Noname manuscript No.
(will be inserted by the editor)

Risk and Complexity in Scenario Optimization

S. Garatti · M.C. Campi

Received: date / Accepted: date

Abstract Scenario optimization is a broad methodology to perform optimization based on empirical knowledge. One collects previous cases, called "scenarios", for the set-up in which optimization is being performed, and makes a decision that is optimal for the cases that have been collected. For convex optimization, a solid generalization theory has been developed that provides guarantees of performance, and constraint satisfaction, of the scenario solution. In this paper, we open a new direction of investigation: the risk that a performance is not achieved, or that constraints are violated, is studied jointly with the complexity (as precisely defined in the paper) of the solution. It is shown that the joint probability distribution of risk and complexity is concentrated in such a way that the complexity carries fundamental information to tightly judge the risk. This result is obtained without availing of extra knowledge on the underlying optimization problem than that carried by the scenarios, in particular, no extra knowledge on the distribution by which scenarios are generated is assumed, so that the result is broadly applicable. This deep-seated result unveils a fundamental and general structure of data-based optimization and suggests practical approaches for risk assessment.

1 Introduction: scenario programs and the generalization issue

The scenario approach is a framework to perform optimization in uncertain environments where one has access to a record of past cases for the set-up where the present decision has to be made. The prototype convex scenario program is written as

$$\begin{aligned} &\min_{x\in\mathcal{X}} \ c^T x \\ &\text{subject to: } \ x \in \bigcap_{i=1,\ldots,N} \mathcal{X}_{\delta_i}, \end{aligned} \tag{1}$$

S. Garatti
Dipartimento di Elettronica, Informazione e Bioingegneria - Politecnico di Milano, piazza Leonardo da Vinci 32, 20133 Milano, Italia.
E-mail: simone.garatti@polimi.it
Web-site: http://home.deib.polimi.it/sgaratti/

M.C. Campi
Dipartimento di Ingegneria dell'Informazione - Università di Brescia, via Branze 38, 25123 Brescia, Italia.
Tel.: +39-030-3715458
Fax: +39-030-380014
E-mail: marco.campi@unibs.it
Web-site: http://bsing.ing.unibs.it/∼campi/

where $x \in \mathbb{R}^d$ is a vector of optimization variables and c is a constant vector of weights; $\mathcal{X} \subseteq \mathbb{R}^d$ is a convex set and $\mathcal{X}_{\delta_i}$ are instances of a family $\{\mathcal{X}_\delta\}$ of convex constraints parameterized by δ. Parameter δ is modeled as a random variable defined over a probability space $(\Delta, \mathcal{F}, \mathbb{P})$, and δ_i, $i = 1, \ldots, N$, is an independent random sample of δ values.[1] The interpretation of (1) is that the δ_i's are observations (or "scenarios") of an uncertain phenomenon (e.g. demand in the energy market or temperature in a given environment), and one makes a decision which is optimal according to the cost function $c^T x$ (e.g. minimize energy production or minimize the ranges for the forecast of monthly temperatures) while also satisfying the constraints that come from previous cases δ_i's (e.g. energy balance for a record of demands or including in the ranges the temperatures that have been recorded in the past). See [5,45,21] for broader presentations of data-based optimization.

Program (1) is convex, and this sets its fundamental structure: the optimization domain, $\mathcal{X}$, as well as the constraints, $x \in \mathcal{X}_{\delta_i}$, are convex and the cost function is linear. Note that linearity of the cost function is not a limiting assumption within a convex set-up because any program with a convex, but nonlinear, cost function can be re-written as one with a linear cost function by an epigraphic reformulation, [6]. Convexity makes solving program (1) computationally tractable even in high dimensions and in the presence of many optimization variables.

Albeit clearly not all problems are convex, the set-up of (1) is truly vast and encompasses problems that come from a variety of fields that range from finance, [38, 39,40,27], to control, [15,25,50,41,26], from prediction, [10,18,19,20], to machine learning, [9,32]. A first common situation is the minimization of a loss function $\ell(v, \delta)$ that depends on one's choice v and on an uncertain variable δ.[2] Given a sample of scenarios, solving the worst-case program

$$\min_{v} \max_{i=1,\ldots,N} \ell(v, \delta_i)$$

is equivalent to (1) after one introduces a new variable $z \in \mathbb{R}$, called the "performance variable", and set $x = (v, z)$, $c^T x = [0 \ \cdots \ 0 \ 1]x = z$, and $\mathcal{X}_{\delta_i} = \{(v, z) : z \geq \ell(v, \delta_i)\}$. In this context, enforcing the constraints $x \in \mathcal{X}_{\delta_i}$, $i = 1, \ldots, N$, leads to worst-case optimization over the scenarios; in other words, one makes a choice such that no other selection of the optimization variable v would lead to a better value simultaneously over all the scenario-based loss functions $\ell(v, \delta_i)$. In finance, this set-up is known as empirical Value at Risk (VaR), [24]. More generally, constraints $x \in \mathcal{X}_{\delta_i}$ in (1) reflect needs of various type that go from saturation limits in control applications to obstacle avoidance in mobile robotics, from resource availability in management problems to bandwidth capacity in telecommunications.

1.1 Generalization theory

In recent years, a broad stream of literature has dealt with the generalization properties of sample-based solutions in stochastic programming, [31,33,2,28,29,3,38,43, 34]. In this section, we specifically refer to the properties of the solution obtained by solving (1). In (1), one fundamental feature is that the scenarios are observations

[1] No limitations are imposed on Δ like e.g. that Δ is a subset of a Eucledian space or of a vector space, nor is Δ endowed with a metric or a topology. Δ is just a generic set that forms a probability space together with $\mathcal{F}$ and $\mathbb{P}$. Hence, ideas like "the sample δ_i, $i = 1, \ldots, N$, covers, or fills up, Δ" are void of any meaning. This generality in the definition of Δ is important for the widespread applicability of the theory.

[2] Assume that function $\ell(v, \delta)$ is convex in v for any given value of δ, while its dependence on δ is arbitrary.

so that the generalization features of the solution cannot rely on testing it against new scenarios drawn from an available distribution of uncertainty.

To describe the existing generalization results, start by introducing the notation x_N^* for the solution to program (1),[3] and the following definition of risk.

Definition 1 (risk) The *risk* of a given $x \in \mathcal{X}$ is defined as

$$V(x) = \mathbb{P}\{\delta \in \Delta : x \notin \mathcal{X}_\delta\}.$$

Hence, $V(x)$ is the probability with which constraints are not satisfied by x. Note that $V(\cdot)$ is a deterministic function from $\mathcal{X}$ to $[0,1]$. The risk of x_N^* is the random variable $V(x_N^*)$ obtained by computing $V(\cdot)$ corresponding to the solution x_N^* of (1). Note that $V(x_N^*)$ is stochastic through the dependence of x_N^* on $\delta_1, \ldots, \delta_N$. □

When the constraints stem from an uncertain loss function $\ell(v, \delta)$ as described above, $V(x) = V(v, z)$ quantifies the probability that in a new case the loss associated with v exceeds z, so that the risk of (v_N^*, z_N^*) is the probability that applying the choice v_N^* results in a loss greater than z_N^*. More generally, $V(x_N^*)$ is a measure of the probability that some undesired event or condition occurs when the solution x_N^* is applied. If $V(x_N^*) \leq \epsilon$, then the risk for the solution to violate the random constraints $x \in \mathcal{X}_\delta$ is no more than ϵ. According to the stochastic programming terminology, this is expressed that x_N^* is a chance-constrained feasible point at level ϵ, [22,43].

Together with the optimal cost value $c^T x_N^*$, $V(x_N^*)$ represents the fundamental quantity to evaluate the level of satisfaction one has in the solution x_N^*. Interestingly, the optimal cost value $c^T x_N^*$ becomes available to the user after the solution x_N^* has been computed. In contrast, the value of $V(x_N^*)$ depends on the distribution of δ, which in real applications is normally unknown or only imprecisely known: hence, $V(x_N^*)$ cannot be computed even after the optimization process has been completed. The problem of estimating $V(x_N^*)$ has attracted much attention over the past 10 years, and deep results have been established which (in a sense that we shall discuss in detail later) affirm that the distribution of $V(x_N^*)$ can be bounded even when no knowledge on the distribution of δ is a priori available. This problem has been first studied in [8,12] and then extended in various directions including constraint violation, [13], regularization, [11], non-convex optimization, [1,26,23], multi-stage problems, [46], and risks at various empirical levels, [17]. Moreover, papers [16,37] introduce algorithms to attain a solution carrying reduced risks. See also [36,29,4,30,35,51] for studies on the connection between scenario optimization and chance-constrained problems. All these results have put the scenario approach on solid quantitative grounds, a fact that has had a role in the widespread acceptance of this methodology in various application domains. In the next subsection, we specifically describe the mathematical results that are relevant to place the contribution of this paper in context, and then introduce the new perspective of this paper.

1.2 Previous results and the approach of this paper

In the paper [12], the fundamental relation

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

[3] Throughout, we assume that a solution exists. If more than one solution exists, a solution is singled out by means of a convex tie-break rule according to the approach of [7].

has been established. Equation (2) bounds the cumulative probability distribution of $V(x_N^*)$, the bound is universally valid for any scenario program in the form of (1) and, importantly, it is not improvable since it is exact (i.e. $\mathbb{P}^N\{V(x_N^*) \leq \epsilon\} = 1 - \sum_{i=0}^{d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$) for a class of problems, the so-called "fully-supported" problems. The right-hand side of (2) is the cumulative distribution of a Beta variable with $(d, N-d+1)$ degrees of freedom. Figure 1 displays in dashed blue line the corresponding density when $d = 400$ and $N = 1000$ (respectively, number of optimization variables and number of scenarios). In the same figure one can also see

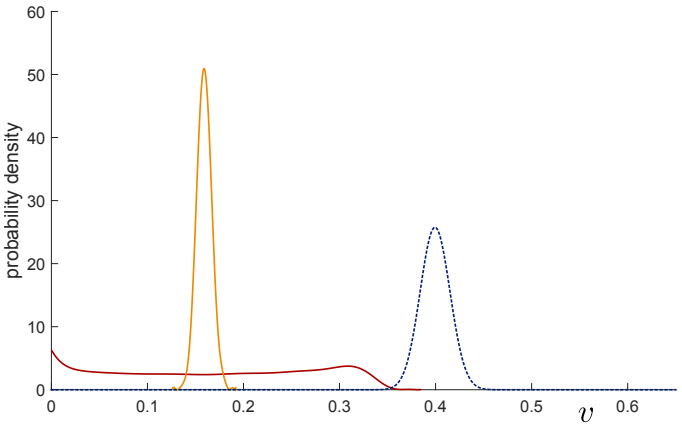

Fig. 1 Beta probability density for $d = 400$, $N = 1000$ (dashed blue) and probability density of $V(x_N^*)$ for two non fully-supported scenario programs (solid brown and red).

in solid brown and red lines the density of $V(x_N^*)$ for two scenario programs that are not fully-supported (these two programs are presented in detail in the simulation Section 3). For these problems, equation (2) holds with strict inequality.

In the later paper [14], it was observed that optimization problems encountered in applications are often not fully-supported, [48,47,42,49]. Moreover, by counting the number of support constraints for the case at hand, one knows that the problem is not fully-supported, and, hence, can be unwilling to use the theory of [12] that is tight for fully-supported problems only. Based on this observation, a new approach was introduced in [14] where one waits before forming an evaluation on $V(x_N^*)$, and the evaluation is based on the number of support constraints that have been found in the scenario program at hand.

The present paper builds on the approach initiated in [14] and we herein fully develop a new theory for the study of the joint distribution of the risk and the complexity in scenario programs. This theory reveals a fundamental correlation structure that links the risk to the complexity and has important applicative implications. We start by making formal the concept of support constraint, taken from [7], and that of complexity.

Definition 2 (support constraint and complexity) A constraint $x \in \mathcal{X}_{\delta_i}$ of the scenario program (1) is called a *support constraint* if its removal changes the solution x_N^*. The complexity of the scenario program (1) is the number of its support constraints. □

In paper [7] it is shown that the number of support constraints of (1) is always less than or equal to d, the number of optimization variables, and, in case of fully-supported problems, (1) has d support constraints with probability 1, whenever

$N \geq d$.

Support constraints are always active constraints. The converse is not true in general, and an active constraint need not be a support constraint as it can be easily understood by considering a situation where, after finding the solution, one more active constraint is added: removing this constraint does not change the solution and this constraint is not of support. In extreme cases, situations can occur where no support constraints exist. When all the active constraints are support constraints, which is the typical case, keeping the support constraints and removing all the other constraints leaves the solution unchanged. Following [14], we call this situation non-degenerate. Non-degeneracy rules out situations in which the constraints accumulate anomalously with nonzero probability.

Definition 3 (non-degeneracy) Program (1) is called non-degenerate if its solution coincides with probability 1 (with respect to the sample δ_i, $i = 1, \ldots, N$) with the solution that is obtained after eliminating all the constraints that are not of support. □

If (1) is non-degenerate, we can reconstruct the solution x^*_N by only using the support constraints. The number of support constraints is therefore a measure of the complexity of representation of x^*_N in terms of a scenario program that has a reduced number of constraints. For short, we at times speak of "complexity of the solution" to mean the complexity of the program that has generated the solution.

Let s^*_N be the complexity of program (1) and $(s^*_N, V(x^*_N))$ be the bi-variate variable of complexity and risk. Since δ_i, $i = 1, \ldots, N$, are independent random elements from $(\Delta, \mathcal{F}, \mathbb{P})$, the N-dimensional sample $(\delta_1, \delta_2, \ldots, \delta_N)$ is a random element from $(\Delta^N, \mathcal{F}^N, \mathbb{P}^N)$ (note that the probability is a product probability due to independence of δ_i, $i = 1, \ldots, N$), and so $(s^*_N, V(x^*_N))$ is a bi-variate random variable over $(\Delta^N, \mathcal{F}^N, \mathbb{P}^N)$. In this paper we study the distribution of $(s^*_N, V(x^*_N))$, that is, the joint distribution of complexity and risk, and Theorem 1 establishes a deep-seated result that this distribution is concentrated so that the risk $V(x^*_N)$ can be estimated from the complexity s^*_N. Figure 2 displays a 99% region for the distribution of $(s^*_N, V(x^*_N))$ when $d = 800$ and $N = 2000$. Given the value of s^*_N, which can be

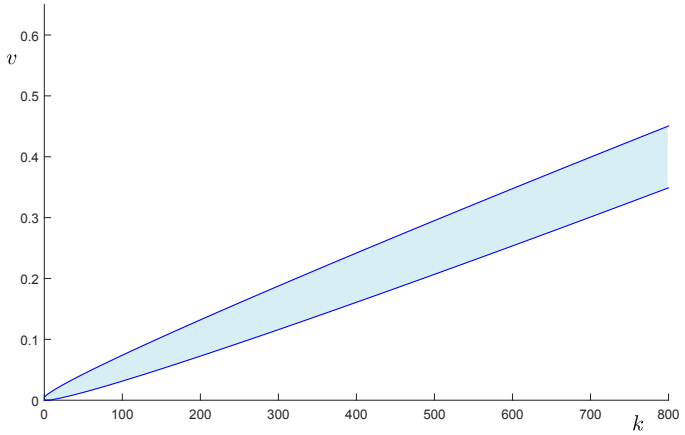

Fig. 2 99% region for the distribution of $(s^*_N, V(x^*_N))$ when $d = 800$ and $N = 2000$. Horizontal axis: value of s^*_N; vertical axis: value of $V(x^*_N)$.

easily computed[4], $V(x_N^*)$ is estimated to be in the line segment that is obtained by cutting the 99% region with the vertical line that originates from the value of s_N^*. Importantly, the result in Theorem 1 holds independently of the probability $\mathbb{P}$ so that it can be used without requiring any additional information on how δ distributes. In simple words, this means that the complexity carries universal information for the judgement of the risk, and this fact makes the theory of this paper broadly usable in applications where the underlying distribution that generates the δ's is not or only partly known.

Before closing this opening section we feel advisable to make explicit two facts that are consequences of what has been discussed so far.
(i) First we would like to compare, and better contrast, the result from [12] with that of this paper. Figure 3 displays the distribution of $(s_N^*, V(x_N^*))$ for an example discussed in Section 3, along with the corresponding marginal distribution of $V(x_N^*)$ (brown line in the figure). The marginal has been already displayed in Figure 1. Similarly, Figure 4 shows the distribution of $(s_N^*, V(x_N^*))$ for another example also discussed in Section 3, and the marginal of $V(x_N^*)$ is that displayed in red in Figure 1. The two bi-variate distributions are pretty different, but both are concentrated

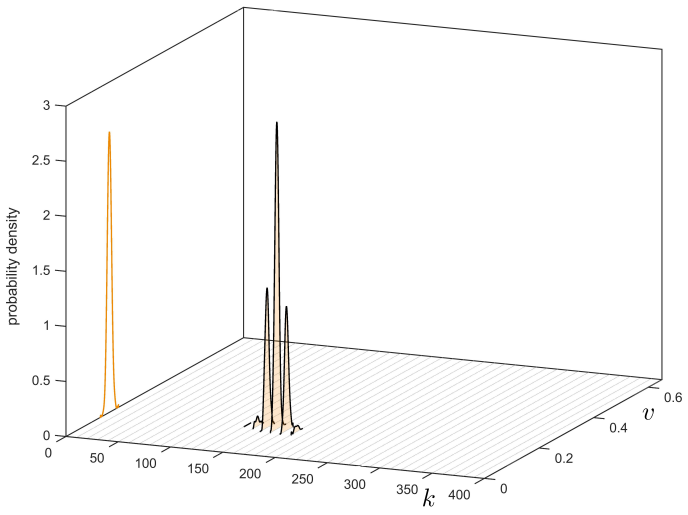

Fig. 3 Distribution of $(s_N^*, V(x_N^*))$ for another example discussed in Section 3.

in the 99% region of Figure 2. The to marginals for $V(x_N^*)$ have a dissimilar shape. As a result, if one studies the distribution of the sole $V(x_N^*)$, as is done in paper [12], then the problem arises that various behaviors are encountered depending on the scenario program at hand, so that tight results valid for all cases are not possible. On the other hand, adopting the broader point of view of studying jointly s_N^* and $V(x_N^*)$ sheds light on the structure of dependence between these two variables and the value of the hidden variable $V(x_N^*)$ can be estimated from the value of the easy-to-measure variable s_N^* independently of all other elements that characterize each single problem.
(ii) Strictly connected with point (i), a second fact is worth mentioning. For a given problem, one can investigate a number of specificities beyond complexity, which

[4] To this purpose, it is enough to eliminate one by one the constraints and recompute the solution, the scenario constraints are those whose elimination determines a change in the solution.

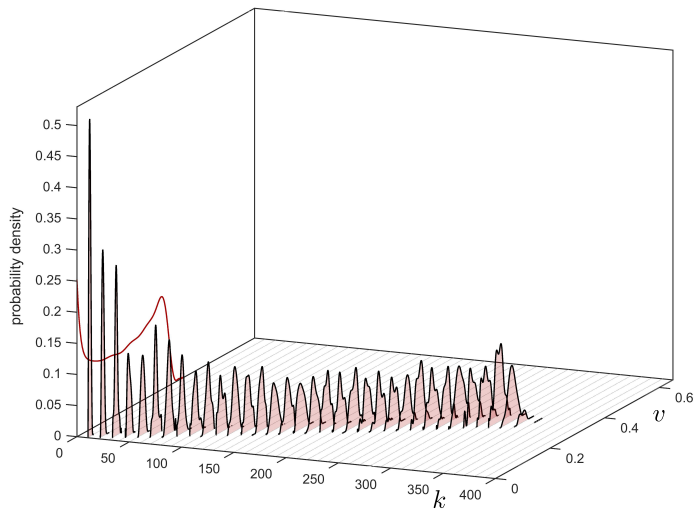

Fig. 4 Distribution of $(s_N^*, V(x_N^*))$ for an example discussed in Section 3.

includes the shape of the constraints, its location in the optimization domain etc. and possibly use prior knowledge to refine the judgement of the risk. When doing so one has to keep in mind that the margin of improvement in the judgment of the risk is limited to only reducing the remaining spread in the value of $V(x_N^*)$ after having evaluated s_N^* which already contains much of the information to judge the value of $V(x_N^*)$.

1.3 Structure of the rest of paper

In the next section, we introduce the main result, Theorem 1, and discuss its use. An example is provided in Section 3 that aims to illustrate the theory. In Section 4, we broaden our point of view even beyond optimization and show that the theory of this paper carries over to more general decision problems. We show in particular that this allows one to deal with partial violation of the scenario constraints, which leads to a more flexible scheme for optimization where one can trade risk for performance. All the proofs are presented in Section 5.

2 Main result and its practical use

Studying the program (1) that is based on N scenarios requires considering infinitely many other scenario programs that have the same structure as (1) with, however, an arbitrary number of scenarios:

$$\begin{aligned} &\min_{x\in\mathcal{X}} \ c^T x \\ &\text{subject to: } \ x \in \bigcap_{i=1,\ldots,m} \mathcal{X}_{\delta_i}, \end{aligned} \tag{3}$$

where $m = 0, 1, 2, \ldots$ is any integer and it is meant that the "subject to" line is dropped when $m = 0$, and δ_i, $i = 1, \ldots, m$, is an independent sample from $(\Delta, \mathcal{F}, \mathbb{P})$. Hence, notice that (1) is the same as (3) when $m = N$.

Assumption 1 (existence and uniqueness) *For every m and for every sample δ_i, $i = 1, \ldots, m$, program (3) 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. This is the same approach as in [7]. The so-obtained solution is indicated with x^*_m and it will be simply called the "solution" of (3).* □

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. $t_1(x) = x_1$, $t_2(x) = x_2$, ...

The definition of support constraint in Definition 2 extends in an obvious manner to (3): a constraint $x \in \mathcal{X}_{\delta_i}$ of (3) is called a support constraint if its removal changes the solution x^*_m. We introduce the following non-degeneracy assumption.

Assumption 2 (non-degeneracy) *For every m, the solution x^*_m to program (3) coincides with probability 1 (with respect to the sample δ_i, $i = 1, \ldots, m$) 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. More comments on this assumption is provided in Section 4.1 where it is shown that this assumption can be relaxed in a more abstract set-up.

The following theorem is our main result in this first part of the paper, it characterizes the support of the distribution of the bi-variate variable $(s^*_N, V(x^*_N))$, where s^*_N is the complexity of the solution (the number of support constraints) and $V(x^*_N)$ is its risk (the probability that a new constraint is not satisfied).

Theorem 1 *Consider program (1) with $N > d$. Given a "confidence parameter" $\beta \in (0, 1)$, for any $k = 0, 1, \ldots, d$ consider the polynomial equation in the t variable*

$$\binom{N}{k} t^{N-k} - \frac{\beta}{2N} \sum_{i=k}^{N-1} \binom{i}{k} t^{i-k} - \frac{\beta}{6N} \sum_{i=N+1}^{4N} \binom{i}{k} t^{i-k} = 0. \tag{4}$$

This equation has exactly two solutions in $[0, +\infty)$, which we denote with $\underline{t}(k)$ and $\overline{t}(k)$ ($\underline{t}(k) \leq \overline{t}(k)$) . Let $\underline{\epsilon}(k) := \max\{0, 1 - \overline{t}(k)\}$ and $\overline{\epsilon}(k) := 1 - \underline{t}(k)$. Under Assumptions 1 and 2, it holds that

$$\mathbb{P}^N\{\underline{\epsilon}(s^*_N) \leq V(x^*_N) \leq \overline{\epsilon}(s^*_N)\} \geq 1 - \beta. \tag{5}$$

□

Proof: The proof is given in Section 5.

The theorem assigns lower and upper bounds on $V(x^*_N)$ that hold with high confidence $1-\beta$. The bounds depend on the random variable s^*_N, the number of support constraints, which can be assessed by the user after computing the solution x^*_N. A more explicit way of writing (5) is

$$\mathbb{P}^N \left(\bigcup_{k=0}^{d} \{s^*_N = k \text{ and } \underline{\epsilon}(k) \leq V(x^*_N) \leq \overline{\epsilon}(k)\} \right) \geq 1 - \beta. \tag{6}$$

Hence, a user who computes the number of support constraints and claims the risk to be between the limits $\underline{\epsilon}(k)$ and $\overline{\epsilon}(k)$ (where k is the assessed number of support

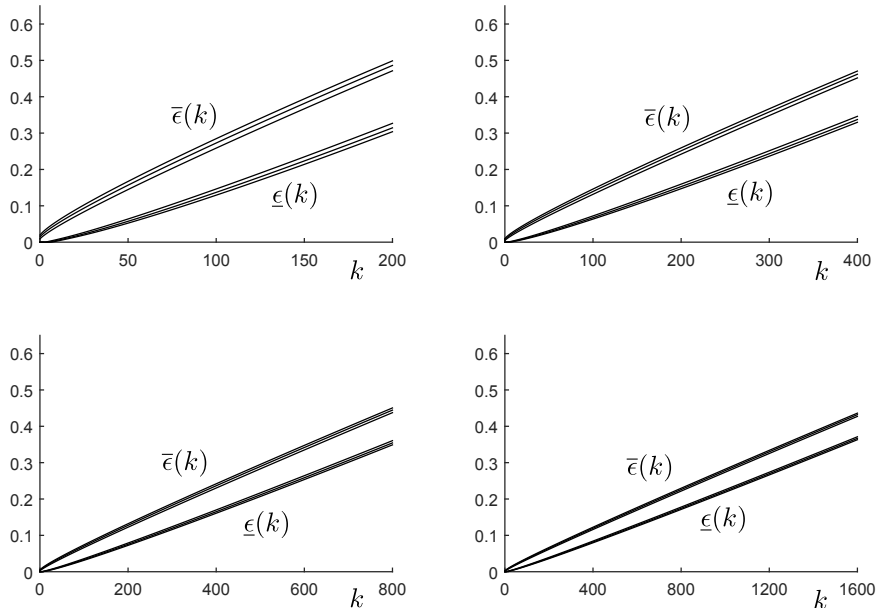

Fig. 5 Profile of curves $\underline{\epsilon}(k)$ and $\overline{\epsilon}(k)$. (a) $N = 500$, $d = 200$; (b) $N = 1000$, $d = 400$; (c) $N = 2000$, $d = 800$; (d) $N = 4000$, $d = 1600$. The three curves are for $\beta = 0.1$, $\beta = 0.01$; $\beta = 0.001$.

constraints) incurs a probability $1 - \beta$ of making a wrong statement. This result holds true for all convex scenario programs in the form of (1). Figure 5 shows the confidence regions obtained from Theorem 1 for various values of N and d. For practical values of N and d, the two functions $\underline{\epsilon}(\cdot)$ and $\overline{\epsilon}(\cdot)$ are close enough to each other so that, after evaluating s^*_N, tight and useful information about $V(x^*_N)$ is obtained. As N grows, the two functions get progressively closer and eventually converge to one another.

2.1 Distribution-free has little cost

As we have seen, Theorem 1 applies independently of the distribution $\mathbb{P}$, that is, it is a distribution-free result. Hence, it can be used without knowledge of $\mathbb{P}$, a fact that plays an important role in applications. In this section we show that there are problems with a completely specified probability $\mathbb{P}$ where the bi-variate variable $(s^*_N, V(x^*_N))$ has a distribution whose support is not too dissimilar to what is found by applying Theorem 1. Hence, the price paid for a distribution-free result is small relatively to knowing that one of these problems is being run. The interpretation is that the number of support constraints carries the fundamental information to judge the risk, and the residual uncertainty in the risk after that the number of support constraints has been seen (two samples of scenarios that lead to the same number of support constraints may carry a different risk) is only marginally increased by requiring that the result can be applied with no knowledge on $\mathbb{P}$.

To put the above discussion on solid grounds, consider a fully-supported problem in dimension k, [12]. For such a problem, the number of support constraints is k with probability 1. It is not hard to embed this problem into another one that has d optimization variables while the problem continues to have k support constraints with probability 1, so that $s^*_N = k$ with probability 1. If we now apply Theorem 1 in [12] to this problem we see that the distribution of $V(x^*_N)$ is a Beta$(k, N-k+1)$ distribution, i.e., $\mathbb{P}^N\{V(x^*_N) \leq \epsilon\} = 1 - \sum_{i=0}^{k-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$. Now let $\underline{\alpha}(k)$ be the

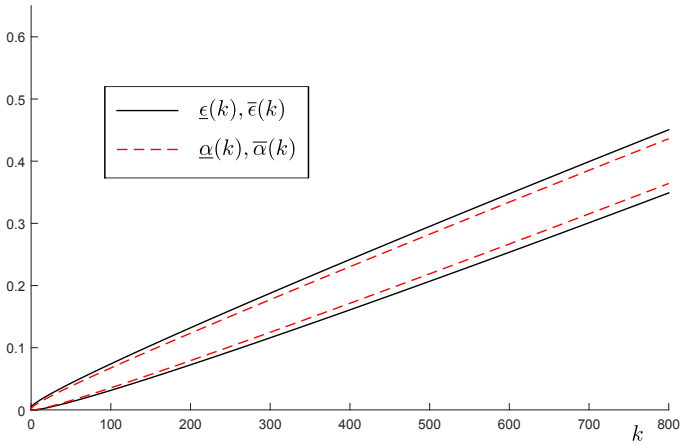

Fig. 6 Profile of curves $\underline{\epsilon}(k)$, $\overline{\epsilon}(k)$ and $\underline{\alpha}(k)$, $\overline{\alpha}(k)$, $N = 2000$, $d = 800$, $\beta = 0.001$.

value such that $1 - \sum_{i=0}^{k-1} \binom{N}{i} \underline{\alpha}(k)^i (1 - \underline{\alpha}(k))^{N-i} = \beta$ (i.e., $\underline{\alpha}(k)$ is the threshold value that clips the left tail of the Beta distribution with probability β) and, similarly, let $\overline{\alpha}(k)$ be the threshold value that clips the right tail with probability β, i.e., $1 - \sum_{i=0}^{k-1} \binom{N}{i} \overline{\alpha}(k)^i (1 - \overline{\alpha}(k))^{N-i} = 1 - \beta$. In order for equation (6) to be true for this problem it is necessary that

$$\underline{\epsilon}(k) \leq \underline{\alpha}(k) \quad \text{and} \quad \overline{\alpha}(k) \leq \overline{\epsilon}(k), \tag{7}$$

for, otherwise, just one side on the inequality $\underline{\epsilon}(k) \leq V(x_N^*) \leq \overline{\epsilon}(k)$ would be violated with a probability that exceeds β. Figure 6 profiles $\underline{\epsilon}(k)$ and $\overline{\epsilon}(k)$ against $\underline{\alpha}(k)$ and $\overline{\alpha}(k)$. The fact that the curves are close to each other shows that the price payed for a distribution free result is marginal as compared to only considering fully-supported problems.

2.2 Computational aspects

The discussion in the previous section suggests an easy way to compute function $\underline{t}(\cdot)$ and $\overline{t}(\cdot)$ in Theorem 1. The two relations in (7) give respectively $\overline{t}(k) \geq 1 - \underline{\alpha}(k)$ and $\underline{t}(k) \leq 1 - \overline{\alpha}(k)$. Hence, the two solutions of equation (4) must lie in the two bold intervals in Figure 7. To determine $\underline{t}(k)$, a bisection procedure can be run

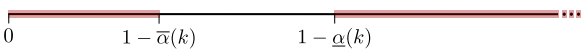

Fig. 7 Intervals to which $\underline{\epsilon}(k)$ and $\overline{\epsilon}(k)$ belong.

starting from the extreme points 0 and $1 - \overline{\alpha}(k)$, while, to determine $\overline{t}(k)$, one first checks if polynomial (4) has the same sign in $1 - \underline{\alpha}(k)$ and 1 (in which case one comes to know that $\overline{t}(k) > 1$ so that $\underline{\epsilon}(k)$ in Theorem 1 has value 0) and, when the signs in $1 - \underline{\alpha}(k)$ and 1 are different, a bisection procedure with initial extreme points $1 - \underline{\alpha}(k)$ and 1 is run to find $\overline{t}(k)$. A MATLAB code implementing these two bisection procedures is provided in Appendix A. The reader can cut and paste into the MATLAB workspace the code for a handy implementation.

3 An example

1000 points p_i are independently extracted in $\mathbb{R}^{400}$ according to a probability density $\mathbb{P}$ and presented to us. We want to translate the negative orthant in $\mathbb{R}^{400}$ (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 solve the scenario program

$$\begin{aligned} &\min_{x\in\mathbb{R}^{400}} \sum_{j=1}^{400} x_j \\ &\text{subject to: } x_j \geq p_{i,j}, \quad i = 1,\ldots,N, \end{aligned} \tag{8}$$

where j denotes component.

Program (8) was solved for two different probability densities $\mathbb{P}$. In the first case,

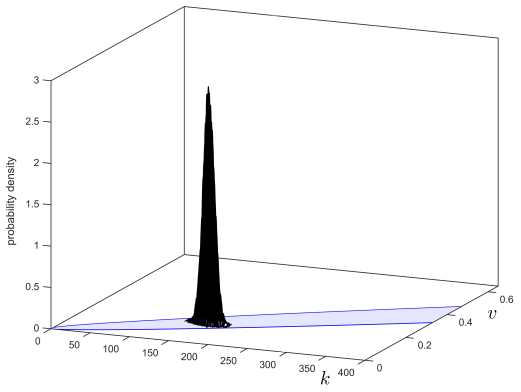

Fig. 8 Distribution of $(s_N^*, V(x_N^*))$ for the the first probability density, $N = 1000$, $d = 400$. The blue region is the 99.9% region given by Theorem 1.

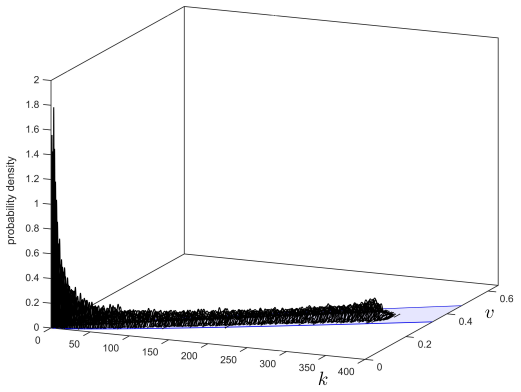

Fig. 9 Distribution of $(s_N^*, V(x_N^*))$ for the second probability density, $N = 1000$, $d = 400$. The blue region is the 99.9% region given by Theorem 1.

the points were given by relation $p_i = q_i + c_i$ where the q_i's were independently

drawn from a 400-dimensional Gaussian distribution with unitary variance, $G(0, I)$, and the c_i's were vectors with 400 equal components taken from $[0, 5]$ with uniform distribution. In the second case, the p_i's were again given by $p_i = q_i + c_i$ with the q_i's generated as in the previous case but the c_i's were this time vectors of equal components with value 0 with probability 99% and a value taken from a Gaussian distribution $G(0, 4)$ with probability 1%. Program (8) was solved 100000 times for both probability densities and each time the values of s^*_{1000} and $V(x^*_{1000})$ were recorded. This gave the empirical distributions reported in Figures 8 and 9,[5] where the region given by Theorem 1 for $\beta = 0.001$ is also displayed.

For the second probability density, the simulation set-up as above was then repeated for $N = 2000$ and $N = 4000$, while all other quantities were kept the same as for $N = 1000$. The results are on display in Figures 10 and 11. One can notice

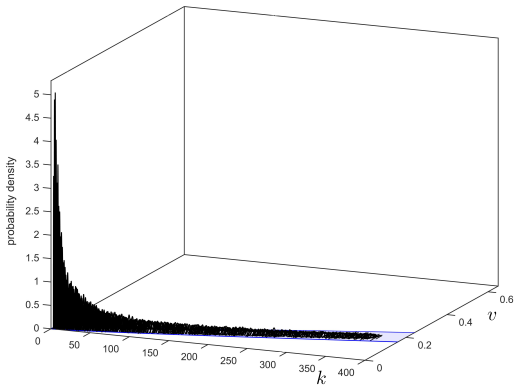

Fig. 10 Distribution of $(s^*_N, V(x^*_N))$ for the second probability density, $N = 2000$, $d = 400$. The blue region is the 99.9% region given by Theorem 1.

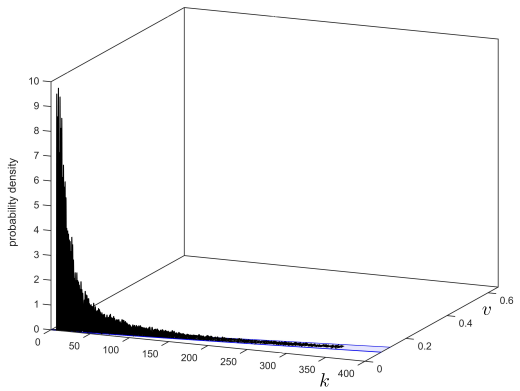

Fig. 11 Distribution of $(s^*_N, V(x^*_N))$ for the second probability density, $N = 4000$, $d = 400$. The blue region is the 99.9% region given by Theorem 1.

[5] These are the same empirical distributions as in Figures 3 and 4 (in Figures 3 and 4 only some values of k were displayed).

the evolution of the empirical distribution, which, for any given k, tends to move towards lower values of the risk while the random spreads also progressively reduce.

4 A more abstract theory

The theory developed in previous sections can be carried over to a more abstract set-up which incorporates only the salient features that play a role in the derivation of the results. Such an abstract theory is presented here, followed by two application examples that illustrate its usefulness: in Section 4.1, the upper bound $V(x_N^*) \leq \bar{\epsilon}(s_N^*)$ is studied under a more general condition than the non-degeneracy Assumption 2, while Section 4.2 deals with the possibility of relaxing the constraints so as to improve the value of the cost function.

In analogy with Section 2, for any given $m = 0, 1, 2, \ldots$ start by considering an independent sample δ_i, $i = 1, \ldots, m$, from $(\Delta, \mathcal{F}, \mathbb{P})$. The optimization domain $\mathcal{X}$ of Section 2 is here substituted by a generic set $\mathcal{Z}$, called the "decision set".[6]. To any δ there is associated a set $\mathcal{Z}_\delta \subseteq \mathcal{Z}$. In Section 2, x_m^* was generated by program (3), that is, program (3) there defined maps (one for any m) from Δ^m to $\mathcal{X}$. Here, we consider generic maps

$$M_m : \Delta^m \to \mathcal{Z}, \quad m = 0, 1, 2, \ldots$$

and write $z_m^* = M_m(\delta_1, \ldots, \delta_m)$. The interpretation is that z_m^* is a decision made according to a rule M_m applied to a set of scenarios $\delta_1, \ldots, \delta_m$. Throughout this Section 4 the word "decision" refers to z_m^*, and the symbol "z" will only be used in relation to a decision.

The following assumption applies to M_m.

Assumption 3 (properties on M_m)
(i) M_m is permutation-invariant: $M_m(\delta_1, \ldots, \delta_m) = M_m(\delta_{i_1}, \ldots, \delta_{i_m})$ if $\delta_{i_1}, \ldots, \delta_{i_m}$ is a permutation of $\delta_1, \ldots, \delta_m$;
(ii) given m values δ_i, $i = 1, \ldots, m$, augment them with n more values $\delta_{m+1}, \ldots, \delta_{m+n}$, where m and n are generic integers. If the decision z_m^* obtained from the first m values of δ is in the sets associated with the extra n values of δ, that is, $z_m^* \in \mathcal{Z}_{\delta_i}$ for $i = m+1, \ldots, m+n$, then it holds that $M_{m+n}(\delta_1, \ldots, \delta_{m+n}) = z_m^*$, that is, the decision obtained after adding the new δ's remains unchanged.
(iii) instead, if δ_i, $i = 1, \ldots, m$, is augmented with values $\delta_{m+1}, \ldots, \delta_{m+n}$ such that one or more δ_i, $i = m+1, \ldots, m+n$, has associated a set which does not contain z_m^*, that is, $z_m^* \notin \mathcal{Z}_{\delta_i}$ for one or more $i = m+1, \ldots, m+n$, then it holds that $M_{m+n}(\delta_1, \ldots, \delta_{m+n}) \neq z_m^*$. □

Notice that Assumption 3 does not require that z_m^* is in $\mathcal{Z}_{\delta_i}$, $i = 1, \ldots, m$.

The notion of risk of a $z \in \mathcal{Z}$ is an obvious extension from Section 2: $V(z) = \mathbb{P}\{\delta \in \Delta : z \notin \mathcal{Z}_\delta\}$. Also the notion of support constraint carries over from Section 2, but we here prefer to speak of "support element" since, as we have remarked above, z_m^* is not forced to be in $\mathcal{Z}_{\delta_i}$, $i = 1, \ldots, m$, that is the $\mathcal{Z}_{\delta_i}$'s do not act here as constraints: $\mathcal{Z}_{\delta_i}$ is called a *support element* if $M_{m-1}(\delta_1, \ldots, \delta_{i-1}, \delta_{i+1}, \ldots, \delta_m) \neq M_m(\delta_1, \ldots, \delta_m)$. The number of support elements is denoted by $\tilde{s}_m^*$. Finally, the assumption of non-degeneracy remains essentially unchanged.

[6] $\mathcal{Z}$ can be any set, without any Euclidean structure. We change notation from $\mathcal{X}$ to $\mathcal{Z}$ because in some applications $\mathcal{Z}$ is the same as $\mathcal{X}$ augmented with extra elements; concrete examples of decision sets are provided in Sections 4.1 and 4.2

Assumption 4 (non-degeneracy of M_m)
For every m, the decision $z^*_m = M_m(\delta_1, \ldots, \delta_m)$ coincides with probability 1 (with respect to the sample δ_i, $i = 1, \ldots, m$) with the decision that is obtained after eliminating all the elements that are not of support. □

The fact that the set-up of this section encompasses as a particular case that of Section 2 is shown in Section 5.2.

We are now ready to state the main theorem of this section, Theorem 2. The essence of this theorem is that the thesis of Theorem 1 carries over to the present more abstract set-up. One difference with Theorem 1 is that in Theorem 2 the number of support elements is not a-priori upper bounded (as it was the number of support constraints in Theorem 1, which could not exceed d), so that we here have to also account for the case $k = N$ (number of support elements equal to the number of scenarios), which leads to considering equation (10).

Theorem 2 *Given a "confidence parameter" $\beta \in (0, 1)$, for any $k = 0, 1, \ldots, N-1$ consider the polynomial equation in the t variable*

$$\binom{N}{k} t^{N-k} - \frac{\beta}{2N} \sum_{i=k}^{N-1} \binom{i}{k} t^{i-k} - \frac{\beta}{6N} \sum_{i=N+1}^{4N} \binom{i}{k} t^{i-k} = 0, \tag{9}$$

and, for $k = N$, consider the polynomial equation

$$1 - \frac{\beta}{6N} \sum_{i=N+1}^{4N} \binom{i}{k} t^{i-N} = 0. \tag{10}$$

For any $k = 0, 1, \ldots, N-1$ equation (9) has exactly two solutions in $[0, +\infty)$, which we denote with $\underline{t}(k)$ and $\overline{t}(k)$ ($\underline{t}(k) \leq \overline{t}(k)$). Instead, equation (10) has only one solution in $[0, +\infty)$, which we denote with $\overline{t}(N)$, while we define $\underline{t}(N) = 0$. Let $\underline{\epsilon}(k) := \max\{0, 1 - \overline{t}(k)\}$ and $\overline{\epsilon}(k) := 1 - \underline{t}(k)$, $k = 0, 1, \ldots, N$. Under Assumptions 3 and 4, it holds that

$$\mathbb{P}^N\{\underline{\epsilon}(\tilde{s}^*_N) \leq V(z^*_N) \leq \overline{\epsilon}(\tilde{s}^*_N)\} \geq 1 - \beta. \tag{11}$$

□

Proof: The proof is given in Section 5.

Theorem 2 allows one to evaluate the risk for decision problems of various type. We next apply this Theorem 2 to two specific setups.

4.1 Application no.1: A theorem for the scenario program (1) with a relaxed degeneracy condition

In Section 2, Theorem 1 was proven under the non-degeneracy Assumption 2. We here show that a statement on the generalization properties of (1) can be obtained under the milder condition that the instance at hand of (1) is non-degenerate, a condition that can be directly verified on the program that is being run as opposed to Assumption 2 which pertains to all choices of scenario samples.

Given a sample δ_i, $i = 1, \ldots, m$, the scenario program (3) defines a solutions x^*_m, possibly after a tie-break rule is applied as indicated in Assumption 1. Here, we introduce a decision z^*_m generated by (3) which consists of x^*_m augmented with the values of the δ_i's that correspond to active constraints of (3), where each of these

δ_i's is equipped with an integer number that indicates how many times the same value of δ_i has been seen (that is, if δ_1 is active, δ_3 takes on the same value as δ_1, and no other δ_i takes on the same value, then the value of δ_1 is included in the decision, followed by the number 2). Hence, formally,

$$z_m^* = \big(x_m^*, \big\{(\delta; n) \text{ where } \delta = \delta_i \text{ for some } i \in \{1, \dots, m\} \text{ such that } \mathcal{X}_{\delta_i} \text{ is active for problem (3), and } n = \text{ number of times with which the same value } \delta \text{ appears in (3)}\big\}\big),$$

and $\mathcal{Z}$ is the set of all $z = (x, \{(\delta^{(j)}; n^{(j)}),\ j = 1, \dots, p\})$ for some integer $p \geq 0$, where $x \in \mathcal{X}$, $\delta^{(j)} \in \Delta$, $n^{(j)} \in \mathbb{N}$, $j = 1, \dots, p$. Given a δ, let $\mathcal{Z}_\delta = \{z \in \mathcal{Z} :\ x \in \mathcal{X}_\delta \text{ and } \mathcal{X}_\delta \text{ is non-active at } x\}$. Hence, $V(z)$ is defined as the probability of the set of δ's such that either $x \notin \mathcal{X}_\delta$ or $\mathcal{X}_\delta$ is active at x. Clearly, $V(x_m^*) \leq V(z_m^*)$ as the latter also includes active constraints. An easy inspection shows that Assumption 3 holds with these definitions. Moreover, the support elements are here those associated to the δ_i's such that $\mathcal{X}_{\delta_i}$ is active at x_m^* and it follows immediately that the non-degeneracy Assumption 4 also holds. Hence, Theorem 2 can be applied to this context to obtain a result on $V(z_N^*)$. From this result we next show that a statement on $V(x_N^*)$ can be obtained. First, since $V(x_m^*) \leq V(z_m^*)$, we drop in the event of equation (11) the left inequality and write $\mathbb{P}^N\{V(x_N^*) \leq \overline{\epsilon}(\tilde{s}_N^*)\} \geq 1 - \beta$ or, equivalently, $\mathbb{P}^N\{V(x_N^*) > \overline{\epsilon}(\tilde{s}_N^*)\} \leq \beta$. Next, we want to express a judgement on $V(x_N^*)$ only when the active constraints coincide with the support constraints of (1) (which implies that the instance at hand of (1) is non-degenerate). Note that the validity of this condition is verified by inspecting program (1) for the sample δ_i, $i = 1, \dots, N$, that is being used and this is very different from enforcing the non-degeneracy Assumption 2 which refers to all possible extractions of m constraints. When in the instance at hand of program (1) the active constraints coincide with the support constraints, $\tilde{s}_N^*$ in Theorem 2 (number of active constraints) coincides with s_N^* in Theorem 1 (number of support constraints). Moreover, if $N > d$, then we certainly have $\tilde{s}_N^* = s_N^* \leq d$, so that the polynomial equation (10) in Theorem 2 can be dropped as it never happens that $\tilde{s}_N^* = N$. We have obtained the following corollary of Theorem 2.

Corollary 1 *Consider program* (1) *with* $N > d$. *Given a "confidence parameter"* $\beta \in (0, 1)$, *for any* $k = 0, 1, \dots, d$ *consider the polynomial equation in the* t *variable*

$$\binom{N}{k} t^{N-k} - \frac{\beta}{2N} \sum_{i=k}^{N-1} \binom{i}{k} t^{i-k} - \frac{\beta}{6N} \sum_{i=N+1}^{4N} \binom{i}{k} t^{i-k} = 0. \tag{12}$$

This equation has exactly two solutions in $[0, +\infty)$, *which we denote with* $\underline{t}(k)$ *and* $\overline{t}(k)$ $(\underline{t}(k) \leq \overline{t}(k))$. *Let* $\underline{\epsilon}(k) := \max\{0, 1 - \overline{t}(k)\}$ *and* $\overline{\epsilon}(k) := 1 - \underline{t}(k)$. *We say that the instance at hand of program* (1) *satisfies the condition "*$A = S$*" if the active constraints coincide with the support constraints (so that this instance is non-degenerate). Under Assumptions 1, it holds that*

$$\mathbb{P}^N\{V(x_N^*) > \overline{\epsilon}(s_N^*) \text{ and "}A = S\text{" holds }\} \leq \beta. \tag{13}$$

□

4.2 Application no.2: optimization with constraints relaxation

In this section, we consider scenario programs where, unlike (1), one is allowed to violate constraints for the purpose of improving the cost value. We assume that constraints violation has itself a cost and in the limit when this cost goes to infinity

the original problem (1) is recovered.

Matters of convenience suggest that constraints are written in this section as $f(x,\delta) \leq 0$, where $f(x,\delta)$ is a convex function in x for any given δ (referring back to the notation in Section 1, we therefore have that $\mathcal{X}_\delta = \{x : f(x,\delta) \leq 0\}$). The reason for this choice is that the value of the function f is used to express the "regret" for violating a constraint: for a given δ, the regret at x is $f(x,\delta)$ and the steepness of this function describes the marginal increase of regret when the solution is moved in a given direction. In this set-up, we consider the following scenario program, which generalizes (1):

$$\begin{aligned} \min_{\substack{x\in\mathcal{X} \\ \xi_i\geq 0, i=1,\ldots,N}} \quad & c^T x + \rho \sum_{i=1}^{N} \xi_i \\ \text{subject to:} \quad & f(x,\delta_i) \leq \xi_i, \quad i = 1,\ldots,N, \end{aligned} \tag{14}$$

where, as before, δ_i, $i = 1,\ldots,N$, is an independent random sample from Δ. Note that (14) has $d+N$ optimization variables, namely, x and ξ, $i = 1,\ldots,N$. If $\xi_i > 0$, the constraint $f(x,\delta_i) \leq 0$ is relaxed to $f(x,\delta_i) \leq \xi_i$ and this generates the regret ξ_i. Parameter ρ is used to set a suitable trade-off between the original cost function and the cost generated by the regret for violating constraints. When $\rho \to \infty$, one goes back to the original program (1) where no constraint violation is allowed.

The following assumption is the equivalent of the existence and uniqueness Assumption 1 for the generalized set-up of this section.

Assumption 5 (existence and uniqueness) *Consider programs as in* (14) *where* N *is substituted with an index* $m = 0, 1, \ldots$ *and* δ_i, $i = 1,\ldots,m$, *is an independent sample from* $(\Delta, \mathcal{F}, \mathbb{P})$. *For every* m *and for every sample* δ_i, $i = 1,\ldots,m$, *these programs admit solution. If for one of these programs 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.*[7] □

Moreover, we make the following assumption.

Assumption 6 *For every* x, $\mathbb{P}\{\delta : f(x,\delta) = 0\} = 0$. □

This is a non-accumulation assumption on functions $f(x,\delta)$, and, when constraints $f(x,\delta_i) \leq 0$ are enforced in the scenario program as is done in (1), it implies the non-degeneracy Assumption 2.

Let $x^*_m, \xi^*_{i,m}$, $i = 1,\ldots,m$, be the solution of (14) with m in place of N. The abstract theory of Section 4 does not apply directly to $x^*_m, \xi^*_{i,m}$, $i = 1,\ldots,m$, and we have first to define the concept of decision z^*_m. This is x^*_m augmented with the values of the δ_i's that correspond to constraints $f(x,\delta_i) \leq 0$ that are violated at the solution (i.e. $f(x^*_m,\delta_i) > 0$), where each of these δ_i's is equipped with an integer number that indicates how many times the same value of δ_i has been seen. Formally,

$$\begin{aligned} z^*_m = \big(x^*_m, \big\{&(\delta; n) \text{ where } \delta = \delta_i \text{ for some } i \in \{1,\ldots,m\} \text{ such that } \xi^*_{i,m} > 0, \text{ and} \\ &n = \text{number of times with which the same value } \delta \text{ appears in (14) with } m \\ &\text{in place of } N\big\}\big), \end{aligned}$$

[7] Note that only the tie with respect to x is broken by $t_1(x)$, $t_2(x)$, $t_3(x)$, ... On the other hand, for a given x^*_m the values of ξ_i, $i = 1,\ldots,m$, remain univocally determined at optimum by relation $\xi^*_{i,m} = f(x^*_m,\delta_i)$, so that no tie on ξ_i, $i = 1,\ldots,m$, can persist after the tie on x is broken.

and $\mathcal{Z}$ is the set of all $z = (x, \{(\delta^{(j)}; n^{(j)}),\ j = 1, \ldots, p\})$ for some integer $p \geq 0$, where $x \in \mathcal{X}$, $\delta^{(j)} \in \Delta$, $n^{(j)} \in \mathbb{N}$, $j = 1, \ldots, p$. We take as $\mathcal{Z}_\delta$ the set of $z \in \mathcal{Z}$ for which $f(x, \delta) \leq 0$ and $(\delta^{(j)}; n^{(j)}), j = 1, \ldots, p$, are arbitrary both in number (i.e. p is any positive integer) and value. Correspondingly, $V(z) = V(x, \{(\delta^{(j)}; n^{(j)}), j = 1, \ldots, p\})$ is defined as the probability of the set of δ's such that $x \notin \mathcal{X}_\delta$. Assumptions 3 and 4 are satisfied in this context, as we next show. Condition (i) in Assumption 3 is clearly true. To show (ii), argue as follows. If $z^*_m \in \mathcal{Z}_{\delta_i}$ for $i = m+1, \ldots, m+n$, then $f(x^*_m, \delta_i) \leq 0$ for $i = m+1, \ldots, m+n$. Hence, augmenting the solution of (14) with m in place of N with $\xi_i = 0$ for $i = m+1, \ldots, m+n$ gives a feasible point x^*_m, $\xi_i = \xi^*_{i,m}$, $i = 1, \ldots, m$, $\xi_i = 0$, $i = m+1, \ldots, m+n$ for (14) with $m+n$ in place of N that attains the same value as the optimal value of (14) with m in place of N. It is claimed that this is the optimal solution of (14) with $m+n$ in place of N. Indeed, if a better solution $\bar{x}, \bar{\xi}_i, i = 1, \ldots, m+n$ existed for (14) with $m+n$ in place of N, then $\bar{x}, \bar{\xi}_i, i = 1, \ldots, m$ would be superoptimal for (14) with m in place of N since the dropped $\bar{\xi}_i$, $i = m+1, \ldots, m+n$ give a non-negative contribution. To the optimal solution $x^*_{m+n} = x^*_m$, $\xi^*_{i,m+n} = \xi^*_{i,m}$, $i = 1, \ldots, m$, $\xi^*_{i,m+n} = 0$, $i = m+1, \ldots, m+n$ of (14) with $m+n$ in place of N there corresponds $z^*_{m+n} = (x^*_{m+n}, \{(\delta; n)$, where $\delta = \delta_i$ for some $i \in \{1, \ldots, m+n\}$ such that $\xi^*_{i,m+n} > 0$, and $n =$ number of times with which the same value δ appears in (14) with $m+n$ in place of $N\}) = (x^*_m, \{(\delta; n)$, where $\delta = \delta_i$ for some $i \in \{1, \ldots, m\}$ such that $\xi^*_{i,m} > 0$, and $n =$ number of times with which the same value δ appears in (14) with m in place of $N\})$, which is the same as z^*_m, showing the validity of (ii). Condition (iii) in Assumption 3 instead easily follows from the fact that if $z^*_m \notin \mathcal{Z}_{\delta_{\bar{i}}}$ for some $\bar{i} \in \{m+1, \ldots, m+n\}$, then either x^*_{m+n} has to move to a new location where $f(x, \delta_{\bar{i}}) \leq 0$ (and therefore $z^*_{m+n} \neq z^*_m$) or $\delta_{\bar{i}}$ has to be added to the solution z^*_m to obtain z^*_{m+n} (and, again $z^*_{m+n} \neq z^*_m$). Turn now to assess the non-degeneracy Assumption 4. Consider program (14) with m in place of N and eliminate a δ_i such that $f(x^*_m, \delta_i) < 0$. The decision associated with the remaining $m-1$ δ_i's is the same decision as that associated with the original program (14) with m in place of N because the constraint corresponding to the eliminated δ_i is non-active. Hence, none of the δ_i such that $f(x^*_m, \delta_i) < 0$ is of support. We further claim that all δ_i's such that $f(x^*_m, \delta_i) \geq 0$ are of support with probability 1. Eliminate any one of them from the program (14) with m in place of N. If the eliminated one is such that $f(x^*_m, \delta_i) > 0$, then the decision clearly changes, so that the δ_i is of support. Suppose instead that the eliminated one is such that $f(x^*_m, \delta_i) = 0$ and, for the sake of contradiction, suppose also that the decision does not change. It follows that x^*_m is obtained by a program that does not contain δ_i and, due to the independence of $\delta_1, \delta_2, \ldots, \delta_m$, it is easily seen that Assumption 6 implies that $f(x^*_m, \delta_i) = 0$ only happens with probability 0. Hence, with probability 1 the δ_i's such that $f(x^*_m, \delta_i) \geq 0$ are of support, and they give the original decision since the simultaneous elimination of the other δ_i's for which $f(x^*_m, \delta_i) < 0$ (non-active) does not change the decision. This means that the problem is non-degenerate.

Since all conditions of the abstract theory are satisfied, Theorem 2 applies. In the present context, $V(z^*_N) = V(x^*_N)$ and $\tilde{s}^*_N =$ number of δ_i such that $f(x^*_N, \delta_i) \geq 0$, which gives us the following corollary of Theorem 2.

Corollary 2 *Consider program* (14) *and let* x^*_N *be the* x *component of its solution. Given a "confidence parameter"* $\beta \in (0, 1)$*, for any* $k = 0, 1, \ldots, N-1$ *consider the polynomial equation in the* t *variable*

$$\binom{N}{k} t^{N-k} - \frac{\beta}{2N} \sum_{i=k}^{N-1} \binom{i}{k} t^{i-k} - \frac{\beta}{6N} \sum_{i=N+1}^{4N} \binom{i}{k} t^{i-k} = 0, \tag{15}$$

and for $k = N$ consider the polynomial equation

$$1 - \frac{\beta}{6N} \sum_{i=N+1}^{4N} \binom{i}{k} t^{i-N} = 0. \tag{16}$$

For any $k = 0, 1, \ldots, N-1$ equation (15) has exactly two solutions in $[0, +\infty)$, which we denote with $\underline{t}(k)$ and $\overline{t}(k)$ ($\underline{t}(k) \leq \overline{t}(k)$). Instead, equation (16) has only one solution in $[0, +\infty)$, which we denote with $\overline{t}(N)$, while we define $\underline{t}(N) = 0$. Let $\underline{\epsilon}(k) := \max\{0, 1 - \overline{t}(k)\}$ and $\overline{\epsilon}(k) := 1 - \underline{t}(k)$, $k = 0, 1, \ldots, N$. Under Assumptions 5 and 6, it holds that

$$\mathbb{P}^N\{\underline{\epsilon}(\tilde{s}_N^*) \leq V(x_N^*) \leq \overline{\epsilon}(\tilde{s}_N^*)\} \geq 1 - \beta. \tag{17}$$

where $\tilde{s}_N^*$ = number of δ_i such that $f(x_N^*, \delta_i) \geq 0$. □

Corollary 2 provides a quantitative evaluation of risk in the context of optimization with constraint relaxation. By a comparison with Theorem 1, we see that the number of support constraints s_N^* of the scenario program (1) is here substituted by $\tilde{s}_N^*$. $\tilde{s}_N^*$ accounts for the constraints of (1) that are violated, i.e. $f(x, \delta_i) > 0$, plus those that are active, i.e. $f(x, \delta_i) = 0$, at $x = x_N^*$. One can prove that $f(x_N^*, \delta_i) = 0$ in at most d cases with probability 1, showing that functions $\underline{\epsilon}(\cdot)$ and $\overline{\epsilon}(\cdot)$ have to be evaluated at an integer equal to the number of violated constraints plus at most an excess of d to compute the bound for the risk.

Example 1 A manufacturer produces goods in d different workplaces and x_j, $j = 1, \ldots, d$, is the quantity planned to be produced in workplace j. For the production, n different resources are employed. The quantity of resource k, $k = 1, \ldots, n$, used in workplace j to produce a unitary amount of goods is subject to random fluctuation and is denoted by $q_{j,k}(\delta)$. Each resource is available in a limited amount a_k. The goal of the manufacturer is to maximize the production while keeping low the probability of being in need of resources that exceed the available amount.

Assuming that a record $\{q_{j,k}(\delta_i),\ j = 1, \ldots, d,\ k = 1, \ldots, n\}$, $i = 1, \ldots, N$, of values for $\{q_{j,k}(\delta),\ j = 1, \ldots, d,\ k = 1, \ldots, n\}$ is available, the problem is modeled according to the scenario approach as follows

$$\min_{x_j \geq 0,\ j=1,\ldots,d} \ -\sum_{j=1}^{d} x_j$$
$$\text{subject to: } \begin{cases} \sum_{j=1}^{d} q_{j,1}(\delta_i) x_j \leq a_1, \\ \vdots \qquad\qquad\qquad\qquad i = 1, \ldots, N. \\ \sum_{j=1}^{d} q_{j,n}(\delta_i) x_j \leq a_n, \end{cases} \tag{18}$$

A simulation was performed with $d = 50$, $n = 2$ and $N = 2000$, which gave the result $-\sum_{j=1}^{50} x_{j,2000}^* = -16.66$, $s_{2000}^* = 4$. With the choice $\beta = 10^{-6}$, an application of Theorem 1 provided the following interval for the risk that the available amounts of resources are exceeded: $0 \leq V(x_N^*) \leq 0.014$.
Further, the manufacturer decides to increase the production and towards this goal accepts some rise in the risk of running out of resources. To design the new production strategy, the manufacturer uses the constraints relaxation approach presented

in this section and solve the optimization problem

$$\begin{aligned} \min_{\substack{x_j \geq 0, j=1,\dots,50 \\ \xi_i \geq 0, i=1,\dots,2000}} \quad & -\sum_{j=1}^{50} x_j + \rho \sum_{i=1}^{N} \xi_i \\ \text{subject to:} \quad & \left(\max \left\{ \sum_{j=1}^{50} q_{j,1}(\delta_i) x_j - a_1, \sum_{j=1}^{50} q_{j,2}(\delta_i) x_j - a_2, 0 \right\} \right)^2 \leq \xi_i \\ & i = 1, \dots, 2000. \end{aligned} \tag{19}$$

Notice that this is the same as (14) with[8]

$$f(x, \delta) = \left(\max \left\{ \sum_{j=1}^{50} q_{j,1}(\delta) x_j - a_1, \sum_{j=1}^{50} q_{j,2}(\delta) x_j - a_2, 0 \right\} \right)^2 . \tag{20}$$

As for the value of ρ, its selection can be tricky, because how ρ impacts on production/risk can be difficult to forecast, and we here refer to an approach that can be of general utility in other applications as well. The manufacturer sets out to solve (19) for an array of values of ρ. For each value, the production increase is calculated from the solution, while Corollary 2 gives an interval for the corresponding risk. Selecting $\beta = 10^{-6}$, the intervals that were found for 22 distinct values of ρ are displayed in Figure 12, where the abscissa gives the corresponding values of $\tilde{s}^*_{2000}$.[9] In the same

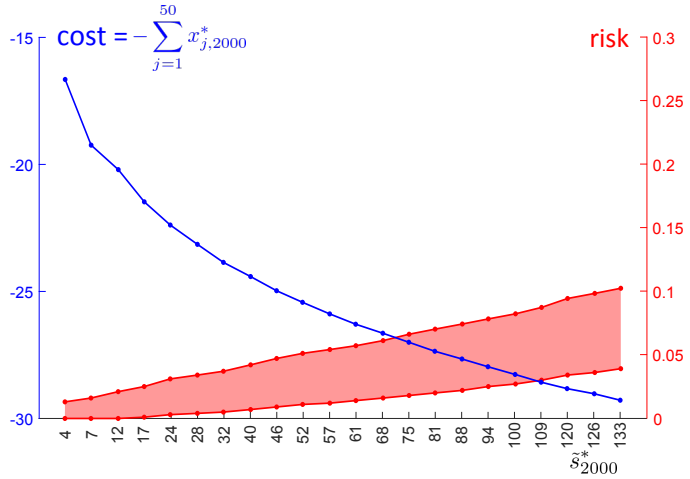

Fig. 12 Cost $= -\sum_{j=1}^{50} x^*_{j,2000}$ vs. risk. In abscissa is the the number $\tilde{s}^*_{2000}$ of scenarios for which $f(x^*_{2000}, \delta_i) \geq 0$.

[8] Notice that, strictly speaking, this choice of $f(x, \delta)$ does not satisfy Assumption 6. Reason is that setting to zero $f(x, \delta)$ when $\sum_{j=1}^{50} q_{j,1}(\delta) x_j - a_1$ and $\sum_{j=1}^{50} q_{j,2}(\delta) x_j - a_2$ are negative, as is done in (20), generates regions with positive volume in the domain in $\mathbb{R}^{50}$ for x where $f(x, \delta) = 0$. However, an easy inspection of the derivations in Subsection 4.2 shows that the requirement of Assumption 6 that, for every x, $\mathbb{P}\{\delta : f(x, \delta) = 0\} = 0$ can be relaxed to requiring that, for every x, $\mathbb{P}\{\delta : x \text{ is on the boundary of the constraint } \{f(x, \delta) \leq 0\}\} = 0$, and theory goes through unaltered with the only modifications that, throughout, "$f(x, \delta) = 0$" becomes "x is on the boundary of the constraint $\{f(x, \delta) \leq 0\}$", "$f(x, \delta) < 0$" becomes "x is in the interior of the constraint $\{f(x, \delta) \leq 0\}$", and "$f(x, \delta) \geq 0$" becomes "x violates or is on the boundary of the constraint $\{f(x, \delta) \leq 0\}$". While we have preferred in the general presentation the simpler formulation of Assumption 6, this second formulation leads to zero volume regions in the domain in $\mathbb{R}^{50}$ for x in the present example.

[9] Since the intervals in Figure 12 are obtained by a repeated application of Corollary (2), the confidence that $\underline{\epsilon}(\tilde{s}^*_{2000}) \leq V(x^*_{2000}) \leq \overline{\epsilon}(\tilde{s}^*_{2000})$ for all the 22 values of ρ simultaneously is $1 - 22 \cdot \beta$.

figure, the plot of cot $= -\sum_{j=1}^{50} x^*_{j,2000}$ is also profiled. A suitable trade-off between production and risk can be obtained by a direct inspection of the figure. In this case, the value of ρ that gives $\tilde{s}^*_{2000} = 46$ was selected, resulting in a good 50% of production increase, while the estimated interval for the risk moved from $[0, 0.014]$ to $[0.009, 0.047]$. □

5 Proofs

5.1 Proof of Theorem 2

We start by showing that equation (9) has two solutions in $[0, +\infty)$ and that equation (10) has one solution in $[0, +\infty)$.

Denote by $\varphi_k(t)$, $k = 0, 1, \ldots, N-1$, the polynomial in the left-hand side of (9) and by $\varphi_N(t)$ the polynomial in the left-hand side of (10), which we rewrite here by making explicit the number $H = 3N$ of terms in the rightmost summations:[10]

$$\varphi_k(t) = \begin{cases} \binom{N}{k} t^{N-k} - \frac{\beta}{2N} \sum_{i=k}^{N-1} \binom{i}{k} t^{i-k} - \frac{\beta}{2H} \sum_{i=N+1}^{N+H} \binom{i}{k} t^{i-k}, & 0 \le k < N \\ 1 - \frac{\beta}{2H} \sum_{i=N+1}^{N+H} \binom{i}{N} t^{i-N}, & k = N. \end{cases} \tag{21}$$

Let us start with $\varphi_N(t)$. By construction, $\varphi_N(0) = 1$ and $\varphi_N(t)$ is strictly decreasing with $\varphi_N(t) \to -\infty$ as $t \to +\infty$. Hence, $\varphi_N(t)$ has a unique root $\bar{t}(N)$ in $[0, +\infty)$, and, moreover,

$$\varphi_N(t) > 0 \text{ for } t \in [0, \bar{t}(N)), \text{ while } \varphi_N(t) < 0 \text{ for } t \in (\bar{t}(N), +\infty). \tag{22}$$

Turn now to $\varphi_k(t)$, $k = 0, 1, \ldots, N-1$. Notice first that the following recursive equation holds

$$\varphi_k(t) = -\frac{\beta}{2N} + (k+1) \int_0^t \varphi_{k+1}(\tau)\, \mathrm{d}\tau, \quad k = 0, 1, \ldots, N-1, \tag{23}$$

as it can be verified by a direct calculation. Using (23), we want to show that all the $\varphi_k(t)$, $k = 0, 1, \ldots, N-1$, follow the same pattern: a. $\varphi_k(0) = -\frac{\beta}{2N} < 0$; b. with the sole exception of $\varphi_{N-1}(t)$ that is increasing in $t = 0_+$, $\varphi_k(t)$ is initially decreasing, then it is increasing with $\varphi_k(t) > 0$ at its maximum, and then decreasing again; c. $\varphi_k(t) \to -\infty$ as $t \to +\infty$. Facts a,b,c are obtained by using (23) for $k = N-1$ (and recalling the properties of $\varphi_N(t)$), and then proceeding backward, $k = N-2, N-3, \ldots, 0$, where the only point that deserves an explanation is that $\varphi_k(t) > 0$ at its maximum. To show this, notice that from (21) we have

$$\varphi_0(1) = 1 - \frac{\beta}{2N} \sum_{i=0}^{N-1} 1 - \frac{\beta}{2H} \sum_{i=N+1}^{N+H} 1 = 1 - \beta > 0,$$

and, looking again at (23), $\varphi_0(1) > 0$ would not be possible if it were $\varphi_k(t) \le 0\ \forall t$ for some k. From a,b,c, it follows that each $\varphi_k(t)$, $k = 0, 1, \ldots, N-1$, has exactly two roots, $\underline{t}(k)$ and $\bar{t}(k)$, in $[0, +\infty)$. Moreover,

$$\varphi_k(t) < 0 \text{ for } t \in [0, \underline{t}(k)) \cup (\bar{t}(k), +\infty), \text{ while } \varphi_k(t) > 0 \text{ for } t \in (\underline{t}(k), \bar{t}(k)). \tag{24}$$

[10] The reason for introducing H is that the theorem will be proven in a slightly more general form where H is any integer ≥ 1 and not just $3N$. The choice $H = 3N$ is typical, and this is why Theorem 2 was stated with $H = 3N$. However, the extra generality allowed by other values of H can turn out to be useful to tighten the bounds $\underline{\epsilon}(\cdot)$ and $\bar{\epsilon}(\cdot)$ in some cases when N is not too large. This issue is not further discussed in this paper.

We next prove relation (11). For all positive integers $k = 0, 1, \dots$ define

$$F_k(v) = \mathbb{P}^k\{V(z_k^*) \le v \ \wedge \ \tilde{s}_k^* = k\}, \tag{25}$$

where $z_k^* = M_k(\delta_1, \dots, \delta_k)$, $\tilde{s}_k^*$ = number of support elements, and $\wedge$ is the "and" operator. In words, $F_k(v)$ is the probability that, with a sample of k instances of δ, all of them are of support and the decision has risk no more than v. The F_k's are generalized distribution functions, [44]. Functions $F_0, F_1, \dots$ are different from one problem to another and, as we shall show, for a given problem the left-hand side of (11) can be computed from $F_0, F_1, \dots$.

Start by noting that the events $\{\tilde{s}_N^* = k_1\}$ and $\{\tilde{s}_N^* = k_2\}$ are not overlapping for $k_1 \neq k_2$, so that

$$\mathbb{P}^N\{\underline{\epsilon}(\tilde{s}_N^*) \le V(z_N^*) \le \overline{\epsilon}(\tilde{s}_N^*)\} = \sum_{k=0}^{N} \mathbb{P}^N\{\underline{\epsilon}(k) \le V(z_N^*) \le \overline{\epsilon}(k) \ \wedge \ \tilde{s}_N^* = k\}. \tag{26}$$

Focus on one event $S_k := \{\underline{\epsilon}(k) \le V(z_N^*) \le \overline{\epsilon}(k) \ \wedge \ \tilde{s}_N^* = k\} \subseteq \Delta^N$ and, for each sample $\delta_1, \dots, \delta_N \in S_k$, evaluate the indexes of the δ_i's that correspond to the support elements. Group together all the samples with the same indexes. In this way, S_k is partitioned in $\binom{N}{k}$ subsets. All these subsets have the same probability because $\delta_1, \dots, \delta_N$ are independent and identically distributed. Hence,

$$\mathbb{P}^N\{\underline{\epsilon}(\tilde{s}_N^*) \le V(z_N^*) \le \overline{\epsilon}(\tilde{s}_N^*)\} = \binom{N}{k} \mathbb{P}^N\{A\}, \tag{27}$$

where A is one of these subsets, say the one where the indexes are $1, 2, \dots, k$, that is,

$$A := \{\underline{\epsilon}(k) \le V(z_N^*) \le \overline{\epsilon}(k) \ \wedge \ \tilde{s}_N^* = k \ \wedge \ \delta_1, \dots, \delta_k \text{ are of support}\}.$$

We show below that set A is equal to

$$B := \{\underline{\epsilon}(k) \le V(z_k^*) \le \overline{\epsilon}(k) \ \wedge \ \tilde{s}_k^* = k \ \wedge \ z_k^* \in \mathcal{Z}_{\delta_i}, \ i = k+1, \dots, N\}$$

up to a zero probability set.

We first show that $A \subseteq B$ up to a zero probability set. Since in A the support elements are the first k, by the non-degeneracy Assumption 4, $z_N^* = z_k^*$ up to a zero probability set. Thus, $\underline{\epsilon}(k) \le V(z_N^*) \le \overline{\epsilon}(k)$ implies $\underline{\epsilon}(k) \le V(z_k^*) \le \overline{\epsilon}(k)$ up to a zero probability set. Moreover, $z_k^* \in \mathcal{Z}_{\delta_i}$, $i = k+1, \dots, N$, because if $z_k^* \notin \mathcal{Z}_{\delta_{\bar{i}}}$ for some $\bar{i} \in \{k+1, \dots, N\}$, then $z_N^* \neq z_k^*$ by property (iii) in Assumption 3. We finally show that all the $\delta_1, \dots, \delta_k$ are of support for $M_k(\delta_1, \dots, \delta_k)$, which gives $\tilde{s}_k^* = k$. Indeed, if one among $\delta_1, \dots, \delta_k$, say δ_1, were not of support, then $M_{k-1}(\delta_2, \dots, \delta_k) = z_k^*$. But then, by adding $\delta_{k+1}, \dots, \delta_N$ – which correspond to $\mathcal{Z}_{\delta_i}$ such that $z_k^* \in \mathcal{Z}_{\delta_i}$ – one would obtain $M_{N-1}(\delta_2, \dots, \delta_N) = z_k^* = z_N^*$ by property (ii) in Assumption 3, and, hence, δ_1 would not be of support for $M_N(\delta_1, \dots, \delta_N)$, which is a contradiction.

Next we show that $B \subseteq A$ up to a zero probability set. Since in B it holds that $z_k^* \in \mathcal{Z}_{\delta_i}$, $i = k+1, \dots, N$, by property (ii) in Assumption 3 we obtain that $z_N^* = z_k^*$ and, thus, relation $\underline{\epsilon}(k) \le V(z_k^*) \le \overline{\epsilon}(k)$ implies that $\underline{\epsilon}(k) \le V(z_N^*) \le \overline{\epsilon}(k)$. We next show that $\tilde{s}_N^* = k \wedge \delta_1, \dots, \delta_k$ are of support, which is equivalent to say that $\delta_1, \dots, \delta_k$ are the only support scenarios for $M_N(\delta_1, \dots, \delta_N)$, up to a zero probability set. First, none of the $\delta_{k+1}, \dots, \delta_N$ can be of support for $M_N(\delta_1, \dots, \delta_N)$. Indeed, remove one of these scenarios, say δ_N, from the sample $\delta_1, \dots, \delta_N$. Since $\delta_1, \dots, \delta_{N-1}$

is $\delta_1, \ldots, \delta_k$ with the addition of δ_i, $i = k+1, \ldots, N$, for which $z_k^* \in \mathcal{Z}_{\delta_i}$, by property (ii) in Assumption 3 one obtains $M_{N-1}(\delta_1, \ldots, \delta_{N-1}) = z_k^* = z_N^*$. This shows that δ_N is not of support. To next show that $\delta_1, \ldots, \delta_k$ are of support for $M_N(\delta_1, \ldots, \delta_N)$ up to a zero probability set, proceed by contradiction, and assume that at least one of the first k scenarios is not of support with non-zero probability. Since $\delta_{k+1}, \ldots, \delta_N$ are not of support, then the support scenarios for $M_N(\delta_1, \ldots, \delta_N)$ would be with non-zero probability a strict subset of $\delta_1, \ldots, \delta_k$. However, since properties (ii)-(iii) in Assumption 3 imply that the decision obtained from a strict subset of $\delta_1, \ldots, \delta_k$ must be different from z_k^* (indeed, to be the same, by property (iii), the $\mathcal{Z}_{\delta_i}$ corresponding to the missing scenarios must include z_k^*; but, then, by property (ii), adding all the missing scenarios except one the solution would still be z_k^*, contradicting the assumption that all the $\delta_1, \ldots, \delta_k$ are of support for $M_k(\delta_1, \ldots, \delta_k)$), then we would have that the decision obtained from the support scenarios of $M_N(\delta_1, \ldots, \delta_N)$ would be different from z_k^*, and, hence, different from $z_N^* = M_N(\delta_1, \ldots, \delta_N)$, with non-zero probability. This, however, contradicts the non-degeneracy Assumption 4.

We next show that

$$\mathbb{P}^N\{B\} = \int_{[\underline{\epsilon}(k), \overline{\epsilon}(k)]} (1-v)^{N-k} \mathrm{d}F_k(v). \tag{28}$$

Indeed, owing to the independence of $\delta_1, \ldots, \delta_N$, $(1-v)^{N-k}$ is the conditional probability that $z_k^* \in \mathcal{Z}_{\delta_i}$, $i = k+1, \ldots, N$, given that $V(z_k^*) = v$ and $\tilde{s}_k^* = k$. Then, recalling the definition of F_k in (25), (28) follows from [44, Chapter II, Section 7, Equation (17)].

Since $\mathbb{P}^N\{A\} = \mathbb{P}^N\{B\}$ (which follows from the fact that $A = B$ up to a zero probability set), substituting (28) in (27) and further (27) in (26) yields

$$\mathbb{P}^N\{\underline{\epsilon}(\tilde{s}_N^*) \leq V(z_N^*) \leq \overline{\epsilon}(\tilde{s}_N^*)\} = \sum_{k=0}^{N} \binom{N}{k} \int_{[\underline{\epsilon}(k), \overline{\epsilon}(k)]} (1-v)^{N-k} \mathrm{d}F_k(v). \tag{29}$$

Equation (29) provides the fundamental formula by which $\mathbb{P}^N\{\underline{\epsilon}(\tilde{s}_N^*) \leq V(z_N^*) \leq \overline{\epsilon}(\tilde{s}_N^*)\}$ can be computed from $F_0, F_1, \ldots$. To proceed in the evaluation of the right-hand side of (29), we have now to characterize the distributions $F_0, F_1, \ldots$, which is done in the following.

If the same argument used to derive (29) is repeated with z_m^*, $m = 0, 1, \ldots$, in place of z_N^* and with 0 in place of $\underline{\epsilon}(\tilde{s}_N^*)$ and and 1 in place of $\overline{\epsilon}(\tilde{s}_N^*)$, relation

$$\mathbb{P}^N\{0 \leq V(z_m^*) \leq 1\} = \sum_{k=0}^{m} \binom{m}{k} \int_{[0,1]} (1-v)^{N-k} \mathrm{d}F_k(v)$$

is found. Since $V(z_m^*)$ takes value in $[0, 1]$, the right-hand side of this equality is clearly equal to 1, which shows that $F_0, F_1, \ldots$ must satisfy the following generalized moment conditions:

$$\sum_{k=0}^{m} \binom{m}{k} \int_{[0,1]} (1-v)^{m-k} \mathrm{d}F_k(v) = 1, \quad m = 0, 1, \ldots. \tag{30}$$

Equation (30) provides a characterization of $F_0, F_1, \ldots$ under which the right-hand side of (29) can be evaluated, and program (32) below returns a lower bound to $\mathbb{P}^N\{\underline{\epsilon}(\tilde{s}_N^*) \leq V(z_N^*) \leq \overline{\epsilon}(\tilde{s}_N^*)\}$. That is,

$$\mathbb{P}^N\{\underline{\epsilon}(\tilde{s}_N^*) \leq V(z_N^*) \leq \overline{\epsilon}(\tilde{s}_N^*)\} \geq \gamma, \tag{31}$$

with ($\mathcal{C}$ denotes the positive cone of generalized distribution functions)

$$\begin{aligned}\gamma = \inf_{F_k,\, k=0,1,\ldots} & \sum_{k=0}^{N} \binom{N}{k} \int_{[\underline{\epsilon}(k),\overline{\epsilon}(k)]} (1-v)^{N-k} \mathrm{d}F_k(v) \\ \text{subject to: } & \sum_{k=0}^{m} \binom{m}{k} \int_{[0,1]} (1-v)^{m-k} \mathrm{d}F_k(v) = 1, \quad m = 0,1,\ldots \\ & F_k \in \mathcal{C}, k = 0,1,\ldots.\end{aligned} \tag{32}$$

The last part of the proof consists in showing that $\gamma \geq 1-\beta$, from which (11) follows.

Consider the following truncated version of (32):

$$\begin{aligned}\gamma_H = \inf_{F_0,F_1,\ldots,F_{N+H}} & \sum_{k=0}^{N} \binom{N}{k} \int_{[\underline{\epsilon}(k),\overline{\epsilon}(k)]} (1-v)^{N-k} \mathrm{d}F_k(v) \\ \text{subject to: } & \sum_{k=0}^{m} \binom{m}{k} \int_{[0,1]} (1-v)^{m-k} \mathrm{d}F_k(v) = 1, \quad m = 0,1,\ldots,N+H, \\ & F_0, F_1, \ldots, F_{N+H} \in \mathcal{C}.\end{aligned} \tag{33}$$

Since in (32) and (33) the cost function only depends on $F_0, F_1, \ldots, F_N$ and (33) is less constrained than (32), we have that

$$\gamma \geq \gamma_H. \tag{34}$$

The dual of (33) is

$$\begin{aligned}\gamma_H^* = \sup_{\lambda_0,\lambda_1,\ldots,\lambda_{N+H}} & \sum_{m=0}^{N+H} \lambda_m \\ \text{subject to: } & \sum_{m=k}^{N+H} \lambda_m \binom{m}{k} (1-v)^{m-k} \\ & \leq \begin{cases} \binom{N}{k}(1-v)^{N-k} \cdot \mathbf{1}_{[\underline{\epsilon}(k),\overline{\epsilon}(k)]}(v), & k = 0,1,\ldots,N \\ 0, & k = N+1,\ldots,N+H \end{cases}, \\ & v \in [0,1],\end{aligned} \tag{35}$$

where $\mathbf{1}$ denotes the indicator function. The following derivation, provided for self-containedness, shows that $\gamma_H \geq \gamma_H^*$ (weak duality): for every $F_0, F_1, \ldots, F_{N+H}$ feasible for (33) and $\lambda_0, \lambda_1, \ldots, \lambda_{N+H}$ feasible for (35) it holds that

$$\begin{aligned}& \sum_{k=0}^{N} \binom{N}{k} \int_{[\underline{\epsilon}(k),\overline{\epsilon}(k)]} (1-v)^{N-k} \mathrm{d}F_k(v) \\ &= \sum_{k=0}^{N} \int_{[0,1]} \binom{N}{k} (1-v)^{N-k} \cdot \mathbf{1}_{[\underline{\epsilon}(k),\overline{\epsilon}(k)]}(v) \mathrm{d}F_k(v) \\ &\geq \sum_{k=0}^{N+H} \int_{[0,1]} \sum_{m=k}^{N+H} \lambda_m \binom{m}{k} (1-v)^{m-k} \mathrm{d}F_k(v) \\ &= \sum_{m=0}^{N+H} \lambda_m \sum_{k=0}^{m} \binom{m}{k} \int_{[0,1]} (1-v)^{m-k} \mathrm{d}F_k(v) \\ &= \sum_{m=0}^{N+H} \lambda_m,\end{aligned}$$

so that taking the inf on the left-hand side and the sup on the right-hand side yields

$$\gamma_H \geq \gamma_H^*. \tag{36}$$

Inequalities (31), (34), and (36) give

$$\mathbb{P}^N\left\{\underline{\epsilon}(\tilde{s}_N^*) \leq V(z_N^*) \leq \overline{\epsilon}(\tilde{s}_N^*)\right\} \geq \gamma \geq \gamma_H \geq \gamma_H^*, \tag{37}$$

from which the theorem can be proven by showing that $\gamma_H^* \geq 1-\beta$.

To show that $\gamma_H^* \geq 1-\beta$, perform the substitution $t := 1-v$ and rewrite (35) as

$$\begin{aligned} \gamma_H^* = \sup_{\lambda_0,\lambda_1,\ldots,\lambda_{N+H}} & \sum_{m=0}^{N+H} \lambda_m \\ \text{subject to: } & \sum_{m=k}^{N+H} \lambda_m \binom{m}{k} t^{m-k} \\ & \leq \begin{cases} \binom{N}{k} t^{N-k} \cdot \mathbf{1}_{[1-\overline{\epsilon}(k),1-\underline{\epsilon}(k)]}(t), & k=0,1,\ldots,N \\ 0, & k=N+1,\ldots,N+H \end{cases}, \\ & t \in [0,1]. \end{aligned} \tag{38}$$

Consider now

$$\begin{array}{l} \lambda_m = -\frac{\beta}{2N}, \quad m=0,1,\ldots,N-1, \\ \lambda_m = 1, \quad m=N, \\ \lambda_m = -\frac{\beta}{2H}, \quad m=N+1,N+2,\ldots,N+H. \end{array} \tag{39}$$

We want to show that the selection of λ_m's in (39) is feasible for (38), which gives the sought inequality because for these λ_m's we have that $\sum_{m=0}^{N+H}\lambda_m = 1-\beta$, and $\gamma_H^* \geq 1-\beta$ because γ_H^* is the sup over all the feasible choices of the λ_m's.

Consider first the constraints in (38) for $k > N$. These are trivially satisfied since for the λ_m's in (39) the left-hand side of the inequality is negative for $t \in [0,1]$. When instead $k \leq N$, for the λ_m's in (39), the left-hand side of the inequality in (38) coincides with the polynomial $\varphi_k(t)$ in (21). Since $\varphi_k(t) \leq \binom{N}{k} t^{N-k}$, $k=0,1,\ldots,N$, and since $\varphi_k(t) \leq 0$ for t outside the interval $[1-\overline{\epsilon}(k), 1-\underline{\epsilon}(k)]$ (see (24), (22), and the definitions of $\overline{\epsilon}(k)$ and $\underline{\epsilon}(k)$ in Theorem 2), also the constraints in (38) for $k \leq N$ are satisfied.

Wrapping up, the selection of λ_m's in (39) is feasible for (38) and, hence, $\gamma_H^* \geq 1-\beta$, which used in (37) gives

$$\mathbb{P}^N\left\{\underline{\epsilon}(\tilde{s}_N^*) \leq V(z_N^*) \leq \overline{\epsilon}(\tilde{s}_N^*)\right\} \geq 1-\beta.$$

This concludes the proof. □

5.2 Proof of Theorem 1

The optimization programs (3) complemented with a convex tie-break rule as specified in Assumption 1 define a family of maps M_m from the sample $\delta_1,\ldots,\delta_m$ to the solution x_m^*. We show that this family of maps satisfies the assumptions of Theorem 2 so that Theorem 1 follows from Theorem 2 with the positions $z_m^* = x_m^*$, $\tilde{s}_m^* = s_m^*$ and $\mathcal{Z}_\delta = \mathcal{X}_\delta$, and by noting that equation (10) in Theorem 2 can be dropped in the context of Theorem 1 since $N > d$ implies $\tilde{s}_N^* < N$.

Consider Assumption 3. First of all note that the maps M_m are permutation invariant (point (i) in Assumption 3) because the solution to (3) clearly does not depend on the order in which constraints are sampled. Consider now (ii) in Assumption 3. In (3) the solution is selected as the feasible point that achieves the smallest cost $c^T x$ and, if a tie occurs, the tie is broken by minimizing the convex functions $t_1(x), t_2(x), \ldots$. By adding extra constraints that are satisfied at x_m^*, the feasibility domain shrinks while x_m^* remains feasible. Hence, x_m^* remains the optimal solution and (ii) follows. Referring to (iii), if some δ_i's are added such that at least one δ_i corresponds to a constraint that is not satisfied by x_m^*, then the solution x_m^* has to change and move to a feasible point, and this gives (iii). Finally, notice that the non-degeneracy Assumption 4 is guaranteed by Assumption 2. □

References

1. Alamo, T., Tempo, R., Camacho, E.: A randomized strategy for probabilistic solutions of uncertain feasibility and optimization problems. IEEE Transactions on Automatic Control **54**(11), 2545–2559 (2009)
2. Bayraksan, G., Morton, D.: Assessing solution quality in stochastic programs. Mathematical Programming **108**, 495–514 (2006)
3. Bayraksan, G., Morton, D.: Assessing solution quality in stochastic programs via sampling. In: M. Oskoorouchi (ed.) Tutorials in operations research, pp. 102–122. Informs (2009)
4. Ben-Tal, A., Nemirovski, A.: On safe tractable approximations of chance-constrained linear matrix inequalities. Mathematics of Operations Research **34**(1), 1–25 (2009)
5. Bertsimas, D., Thiele, A.: Robust and data-driven optimization: modern decision-making under uncertainty. In: Tutorials on Operations Research. INFORMS (2006)
6. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
7. Calafiore, G., Campi, M.: Uncertain convex programs: randomized solutions and confidence levels. Mathematical Programming **102**(1), 25–46 (2005). DOI http://dx.doi.org/10.1007/s10107-003-0499-y
8. Calafiore, G., Campi, M.: The scenario approach to robust control design. IEEE Transactions on Automatic Control **51**(5), 742–753 (2006)
9. Campi, M.: Classification with guaranteed probability of error. Machine Learning **80**, 63–84 (2010)
10. Campi, M., Calafiore, G., Garatti, S.: Interval predictor models: identification and reliability. Automatica **45**(2), 382–392 (2009). DOI http://dx.doi.org/10.1016/j.automatica.2008.09.004
11. Campi, M., Carè, A.: Random convex programs with l_1-regularization: sparsity and generalization. SIAM Journal on Control and Optimization **51**(5), 3532–3557 (2013)
12. Campi, M., Garatti, S.: The exact feasibility of randomized solutions of uncertain convex programs. SIAM Journal on Optimization **19**(3), 1211–1230 (2008). DOI http://dx.doi.org/10.1137/07069821X
13. Campi, M., Garatti, S.: A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. Journal of Optimization Theory and Applications **148**(2), 257–280 (2011)
14. Campi, M., Garatti, S.: Wait-and-judge scenario optimization. Mathematical Programming (2016). Doi: 10.1007/s10107-016-1056-9
15. Campi, M., Garatti, S., Prandini, M.: The scenario approach for systems and control design. Annual Reviews in Control **33**(2), 149 – 157 (2009). DOI http://dx.doi.org/10.1016/j.arcontrol.2009.07.001
16. Carè, A., Garatti, S., Campi, M.: FAST - Fast Algorithm for the Scenario Technique. Operations Research **62**(3), 662–671 (2014)
17. Carè, A., Garatti, S., Campi, M.: Scenario min-max optimization and the risk of empirical costs. SIAM Journal on Optimization **25**(4), 2061–2080 (2015)
18. Crespo, L., Giesy, D., Kenny, S.: Interval predictor models with a formal characterization of uncertainty and reliability. In: Proceedings of the 53rd IEEE Conference on Decision and Control (CDC), pp. 5991–5996. Los Angeles, CA, USA (2014)
19. Crespo, L., Kenny, S., Giesy, D.: Random predictor models for rigorous uncertainty quantification. International Journal for Uncertainty Quantification **5**(5), 469–489 (2015)
20. Crespo, L., Kenny, S., Giesy, D., Norman, R., Blattnig, S.: Application of interval predictor models to space radiation shielding. In: Proceedings of the 18th AIAA Non-Deterministic Approaches Conference. San Diego, CA, USA (2016)
21. Delage, E., Ye, Y.: Distributionally robust optimization under moment uncertainty with application to data-driven problems. Operations Research **58**(3), 596–612 (2010)

22. Dentcheva, D.: Optimization models with probabilistic constraints. In: G. Calafiore, F. Dabbene (eds.) Probabilistic and Randomized Methods for Design under Uncertainty. Springer-Verlag, London (2006)
23. Esfahani, P., Sutter, T., Lygeros, J.: Performance bounds for the scenario approach and an extension to a class of non-convex programs. IEEE Transactions on Automatic Control **60**(1), 46–58 (2015)
24. Fabozzi, F., Kolm, P., Pachamanova, D., Focardi, S.: Robust portfolio optimization and management. Wiley (2010)
25. Garatti, S., Campi, M.: Modulating robustness in control design: principles and algorithms. IEEE Control Systems Magazine **33**(2), 36–51 (2013). DOI http://dx.doi.org/10.1109/MCS.2012.2234964
26. Grammatico, S., Zhang, X., Margellos, K., Goulart, P., Lygeros, J.: A scenario approach for non-convex control design. IEEE Transactions on Automatic Control **61**(2), 334–345 (2016)
27. Hong, L., Hu, Z., Liu, G.: Monte Carlo methods for value-at-risk and conditional value-at-risk: a review. ACM Transactions on Modeling and Computer Simulation **24**(4), 22:1–22:37 (2014)
28. Linderoth, J., Shapiro, A., Wright, S.: The empirical behavior of sampling methods for stochastic programming. Annals of Operations Research **142**, 215–241 (2006)
29. Luedtke, J., Ahmed, S.: A sample approximation approach for optimization with probabilistic constraints. SIAM Journal on Optimization **19**, 674–699 (2008). DOI http://dx.doi.org/10.1137/070702928
30. Luedtke, J., Ahmed, S., Nemhauser, G.: An integer programming approach for linear programs with probabilistic constraints. Mathematical Programming **122**(2), 247–272 (2010)
31. Mak, W., Morton, D., Wood, R.: Monte Carlo bounding techniques for determing solution quality in stochastic programs. Operations Research Letters **24**, 47–56 (1999)
32. Margellos, K., Prandini, M., Lygeros, J.: 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)
33. de Mello, T.H.: Variable-sample methods for stochastic optimization. ACM Transactions on Modeling and Computer Simulation **13**, 108–133 (2003)
34. de Mello, T.H., Bayraksan, G.: Monte Carlo sampling-based methods for stochastic optimization. Surveys in Operations Research and Management Science **19**(1), 56–85 (2014)
35. Nemirovski, A.: On safe tractable approximations of chance constraints. European Journal of Operational Research **219**, 707–718 (2012)
36. Nemirovski, A., Shapiro, A.: Convex approximations of chance constrained programs. SIAM Journal on Optimization **17**(4), 969–996 (2006). DOI http://dx.doi.org/10.1137/050622328
37. Nemirovski, A., Shapiro, A.: Scenario approximations of chance constraints. In: G. Calafiore, F. Dabbene (eds.) Probabilistic and Randomized Methods for Design under Uncertainty. Springer-Verlag, London (2006)
38. Pagnoncelli, B., Ahmed, S., Shapiro, A.: Sample average approximation method for chance constrained programming: theory and applications. Journal of Optimization Theory and Applications **142**(2), 399–416 (2009)
39. Pagnoncelli, B., Reich, D., Campi, M.: Risk-return trade-off with the scenario approach in practice: A case study in portfolio selection. Journal of Optimization Theory and Applications **155**(2), 707–722 (2012)
40. Pagnoncelli, B., Vanduffel, S.: A provisioning problem with stochastic payments. European Journal of Operational Research **221**(2), 445–453 (2012)
41. Schildbach, G., Fagiano, L., Frei, C., Morari, M.: The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. Automatica **50**(12), 3009–3018 (2014)
42. Schildbach, G., Fagiano, L., Morari, M.: Randomized solutions to convex programs with multiple chance constraints. SIAM Journal on Optimization **23**(4), 2479–2501 (2013)
43. Shapiro, A., Dentcheva, D., Ruszczyński, A.: Lectures on stochastic programming: modeling and theory. MPS-SIAM, Philadelphia, USA (2009)
44. Shiryaev, A.: Probability. Springer, New York (1996)
45. Thiele, A.: Robust stochastic programming with uncertain probabilities. IMA Journal of Management Mathematics **19**(3), 289–321 (2008). DOI http://dx.doi.org/10.1093/imaman/dpm011
46. Vayanos, P., Kuhn, D., Rustem, B.: A constraint sampling approach for multistage robust optimization. Automatica **48**(3), 459–471 (2012)
47. Welsh, J., Kong, H.: Robust experiment design through randomisation with chance constraints. In: Proceedings of the 18th IFAC World Congress. Milan, Italy (2011)
48. Welsh, J., Rojas, C.: A scenario based approach to robust experiment design. In: Proceedings of the 15th IFAC Symposium on System Identification. Saint-Malo, France (2009)
49. Zhang, X., Grammatico, S., Schildbach, G., Goulart, P., Lygeros, J.: On the sample size of random convex programs with structured dependence on the uncertainty. Automatica **60**, 182–188 (2015)
50. Zhou, Z., Cogill, R.: Reliable approximations of probability-constrained stochastic linear-quadratic control. Automatica **49**(8), 2435–2439 (2013)
51. Zymler, S., Kuhn, D., Rustem, B.: Distributionally robust joint chance constraints with second-order moment information. Mathematical Programming **137**(1-2), 167–198 (2013)

A MATLAB code

The following MATLAB code returns $\underline{\epsilon}(k)$ and $\overline{\epsilon}(k)$ for user assigned k, N, and β.

```
function [epsL, epsU] = epsLU(k,N,bet)

    alphaL = betaincinv(bet,k,N-k+1);
    alphaU = 1-betaincinv(bet,N-k+1,k);
    m1 = [k:1:N];
    aux1 = sum(triu(log(ones(N-k+1,1)*m1),1),2);
    aux2 = sum(triu(log(ones(N-k+1,1)*(m1-k)),1),2);
    coeffs1 = aux2-aux1;
    m2 = [N+1:1:4*N];
    aux3 = sum(tril(log(ones(3*N,1)*m2)),2);
    aux4 = sum(tril(log(ones(3*N,1)*(m2-k))),2);
    coeffs2 = aux3-aux4;

    t1 = 1-alphaL;
    t2 = 1;
    poly1 = 1+bet/(2*N)-bet/(2*N)*sum(exp(coeffs1 - (N-m1')*log(t1))) ...
        -bet/(6*N)*sum(exp(coeffs2 + (m2'-N)*log(t1)));
    poly2 = 1+bet/(2*N)-bet/(2*N)*sum(exp(coeffs1 - (N-m1')*log(t2))) ...
        -bet/(6*N)*sum(exp(coeffs2 + (m2'-N)*log(t2)));
    if ((poly1*poly2) > 0)
        epsL = 0;
    else
        while t2-t1 > 1e-10
            t = (t1+t2)/2;
            polyt = 1+bet/(2*N)-bet/(2*N)*sum(exp(coeffs1 - (N-m1')*log(t))) ...
                -bet/(6*N)*sum(exp(coeffs2 + (m2'-N)*log(t)));
            if polyt > 0
                t1=t;
            else
                t2=t;
            end
        end
        epsL = 1-t2;
    end

    t1 = 0;
    t2 = 1-alphaU;
    poly1 = 1+bet/(2*N)-bet/(2*N)*sum(exp(coeffs1 - (N-m1')*log(t1))) ...
        -bet/(6*N)*sum(exp(coeffs2 + (m2'-N)*log(t1)));
    poly2 = 1+bet/(2*N)-bet/(2*N)*sum(exp(coeffs1 - (N-m1')*log(t2))) ...
        -bet/(6*N)*sum(exp(coeffs2 + (m2'-N)*log(t2)));
    if ((poly1*poly2) > 0)
        epsL = 0;
    else
        while t2-t1 > 1e-10
            t = (t1+t2)/2;
            polyt = 1+bet/(2*N)-bet/(2*N)*sum(exp(coeffs1 - (N-m1')*log(t))) ...
                -bet/(6*N)*sum(exp(coeffs2 + (m2'-N)*log(t)));
            if polyt > 0
                t2=t;
            else
                t1=t;
            end
        end
        epsU = 1-t1;
    end
end
```