our experiments, we reported the cumulative distribution of $\{V(x_t)\}_{t=1}^T$, where T is the total number of steps. This is because we are interested in the frequency with which $V(x_t)$ exceeds the risk tolerance over the *whole* repetitive scenario design process.

The second problem is

$$\min_{x \in \mathbb{R}^{20}} \sum_{i=1}^{20} x^{(i)} \quad \text{s.t.} \quad u^\top x \leq 1, \quad u \sim \mathcal{N}(0, I). \tag{11}$$

This problem has fixed complexity $d = 20$. The evolution of N_t and θ_t over the first 1000 steps is represented in Fig. 3. We also represented the cumulative of $V(x_t)$. We notice that $\theta_t \to 20$, and $V(x_t) \leq \epsilon$ with frequency almost β.

[2]It is worth noting that fixed-complexity scenario problems satisfy the non-degeneracy assumption [2], [3], [13].

[3]The codes are available at `https://github.com/guberger/OnlineScenarioOptimization.jl`.

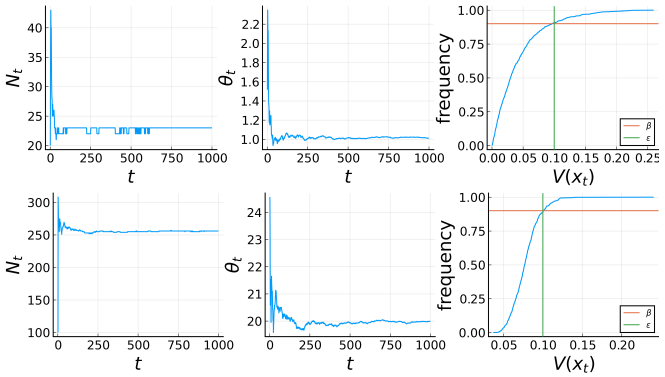


Fig. 3: Fixed-complexity problems; top: (10), bottom: (11). Evolution of N_t and θ_t over $T = 1000$ steps and cumulative of $\{V(x_t)\}_{t=1}^T$. We observe that the algorithm converges very fast to d and $N_\star$. We also observe that the violation probability is smaller than 0.1, (almost) 90% of the time.

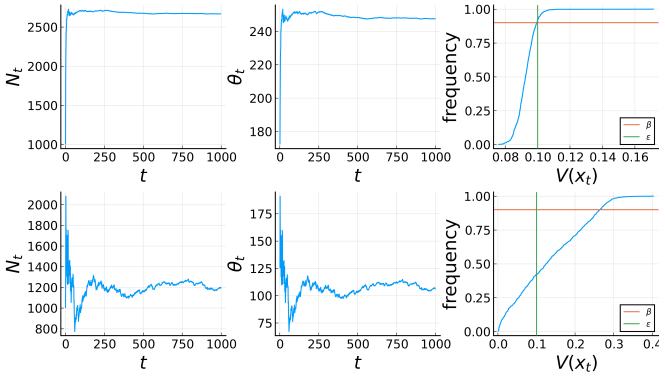


Fig. 4: Non-fixed complexity problems from [13]; top: (12) with P as in [13, Fig. 8], bottom: (12) with P as in [13, Fig. 9]. Evolution of N_t and θ_t over $T = 1000$ steps and cumulative of $\{V(x_t)\}_{t=1}^T$. On the top, we observe that the algorithm converges very fast to some $\theta_\circ$ and $N_\circ$, even though the complexity is not fixed. We also observe that the violation probability is smaller than 0.1, 90% of the time. On the bottom, however, the algorithm does not seem to converge to some θ or N, and the violation probability is smaller than 0.1, less than 90% of the time. An explanation for that is that the problem has "degenerate complexity".

B. Beyond fixed-complexity problems

Next, we consider more general scenario problems. The first problem, borrowed from [13], consists in solving

$$\min_{x \in \mathbb{R}^{400}} \sum_{i=1}^{400} x^{(i)} \quad \text{s.t.} \quad \min_{i=1}^{400} x^{(i)} \geq u, \quad u \sim \mathsf{P}. \tag{12}$$

We let P be the probability distribution used in [13, Fig. 8]. The results are given in Fig. 4. We notice that $V(x_t) \leq \epsilon$ with frequency at least β.

We also tried with the probability distribution used in [13, Fig. 9]. The results are given in Fig. 4. In this case, however, the frequency of $V(x_t) \leq \epsilon$ is *smaller than* β. This failure to meet the safety requirements can be explained by the fact that the problem is far from having fixed complexity (see [13, Fig. 9]). In future work, we plan to investigate algorithmic ways to detect such problems that have a non-fixed complexity with high variance, and provide sound ways to converge to their optimal sample size.

Remark 5: For these two problems, the risk was estimated using the technique in [22, Appendix A], with $S = 10^4$,

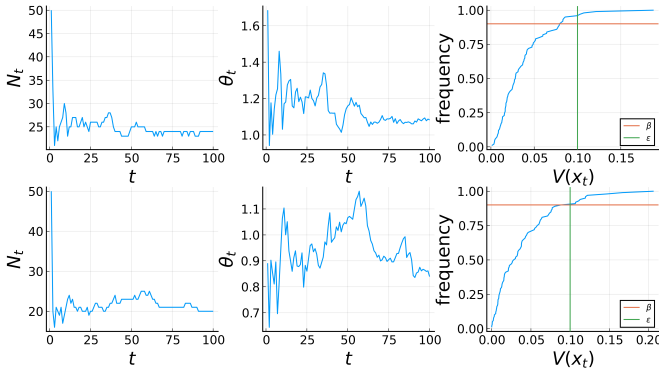


Fig. 5: Path planning problem from Example 1; top: steady distribution (see Section V-B), bottom: time-varying distribution (see Section V-C). Evolution of N_t and θ_t over $T = 100$ steps and cumulative of $\{V(x_t)\}_{t=1}^{T}$. On the top, we observe that the algorithm converges fast to some $\theta_\circ$ and $N_\circ$, even though the complexity is not fixed. On the bottom, we observe that the value of θ_t and N_t is time-varying, as is the distribution. For both, we observe that the violation probability is smaller than 0.1, 90% of the time.

giving an accuracy of $\eta = 0.025$ with probability $1 - 10^{-5}$.

We also applied our technique on the path planning problem in Example 1 with $H = 100$, $\delta = 0.045$, $\tilde{x}_l = [\frac{5}{2}, y-0.8]$ and $\tilde{x}_u = [\frac{5}{2}, y + 0.8]$, where $y \sim \mathcal{N}(1.5, 0.05)$. Note that the problem is nonconvex and the dimension of the decision variable is 200. The results are presented in Fig. 5. We notice that $V(x_t) \leq \epsilon$ with frequency at least β.

C. Time-varying distribution

Finally, we modify the path planning problem in Example 1 so that the distribution on the constraints is time-varying. Namely, we let $\tilde{x}_l(t) = [\frac{5}{2} + \sin(0.1t), y - 0.3]$ and $\tilde{x}_u(t) = [\frac{5}{2} + \sin(0.1t), y + 0.3]$, where $y \sim \mathcal{N}(1.5, 0.05)$ and $t \in \mathbb{N}_{>0}$ is the time step. Because the distribution is shifting, we put more weight on the most recent data points with a rule proportional to the time step: $w_t = t$. The results are presented in Fig. 5. We notice that although the distribution is shifting, $V(x_t) \leq \epsilon$ with frequency at least β.

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 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.

[3] G. C. Calafiore, "Random convex programs," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3427–3464, 2010.

[4] 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.

[5] K. Margellos, P. 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.

[6] P. M. Esfahani, T. Sutter, and J. Lygeros, "Performance bounds for the scenario approach and an extension to a class of non-convex programs," *IEEE Transactions on Automatic Control*, vol. 60, no. 1, pp. 46–58, 2015.

[7] S. Grammatico, X. Zhang, K. Margellos, P. 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.

[8] M. C. Campi and S. Garatti, "Wait-and-judge scenario optimization," *Mathematical Programming*, vol. 67, no. 1, pp. 155–189, 2018.

[9] M. C. Campi, S. Garatti, and F. A. Ramponi, "A general scenario theory for nonconvex optimization and decision making," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4067–4078, 2018.

[10] Y. Yang and C. Sutanto, "Chance-constrained optimization for non-convex programs using scenario-based methods," *ISA transactions*, vol. 90, pp. 157–168, 2019.

[11] R. Rocchetta and L. G. Crespo, "A scenario optimization approach to reliability-based and risk-based design: soft-constrained modulation of failure probability bounds," *Reliability Engineering & System Safety*, vol. 216, no. 107900, 2021.

[12] L. Romao, K. Margellos, and A. Papachristodoulou, "Tight sampling and discarding bounds for scenario programs with an arbitrary number of removed samples," in *Proceedings of the 3rd Conference on Learning for Dynamics and Control*. PMLR, 2021, pp. 312–323.

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

[14] L. Romao, A. Papachristodoulou, and K. Margellos, "On the exact feasibility of convex scenario programs with discarded constraints," *IEEE Transactions on Automatic Control*, vol. 68, no. 4, pp. 1986–2001, 2023.

[15] M. C. Campi and S. Garatti, "Compression, generalization and learning," *Journal of Machine Learning Research*, vol. 24, no. 339, pp. 1–74, 2023.

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

[17] 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.

[18] N. Littlestone and M. Warmuth, "Relating data compression and learnability," 1986, unpublished manuscript.

[19] S. Garatti and M. C. Campi, "The risk of making decisions from data through the lens of the scenario approach," *IFAC-PapersOnLine*, vol. 54, no. 7, pp. 607–612, 2021.

[20] G. C. Calafiore, "Repetitive scenario design," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1125–1137, 2016.

[21] S. Garatti, A. Carè, and M. C. Campi, "Complexity is an effective observable to tune early stopping in scenario optimization," *IEEE Transactions on Automatic Control*, vol. 68, no. 2, pp. 928–942, 2023.

[22] G. O. Berger and R. M. Jungers, "Online complexity estimation for repetitive scenario design," 2025, arXiv preprint arXiv:2509.02103.

[23] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, 10th ed. New York, NY: Dover Publications, 1972.

[24] A. Charnes and W. W. Cooper, "Chance-constrained programming," *Management Science*, vol. 6, no. 1, pp. 73–79, 1959.

[25] J. Jankowski, L. Brudermüller, N. Hawes, and S. Calinon, "VP-STO: Via-Point-based Stochastic Trajectory Optimization for reactive robot behavior," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 125–10 131.

[26] L. Brudermüller, G. Berger, J. Jankowski, R. Bhattacharyya, R. Jungers, and N. Hawes, "CC-VPSTO: Chance-Constrained Via-Point-based Stochastic Trajectory Optimisation for safe and efficient online robot motion planning," 2024, arXiv preprint arXiv:2402.01370.

[27] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous univariate distributions, volume 2*, 2nd ed. New York, NY: John Wiley & Sons, 1995.