Chance-constrained optimization via randomization: feasibility and optimality*

M.C. Campi[†] S. Garatti[‡]

Abstract

In this paper we study the link between a semi-infinite chance-constrained optimization problem and its randomized version, i.e. the problem obtained by sampling a finite number of its constraints.

Extending previous results on the feasibility of randomized convex programs, we establish here the feasibility of the solution obtained after the elimination of a portion of the sampled constraints. Constraints removal allows one to improve the cost function at the price of a decreased feasibility. The cost improvement can be inspected directly from the optimization result, while the theory here developed permits to keep control on the other side of the coin, the feasibility of the obtained solution. In this way, trading feasibility for performance through a *sampling*-and-*discarding* approach is put on solid mathematical grounds by the results of this paper.

The feasibility result obtained in this paper applies to all chance-constrained optimization problems with convex constraints, and has the distinctive feature that it holds true irrespective of the algorithm used for constraints removal. One can thus e.g. use a greedy algorithm – which is computationally low-demanding – and the corresponding feasibility remains guaranteed. We further prove in this paper that if constraints removal is optimally done – i.e. one deletes those constraints leading to the largest possible cost improvement – a precise optimality link to the original semi-infinite chance-constrained problem in addition holds.

Keywords: Chance-constrained optimization, Convex optimization, Randomized methods.

*This paper was supported by the MIUR project "Identification and adaptive control of industrial systems".

[†]Dipartimento di Elettronica per l'Automazione - Università di Brescia, via Branze 38, 25123 Brescia, Italia. E-mail: marco.campi@ing.unibs.it, web-site: http://bsing.ing.unibs.it/~campi/

[‡]Dipartimento di Elettronica e Informatica - Politecnico di Milano, P.zza Leonardo da Vinci 32, 20133 Milano, Italia. E-mail: sgaratti@elet.polimi.it, web-site: http://home.dei.polimi.it/sgaratti/

1 Introduction

Letting $\mathcal{X} \subseteq \mathbb{R}^d$ be a convex and closed domain of optimization, consider a family of constraints $x \in \mathcal{X}_\delta$ parameterized in $\delta \in \Delta$, where the sets $\mathcal{X}_\delta$ are convex and closed. Convexity of the constraints is an assumption in effect throughout this paper. δ is the *uncertain* parameter and it describes different instances of an uncertain optimization scenario. Adopting a probabilistic description of uncertainty, let us suppose that the support Δ for δ is endowed with a σ-algebra $\mathcal{D}$ and that a probability $\mathbb{P}$ is defined over $\mathcal{D}$. $\mathbb{P}$ describes the probability with which the scenario parameter δ takes value in Δ. Then, a chance-constrained program, [8, 16, 20, 21, 23, 11], is written as:

$$\begin{aligned} \mathrm{CCP}_\epsilon: \quad & \min_{x \in \mathcal{X}} c^T x \\ & \text{subject to: } x \in \mathcal{X}_\delta \text{ with probability } \mathbb{P} \geq 1 - \epsilon. \end{aligned} \tag{1}$$

Here, the set of δ's such that $x \in \mathcal{X}_\delta$ is assumed to be an element of $\mathcal{D}$, that is it is a measurable set, so that its probability is well defined. Also, linearity of the objective function is without loss of generality since any objective of the kind $\min_{x \in \mathcal{X}} c(x)$, where $c(x) : \mathcal{X} \to \mathbb{R}$ is a convex function, can be re-written as $\min_{x \in \mathcal{X}, y \geq c(x)} y$, where y is a scalar slack variable. Problem 1 is general enough to encompass chance-constrained LP (linear programs), QP (quadratic programs), SOCP (second-order cone programs), and SDP (semi-definite programs). In general, (1) involves an infinite set of constraints, i.e. Δ has infinite cardinality.

In CCP_ϵ, constraint violation is tolerated, but the violated constraint set must be no larger than ϵ. This parameter ϵ allows to trade robustness for performance: the optimal objective value J^*_ϵ of CCP_ϵ is a decreasing function of ϵ and provides a quantification of such a trade-off. Depending on the application at hand, which can cover a wide range from control to prediction and from engineering design to financial economics, ϵ can take different values and has not necessarily to be thought of as a "small" parameter. The fact that J^*_ϵ is higher for small violation probabilities ϵ is sometimes phrased as the "price of robustness", [4].

An x point satisfying constraints $x \in \mathcal{X}_\delta$ with probability $\mathbb{P} \geq 1 - \epsilon$ is a *feasible* point for the chance-constrained problem and the solution of CCP_ϵ is a feasible point also minimizing the optimization objective. The feasible set of CCP_ϵ is in general non-convex in spite of the convexity of the sets $\mathcal{X}_\delta$, see [21, 22], with a few exceptions, [21, 13, 11]. Consequently, an exact numerical solution of CCP_ϵ is in general very hard to find.

1.1 Objectives of this paper

In this paper we consider randomized approximations of chance-constrained optimization problems.

By suitably sampling of the constraints in Δ (*constraints randomization*), a program with a *finite* number of constraints is obtained (the "scenario" program), and we further allow for removal of constraints from this finite set, so improving the objective according to a chance-constrained philosophy. The removal of constraints can be performed by any rule, possibly a greedy one that can be implemented at low computational effort. The feasibility Theorem 1 establishes the following fact:

> if N constraints are sampled and k of them are eliminated according to any arbitrary rule, the solution that satisfies the remaining $N-k$ constraints is, with high confidence, feasible for the chance-constrained program CCP_ϵ in (1), provided that N and k satisfy a certain condition (3).

This theorem justifies at a very deep theoretical level the use of randomization in chance-constrained and opens up practical routes to address chance-constrained optimization: after sampling, constraints are removed according to some, possibly greedy, procedure and, at the end of the elimination process, the actually incurred optimization cost is inspected for satisfaction, while the theory here developed allows one to keep control on the feasibility of the obtained solution.

A strength of this theory is that it applies to *all* chance-constrained problems with convex constraints, including as special cases linear and quadratic constraints, and constraints expressed in terms of LMI's (Linear Matrix Inequalities) corresponding to the wide class of semi-definite programming problems. Moreover, condition (3) is very tight, in the sense that it returns values for N and k close to the best possible values guaranteeing feasibility, a fact also shown in this paper.

Depending on the elimination rule, the incurred objective value can be close or less close to the optimal objective value J_ϵ^* of the CCP_ϵ program. Further working on this aspect, we prove in Theorem 2 that an optimal elimination of k constraints leads to an objective value that relates to J_ϵ^* in a precise way. Although this latter result is to date of limited practical use owing to the computational complexity of the ensuing combinatorial problem, it sheds light on certain interesting theoretical links.

1.2 Related literature

Randomized techniques are rapidly spreading in the context of stochastic optimization, as witnessed by recent contributions such as [25, 19, 10, 12, 18, 24].
In this paper, we are specifically interested in chance-constrained optimization problems as in equation (1). One approach to attack these problems is that of analytical methods, [2, 3, 4, 17, 22], where chance-constrained programs are reformulated (possibly with a certain degree of approximation) as convex problems whose solution can be found at low computational effort through standard algorithms. This approach offers an interesting and useful solution route, provided that the optimization problem has a certain specific structure. Along the randomized (also known as Sample Average Approximation – SAA) approach where one concentrates on a finite number N of constraints chosen at random, Chapter 4 in [24] provides a thorough presentation of the conditions for the randomized approximation to asymptotically reconstruct the original chance-constrained problem. The present paper, instead, deals with a finite-sample analysis, in that we want to determine the sample size N guaranteeing a given level of approximation. Finite-sample properties are also the topic of [14], which presents a very interesting analysis applicable in a set-up complementary to that of the present paper. Specifically, feasibility results are established for possibly non-convex

constraints, provided that the optimization domain has finite cardinality or in situations that can be reduced to this finite set-up, an application domain that covers many situations of interest.

The approach of the present paper builds on the so-called "scenario" approach of [5, 6, 7]. The crucial progress over [5, 6, 7] here made is that the results in [5, 6, 7] are extended to the case when constraints are a-posteriori removed, a situation of great practical importance any time one wants to trade feasibility for performance.

1.3 Structure of the paper

In the next Section 2, we first more formally introduce the scenario approach with constraints elimination to prepare the terrain for the feasibility Theorem 1, given at the end of the same section. To avoid breaking the flow of discourse, the proof of Theorem 1 is postponed to Section 4 while Section 3 provides complimentary theoretical material. Finally, optimality results are given in Section 5.

2 Randomized chance-constrained optimization: feasibility results

Suppose that N independent and identically distributed scenarios $\delta^{(1)}, \delta^{(2)}, \dots, \delta^{(N)}$ are extracted from Δ according to probability $\mathbb{P}$ (sampling or randomization of Δ). The idea behind the scenario approach of [5, 6, 7] is to substitute the vast multitude of constraints in the infinite initial domain Δ with these N extracted constraints only, and to find the optimal solution that satisfies *all* of these N constraints. This idea has an intuitive appeal in that the resulting optimization problem has a finite number of constraints only, so that it can be solved with standard numerical methods.

Along the above line of proceeding, one embraces a seemingly naive attitude of disregarding most of the constraints, those that have not been sampled, as though the corresponding scenarios could not occur. The strength of this randomized scenario approach stems from observing that the vast majority of these neglected constraints take automatically care of themselves, that is they are automatically satisfied even though they were not considered in the optimization problem. This observation has been recently posed on solid mathematical grounds in the papers [5, 6], and a tight quantification of the feasibility of randomized solutions has been obtained in [7].

In this paper, we move from the observation that finding a solution satisfying *all* of the N extracted constraints can be conservative and that this approach does not allow for any trading of constraint violation for performance; consequently, the achieved optimization value can be too large for the intended purposes and, therefore, unsatisfactory[1]. On the other hand, one can conceive proceeding differently and – following a chance-constrained philosophy – allow for violating part of the sampled constraints to improve the optimization value. This approach may have tremendous potentials in a number of application endeavors where one wants to meet a suitable compromise between performance and constraint satisfaction, and it is investigated in this paper.

A first viable option for constraints removal consists in deleting constraints one by one according to a greedy algorithm. In this scheme, one selects in succession those constraints which lead each time to the largest immediate improvement in the objective function until, say, k constraints are violated. This approach has the great advantage of being implementable at a low computational effort as one has to inspect at each removal a few constraint candidates only, those active at the currently reached solution. At the end of the greedy procedure, one can inspect the incurred objective value for satisfaction, while the theory developed in this paper – and substantiated in Theorem 1 below – provides theoretical guarantees that the solution violates less than a ϵ portion of the constraints in Δ, that is it is feasible for the chance-constrained problem (1). This result extends previous achievements in [5, 6, 7] along a direction of practical importance.

[1]The reader may be interested in going through Appendix A, part A.1, for a simple example illustrating this behavior.

Moving now toward a precise statement of Theorem 1, we first introduce some preliminary definitions and assumptions.

Assumption 1 *Every optimization problem subject to only a finite subset F of constraints from Δ, i.e.*

$$\min_{x \in \mathcal{X}} c^T x \quad \text{subject to: } x \in \mathcal{X}_\delta, \quad \delta \in F \subseteq \Delta, \tag{2}$$

is feasible, and its feasibility domain has a nonempty interior. Moreover, the solution of (2) *exists and is unique.* □

Though this Assumption 1 can be released, it is here made to keep notations and derivations easier to follow.

Theorem 1 holds true for any constraints removal procedure, and not only for the greedy algorithm that we have described above. A general removal procedure is next formalized in the following definition.

Definition 1 *Let $k < N$. An algorithm $\mathcal{A}$ for constraints removal is any rule by which k constraints out of a set of N constraints are selected and removed. The output of $\mathcal{A}$ is the set $\mathcal{A}\{\delta^{(1)}, \ldots, \delta^{(N)}\} = \{i_1, \ldots, i_k\}$ of the indexes of the k removed constraints.* □

The fact that $\mathcal{A}$ can be any removal algorithm provides us with an opportunity to pick the most suitable algorithm for the situation at hand, selecting from a range that goes from a handy greedy algorithm to the optimal algorithm where k constraints are eliminated to best improve the cost objective. To illustrate the type of possible algorithm selection, one choice is a recursive optimal elimination of groups of p, with $p \ll k$, constraints at a time (when $p = 1$ the greedy algorithm is recovered). Yet another choice consists in progressively updating the solution by eliminating all the active constraints at the currently reached solution. This option can be implemented at very low computational cost.

The randomized program where k constraints are removed as indicated by $\mathcal{A}$ is expressed as

$$\begin{aligned} \mathrm{RP}_{N,k}^{\mathcal{A}}: \quad & \min_{x \in \mathcal{X}} c^T x \\ & \text{subject to: } x \in \mathcal{X}_{\delta^{(i)}}, \; i \in \{1, \ldots, N\} - \mathcal{A}\{\delta^{(1)}, \ldots, \delta^{(N)}\}, \end{aligned}$$

and its solution will be hereafter indicated as x^*. We introduce the following assumption.

Assumption 2 *x^* almost surely violates all the k removed constraints.* $\square$

This is a natural assumption that reflects the fact that constraints removal is for the purpose of improving the optimization cost. For this violation assumption to hold, it is e.g. enough that, when at the end of the procedure some of the removed constraints remain satisfied, these constraints are reinstated and new constraints are eliminated until k constraints are violated.

Finally, introduce the following definition.

Definition 2 (violation probability) *The* violation probability *of a given $x \in \mathcal{X}$ is defined as $V(x) = \mathbb{P}\{\delta \in \Delta : x \notin \mathcal{X}_\delta\}$.* $\square$

The next Theorem 1 studies the feasibility of x^*. Note that x^* is a random variable because it depends on the random element $(\delta^{(1)}, \ldots, \delta^{(N)})$. Thus, its violation probability $V(x^*)$ is a random variable itself and it can be less than ϵ (i.e. x^* is feasible for the chance-constrained problem (1)) for some extractions of N constraints and not for others. The theorem establishes the condition under which $V(x^*) > \epsilon$ has any arbitrarily small probability β.

Theorem 1 *Let $\beta \in (0,1)$ be any small confidence parameter value. If N and k are such that*

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

(recall that d is the number of optimization variables), then $V(x^*) \le \epsilon$ holds with probability at least $1 - \beta$. $\square$

k	0	10	20	30	40	50	60	70	80	90
ϵ	0.017	0.031	0.041	0.051	0.059	0.068	0.075	0.083	0.090	0.097

Table 1: ϵ vs. k for $N = 2000$, $\beta = 10^{-10}$ and $d = 5$

In this theorem, and elsewhere, the measurability of set $\{V(x^*) \leq \epsilon\}$, as well as that of other sets in $\Delta^N = \Delta \times \cdots \times \Delta$, is taken as an assumption. The theorem holds true for any optimization problem with convex constraints, any constraints removal algorithm $\mathcal{A}$, and any probability measure $\mathbb{P}$. To avoid breaking the flow of presentation, the proof is given in the next Section 4.1.

Theorem 1 is a *feasibility* theorem and it says that the solution x^* obtained by inspecting N constraints only is a feasible solution for CCP_ϵ with high probability $1 - \beta$, provided that N and k fulfill condition (3). Here, β quantifies the probability $\mathbb{P}^N = \mathbb{P} \times \cdots \times \mathbb{P}$ in Δ^N of observing a "bad" multi-extraction $(\delta^{(1)}, \ldots, \delta^{(N)})$ such that $V(x^*) > \epsilon$. N and k bear a weak (logarithmic) dependence on $1/\beta$ so that parameter β can be pushed down to values so small (e.g. 10^{-10} or even 10^{-20}) that this confidence parameter looses any practical importance without a significant increase of N and k.

Formula (3) establishes a relation among variables N, k, ϵ, and β. A typical use of this formula consists in selecting an N within the computational limit of the used solver, ϵ according to the acceptable level of risk, and β small enough to be negligible, e.g. $\beta = 10^{-10}$, and computing from (3) the largest number k of constraints that can be discarded.

For an easy visualization of Theorem 1, we have represented in Figure 1 the region of N and k values such that condition (3) is satisfied when $\epsilon = 0.1$, $\beta = 10^{-10}$ and $d = 5$. The interpretation is that, if any pair (N, k) is picked from the grey region, then, if N constraints are sampled and k of them are removed, the obtained solution is, with high confidence $1 - 10^{-10}$, feasible for the chance-constrained problem with parameter $\epsilon = 0.1$. Further, Table 1 gives ϵ as a function of k for $N = 2000$, $\beta = 10^{-10}$ and $d = 5$. As we shall discuss in detail in the next Section 3.2, formula (3) provides a tight evaluation of the N, k guaranteeing

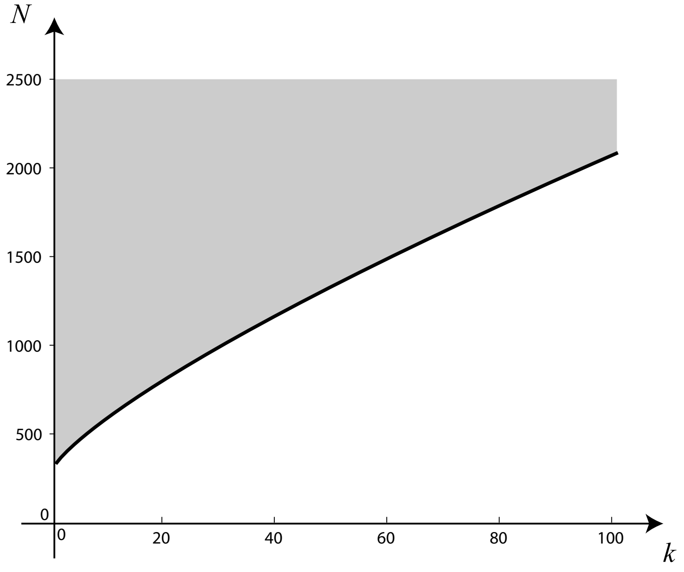

Figure 1: Grey region: values of N and k satisfying condition (3) for $\epsilon = 0.1$, $\beta = 10^{-10}$ and $d = 5$.

$(1-\epsilon)$-feasibility with confidence $1-\beta$.

3 Remarks on Theorem 1

Theorem 1 is amenable to a number of complementary observations as discussed in this section.

3.1 Trading feasibility for performance

In some applications, one may want to reach a suitable compromise between violation and performance, in which case one discards a progressively increasing number of constraints k, while inspecting the corresponding cost improvement. Theorem 1 holds for given N and k; yet, by a repeated application of the theorem one concludes that the feasibility result holds simultaneously for all k values in a certain range with confidence $1-\sum_k \beta_k$. Having a sum of β_k is not a hurdle since β_k is very small in normal situations. By e.g. applying this reasoning to the example in Table 1, one can selects a k achieving his favorite violation/performance trade-off and the corresponding ϵ holds with confidence $1 - 10 \cdot 10^{-10} = 1 - 10^{-9}$.

Example 1 (Chebyshev regression) *The* $N = 2000$ *points* (u_i, y_i), $i = 1, \dots, 2000$, *displayed in Figure 2 are independently generated in* $\mathbb{R}^2$ *according to an unknown probability density* $\mathbb{P}$. *We want to construct an interpolating polynomial of degree 3,* $y =$

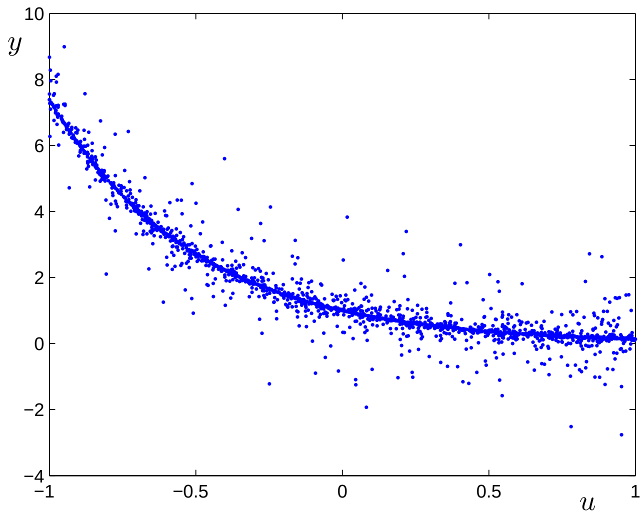

Figure 2: (u_i, y_i) data points.

$x_0 + x_1 u + x_2 u^2 + x_3 u^3$, *where* $x_0, \dots, x_3$ *are parameters to be chosen, so that a strip of minimal vertical width centered around the polynomial contains all the generated points. In mathematical terms, this problem can be cast as the following optimization program*

$$\min_{x_0,\dots,x_4} x_4 \ \text{ subject to: } \left| y_i - [x_0 + x_1 u_i + x_2 u_i^2 + x_3 u_i^3] \right| \leq x_4, \quad i = 1, \dots, 2000, \tag{4}$$

and the optimal polynomial is named "Chebyshev approximation" regressor, see e.g. [1]. The solution we have obtained with the data at hand is shown in Figure 3.

The above problem can be interpreted as an identification problem where u *is the input,* y *is the output,* $\mathbb{P}$ *is the probability describing an underlying data generation mechanism, the* $N = 2000$ *points are the data, and the strip is a data-based descriptor of the generator.*

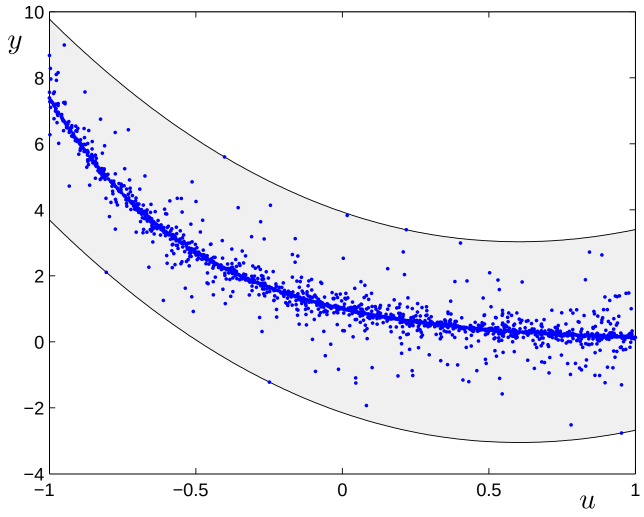

Figure 3: Strip containing all points.

k	0	10	20	30	40	50	60	70	80	90
width	6.08	4.32	3.90	3.46	3.08	2.84	2.54	2.36	2.14	2.06

Table 2: Strip widths vs. k.

Given the next input u*, the interval in the strip corresponding to that* u *provides a prediction of the associated unseen* y*. Correspondingly, one would like to have a strip of small width so as to make a tight prediction, while keeping low the probability of misclassifying the next unseen* y*, that is the probability that* (u, y) *falls outside the strip.*

As is clear from (4)*, each point corresponds to a constraint and one can easily recognize that the probability that* (u, y) *falls outside the strip is the same as the probability that one next unseen constraint is violated. Based on the result for* $k = 0$ *in Table 1, we can claim that the probability that the strip in Figure 3 misclassifies* y *is, with high confidence* $1 - 10^{-10}$*, at most* 0.017*. On the other hand, the strip width is rather large and one may want to trade some probability of misclassification for a smaller interval.*

We further removed some of the extracted (u_i, y_i) *points. Figure 4 depicts, stacked one on top of the other, the intervals obtained by a greedy removal of* $k = 10, 20, \ldots, 90$ *points. The corresponding interval widths are displayed in Table 2. This table should be assessed*

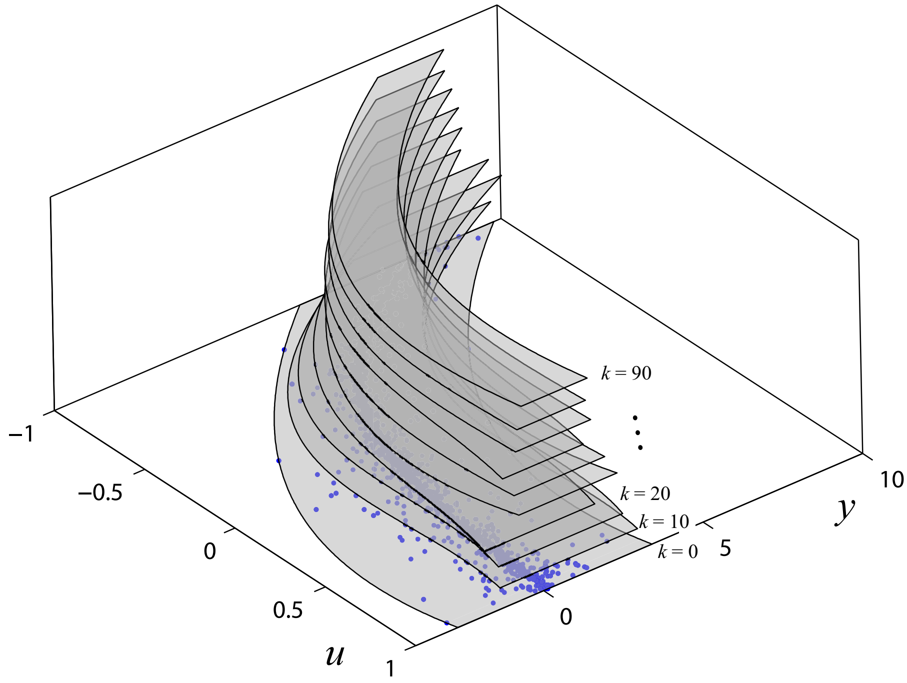

Figure 4: Strips with increasing points violation.

against Table 1 giving the probability ϵ of misclassification, and one can choose his favorite compromise. The confidence in the achieved result is $1 - 10^{-9}$, as explained before. $\square$

3.2 A remark on the quality of bound (3)

We discuss the bound (3) and its margin of improvement.

From Theorem 1, we have that $\binom{k+d-1}{k} \sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$ is an upper bound to $\mathbb{P}^N\{V(x^*) > \epsilon\}$ valid for any removal algorithm $\mathcal{A}$ and for any optimization problem P (that is for any set of constraints $\mathcal{X}_\delta$, $\delta \in \Delta$, probability $\mathbb{P}$, and cost function $c^T x$). I.e.

$$\sup_{\mathrm{P},\mathcal{A}} \mathbb{P}^N\{V(x^*) > \epsilon\} \leq \binom{k+d-1}{k} \sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}.$$

It is a fact that a lower bound to the same probability is

$$\sup_{\mathrm{P},\mathcal{A}} \mathbb{P}^N\{V(x^*) > \epsilon\} \geq \sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}, \tag{5}$$

and this sets a limit to the margin of improvement of the bound in Theorem 1. For a visual

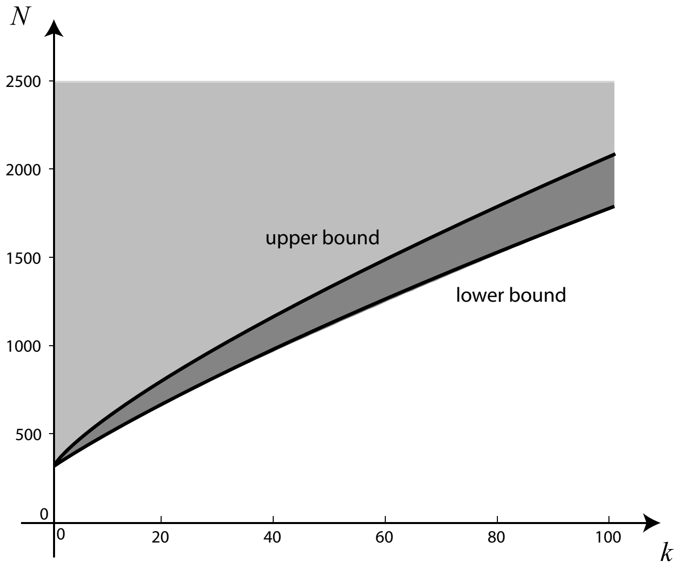

Figure 5: Grey region: values of N and k satisfying condition $\sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq \beta$; Light grey region: values of N and k satisfying condition (3); $\epsilon = 0.1$, $\beta = 10^{-10}$ and $d = 5$.

understanding of this result, in Figure 5 we have represented the region in the N, k domain where condition $\sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq \beta$ using the lower bound is satisfied for $\epsilon = 0.1$, $\beta = 10^{-10}$ and $d = 5$ superimposed to the region where (3) in Theorem 1 using the upper bound holds.

The proof of (5) is provided in Section 4.2.

3.3 An explicit formula for k

Using the Chernoff bound for the Binomial tail ([9]; see also Section 2.3.1 in [27]) yields

$$\sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq e^{-\frac{(\epsilon N - k - d + 1)^2}{2\epsilon N}}, \quad \text{for } \epsilon N \geq k + d - 1.$$

Moreover,

$$\binom{k+d-1}{k} = \frac{(k+d-1)!}{(d-1)!k!} \leq (k+d-1)(k+d-2)\cdots(k+1) \leq (k+d-1)^{d-1} \leq (\epsilon N)^{d-1},$$

where the last inequality follows from condition $\epsilon N \geq k + d - 1$.
Hence, the left-hand-side of (3) is bounded by $(\epsilon N)^{d-1} \cdot e^{-\frac{(\epsilon N - k - d + 1)^2}{2\epsilon N}}$. Given N, ϵ, β, and d, we compute a k such that

$$(\epsilon N)^{d-1} \cdot e^{-\frac{(\epsilon N - k - d + 1)^2}{2\epsilon N}} \leq \beta.$$

This latter equation can be rewritten as

$$(\epsilon N - k - d + 1)^2 \geq 2\epsilon N \ln \frac{(\epsilon N)^{d-1}}{\beta},$$

which, solved for k with the condition $\epsilon N \geq k + d - 1$, gives

$$k \leq \epsilon N - d + 1 - \sqrt{2\epsilon N \ln \frac{(\epsilon N)^{d-1}}{\beta}}. \tag{6}$$

Equation (6) is an explicit handy formula for k which can be used to compute the number of constraints that can be discarded. More precise evaluation can be obtained by numerically solving (3).
We further notice that (6) implies that $\lim_{N\to\infty} \frac{k}{N} = \epsilon$. The interpretation is that the empirical violation $\frac{k}{N}$ tends to the true violation ϵ as N is let increase.

4 Proofs

4.1 Proof of Theorem 1

Introducing the notation $\boldsymbol{\delta} := (\delta^{(1)}, \ldots, \delta^{(N)})$, in this proof we shall write $x^*(\boldsymbol{\delta})$ instead of x^* to emphasize the stochastic nature of the solution. The "bad" extractions from Δ^N leading to a solution $x^*(\boldsymbol{\delta})$ violating a portion of constraints larger than ϵ form an event

$$B = \{\boldsymbol{\delta} \in \Delta^N : \ V(x^*(\boldsymbol{\delta})) > \epsilon\},$$

and, in these notations, the theorem statement can be rephrased as $\mathbb{P}^N\{B\} \leq \beta$.

Given a subset $I = \{i_1, \ldots, i_k\}$ of k indexes from $\{1, \ldots, N\}$, let $x_I^*(\boldsymbol{\delta})$ be the optimal solution of the optimization problem where the constraints with index in I have been removed, i.e.

$$x_I^*(\boldsymbol{\delta}) := \arg\min_{x \in \mathcal{X}} c^T x \quad \text{subject to: } x \in \mathcal{X}_{\delta^{(i)}}, \quad i \in \{1, \ldots, N\} - I. \tag{7}$$

Moreover, let

$$\Delta_I^N = \{\boldsymbol{\delta} \in \Delta^N : \ x_I^*(\boldsymbol{\delta}) \text{ violates the constraints } \delta^{(i_1)}, \ldots, \delta^{(i_k)}\}. \tag{8}$$

Thus, Δ_I^N contains the multi-extractions such that removing the constraints with indexes in I leads to a solution $x_I^*(\boldsymbol{\delta})$ that violates all the removed constraints.
Since the solution of $\mathrm{RP}_{N,k}^{\mathcal{A}}$ almost surely violates k constraints (Assumption 2), it is clear that $x^*(\boldsymbol{\delta}) = x_I^*(\boldsymbol{\delta})$ for some I such that $\boldsymbol{\delta} \in \Delta_I^N$. Thus,

$$B = \{\boldsymbol{\delta} \in \Delta^N : \ V(x^*(\boldsymbol{\delta})) > \epsilon\} \subseteq \bigcup_{I \in \mathcal{I}} \{\boldsymbol{\delta} \in \Delta_I^N : \ V(x_I^*(\boldsymbol{\delta})) > \epsilon\} \tag{9}$$

up to a zero probability set, where $\mathcal{I}$ is the collection of all possible choices of k indexes from $\{1, \ldots, N\}$.
A bound for $\mathbb{P}^N\{B\}$ is now obtained by first bounding $\mathbb{P}^N\{\boldsymbol{\delta} \in \Delta_I^N : \ V(x_I^*(\boldsymbol{\delta})) > \epsilon\}$, and then summing over $I \in \mathcal{I}$.

Fix an $I = \{i_1, \ldots, i_k\}$, and write

$$\begin{aligned}
&\mathbb{P}^N\{\boldsymbol{\delta} \in \Delta_I^N : \ V(x_I^*(\boldsymbol{\delta})) > \epsilon\} \\
&\quad = \int_{(\epsilon,1]} \mathbb{P}^N\{\Delta_I^N | V(x_I^*(\boldsymbol{\delta})) = v\} \ F_V(\mathrm{d}v) \\
&\quad = \int_{(\epsilon,1]} \mathbb{P}^N\{x_I^*(\boldsymbol{\delta}) \text{ violates the constraints } \delta^{(i_1)}, \ldots, \delta^{(i_k)} | V(x_I^*(\boldsymbol{\delta})) = v\} \ F_V(\mathrm{d}v),
\end{aligned} \tag{10}$$

where F_V is the probability distribution of the random variable $V(x_I^*(\boldsymbol{\delta}))$, and $\mathbb{P}^N\{\Delta_I^N | V(x_I^*(\boldsymbol{\delta})) = v\}$ is the conditional probability of the event Δ_I^N under the condition that $V(x_I^*(\boldsymbol{\delta})) = v$ (see eq.(17), § 7, Chapter II of [26]).
To evaluate the integrand in (10), remind that $V(x_I^*(\boldsymbol{\delta})) = v$ means that $x_I^*(\boldsymbol{\delta})$ violates constraints with probability v; then, owing to that extractions are independent, the integrand

equals v^k. Substituting in (10) yields

$$\mathbb{P}^N\{\boldsymbol{\delta} \in \Delta_I^N : V(x_I^*(\boldsymbol{\delta})) > \epsilon\} = \int_{(\epsilon,1]} v^k \; F_V(\mathrm{d}v). \tag{11}$$

To proceed, we have now to appeal to a result on F_V from [7]: $F_V(v) \geq \bar{F}_V(v) := 1 - \sum_{i=0}^{d-1} \binom{N-k}{i} v^i (1-v)^{N-k-i}$, see Theorem 1 in [7] and recall that $F_V(v)$ is the distribution of the violation of a solution obtained with $N-k$ constraints. This inequality is tight, i.e. it holds with equality for a whole class of optimization problems, that called "fully-supported" in [7], Definition 3.

Now, the integrand v^k in (11) is an increasing function of v, so that $F_V(v) \geq \bar{F}_V(v)$ implies that $\int_{(\epsilon,1]} v^k \; F_V(\mathrm{d}v) \leq \int_{(\epsilon,1]} v^k \; \bar{F}_V(\mathrm{d}v)$. This can be verified by the calculation:

$$\begin{aligned}
\int_{(\epsilon,1]} v^k \; F_V(\mathrm{d}v) &= \text{[Theorem 11, §6, Chapter II of [26]]} \\
&= 1 - \epsilon^k F_V(\epsilon) - \int_{(\epsilon,1]} F_V(v) k v^{k-1} \; \mathrm{d}v \\
&\leq 1 - \epsilon^k \bar{F}_V(\epsilon) - \int_{(\epsilon,1]} \bar{F}_V(v) k v^{k-1} \; \mathrm{d}v \\
&= \int_{(\epsilon,1]} v^k \; \bar{F}_V(\mathrm{d}v).
\end{aligned}$$

Hence, $\mathbb{P}^N\{\boldsymbol{\delta} \in \Delta_I^N : V(x_I^*(\boldsymbol{\delta})) > \epsilon\}$ can finally be bounded as follows:

$$\begin{aligned}
\mathbb{P}^N\{\boldsymbol{\delta} \in \Delta_I^N : V(x_I^*(\boldsymbol{\delta})) > \epsilon\} &\leq \int_{(\epsilon,1]} v^k \; \bar{F}_V(\mathrm{d}v) \\
&= \left[\text{the density of } \bar{F}_V \text{ is } d\binom{N-k}{d} v^{d-1}(1-v)^{N-k-d}\right] \\
&= \int_{(\epsilon,1]} v^k \cdot d\binom{N-k}{d} v^{d-1}(1-v)^{N-k-d} \; \mathrm{d}v \\
&= \text{[integration by parts]} \\
&= \frac{d\binom{N-k}{d}}{(k+d)\binom{N}{k+d}} \sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}.
\end{aligned} \tag{12}$$

To conclude the proof, go back to (9) and note that $\mathcal{I}$ contains $\binom{N}{k}$ choices. Thus,

$$
\begin{aligned}
\mathbb{P}^N\{B\} &\leq \sum_{I\in\mathcal{I}} \mathbb{P}^N\{\boldsymbol{\delta}\in\Delta_I^N : \ V(x_I^*(\boldsymbol{\delta}))>\epsilon\} \\
&= \binom{N}{k}\mathbb{P}^N\{\boldsymbol{\delta}\in\Delta_I^N : \ V(x_I^*(\boldsymbol{\delta}))>\epsilon\} \\
&\leq \text{[use (12)]} \\
&\leq \binom{N}{k}\frac{d\binom{N-k}{d}}{(k+d)\binom{N}{k+d}}\sum_{i=0}^{k+d-1}\binom{N}{i}\epsilon^i(1-\epsilon)^{N-i} \\
&= \binom{k+d-1}{k}\sum_{i=0}^{k+d-1}\binom{N}{i}\epsilon^i(1-\epsilon)^{N-i} \\
&\leq \beta,
\end{aligned}
$$

where the last inequality follows from (3). □

4.2 Proof of (5)

Consider a fully-supported problem P (see Definition 3 in [7]). Equation (12) in the proof of Theorem 1 holds true in this case with equality, that is

$$
\mathbb{P}^N\{\boldsymbol{\delta}\in\Delta_I^N : \ V(x_I^*(\boldsymbol{\delta}))>\epsilon\} = \frac{d\binom{N-k}{d}}{(k+d)\binom{N}{k+d}}\sum_{i=0}^{k+d-1}\binom{N}{i}\epsilon^i(1-\epsilon)^{N-i}, \tag{13}
$$

with $x_I^*(\boldsymbol{\delta})$ and Δ_I^N defined as in (7) and (8).

For a multi-extraction $\boldsymbol{\delta}$, let $\#$ be the number of solutions of level k (that is solutions that violate exactly k of the extracted constraints) whose violation is more than ϵ. From (7) and (8), one easily sees that

$$
\# = \sum_{I\in\mathcal{I}} 1_{\{\boldsymbol{\delta}\in\Delta_I^N : \, V(x_I^*(\boldsymbol{\delta}))>\epsilon\}},
$$

where 1_A denotes the indicator function of set A. We now have that

$$\begin{aligned}
\mathrm{E}_{\Delta^N}[\#] &= \int_{\Delta^N} \sum_{I \in \mathcal{I}} 1_{\{\boldsymbol{\delta} \in \Delta_I^N :\, V(x_I^*(\boldsymbol{\delta})) > \epsilon\}} \, \mathbb{P}^N\{\mathrm{d}\boldsymbol{\delta}\} \\
&= \sum_{I \in \mathcal{I}} \mathbb{P}^N\{\boldsymbol{\delta} \in \Delta_I^N \,:\, V(x_I^*(\boldsymbol{\delta})) > \epsilon\} \\
&= [\text{use (13) and recall that } \mathcal{I} \text{ contains } \binom{N}{k} \text{ choices}] \\
&= \binom{N}{k} \frac{d\binom{N-k}{d}}{(k+d)\binom{N}{k+d}} \sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \\
&= \binom{k+d-1}{k} \sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}. \qquad (14)
\end{aligned}$$

Now, letting $S(\#)$ be the set in Δ^N where $\# \neq 0$, that is the set of multi-extractions such that at least one solution of level k violates more than ϵ, the algorithm $\overline{\mathcal{A}}$ that always selects the solution $x^*(\boldsymbol{\delta})$ of level k with the largest violation leads to the fact that $V(x^*(\boldsymbol{\delta})) > \epsilon$ holds on $S(\#)$, that is with a probability $\mathbb{P}^N\{S(\#)\}$. Nobody to date knows the exact value of $\mathbb{P}^N\{S(\#)\}$, but it turns out that we can compute a lower bound to it. In fact, a fully supported problem has $\binom{k+d-1}{k}$ solutions of level k, see e.g. [15], so that $\# \leq \binom{k+d-1}{k}$. Using this fact in (14) yields:

$$\binom{k+d-1}{k} \sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} = \mathrm{E}_{\Delta^N}[\#] \leq \binom{k+d-1}{k} \cdot \mathbb{P}^N\{S(\#)\},$$

from which $\mathbb{P}^N\{S(\#)\} \geq \sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$. Since $\mathbb{P}^N\{S(\#)\}$ is the probability that the solution of algorithm $\overline{\mathcal{A}}$ violates more than ϵ, the found number $\sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$ represents a lower bound to $\sup_{\mathrm{P},\mathcal{A}} \mathbb{P}^N\{V(x^*(\boldsymbol{\delta})) > \epsilon\}$. □

5 Optimality results

In this section we establish the result that the objective value of CCP_ϵ can be approached at will, provided that sampled constraints are optimally removed. Though at the present state of knowledge optimal removal can be impractical due to the ensuing high computational burden, this study has a theoretical interest and sheds further light on the relation between chance-constrained optimization and its randomized counterpart.

Let $\mathcal{A}_{opt}$ be the optimal constraints removal algorithm which leads – among all possible eliminations of k constraints out of N – to the best possible improvement in the cost objective; further, let x^*_{opt} and J^*_{opt} be the corresponding optimal solution and cost value. We have the following theorem.

Theorem 2 *Let $\beta \in (0,1)$ be any small confidence parameter value, and let $\nu \in (0,\epsilon)$ be a performance degradation parameter value. If N and k are such that*

$$\binom{k+d-1}{k} \sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} + \sum_{i=k+1}^{N} \binom{N}{i} (\epsilon-\nu)^i (1-\epsilon+\nu)^{N-i} \leq \beta, \tag{15}$$

then

(i) $V(x^*_{opt}) \leq \epsilon$

(ii) $J^*_{opt} \leq J^*_{\epsilon-\nu}$

simultaneously hold with probability at least $1-\beta$.

As in Theorem 1, point (i) is a feasibility result. Instead, point (ii) states that the performance achieved by x^*_{opt} is no worse than the performance of $\mathrm{CCP}_{\epsilon-\nu}$, where ν is a degradation margin. A result similar to (ii) has also been independently established in [14]. A simple example illustrating Theorem 2 is provided in Appendix A, part A.2.

Proof: Let

$$B_i = \{\boldsymbol{\delta} \in \Delta^N : \ V(x^*_{opt}(\boldsymbol{\delta})) > \epsilon\},$$
$$B_{ii} = \{\boldsymbol{\delta} \in \Delta^N : \ J^*_{opt}(\boldsymbol{\delta}) > J^*_{\epsilon-\nu}\}.$$

We have to prove that $\mathbb{P}^N\{B_i \cup B_{ii}\} \leq \beta$.

Write $\mathbb{P}^N\{B_i \cup B_{ii}\} \leq \mathbb{P}^N\{B_i\} + \mathbb{P}^N\{B_{ii}\}$. By Theorem 1, $\mathbb{P}^N\{B_i\}$ is bounded by $\binom{k+d-1}{k} \sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$. We here bound $\mathbb{P}^N\{B_{ii}\}$.

For the sake of simplicity, in what follows we assume that a solution $x^*_{\epsilon-\nu}$ of $\mathrm{CCP}_{\epsilon-\nu}$ exists. If not, the result is similarly established by a limit reasoning.

Let $\overline{\Delta}_{\epsilon-\nu}$ be the subset of Δ formed by those constraints which are violated by $x^*_{\epsilon-\nu}$. Call $R(\boldsymbol{\delta})$ the number of constraints among the N extracted ones that fall in $\overline{\Delta}_{\epsilon-\nu}$. The objective value obtained by eliminating the $R(\boldsymbol{\delta})$ constraints in $\overline{\Delta}_{\epsilon-\nu}$, say $\overline{J}(\boldsymbol{\delta})$, cannot be worse than $J^*_{\epsilon-\nu}$: $\overline{J}(\boldsymbol{\delta}) \leq J^*_{\epsilon-\nu}$. Thus, if $J^*_{opt}(\boldsymbol{\delta}) > J^*_{\epsilon-\nu}$, then $R(\boldsymbol{\delta}) > k$ for, otherwise, $J^*_{opt}(\boldsymbol{\delta}) \leq \overline{J}(\boldsymbol{\delta}) \leq J^*_{\epsilon-\nu}$. Therefore,

$$\begin{aligned}
\mathbb{P}^N\{B_{ii}\} &= \mathbb{P}^N\{J^*_{opt}(\boldsymbol{\delta}) > J^*_{\epsilon-\nu}\} \\
&\leq \mathbb{P}^N\{R(\boldsymbol{\delta}) > k\} \\
&= [\text{probability that more than } k \text{ among } N \text{ extractions fall in } \overline{\Delta}_{\epsilon-\nu}] \\
&= \sum_{i=k+1}^{N} \binom{N}{i} (\mathbb{P}\{\overline{\Delta}_{\epsilon-\nu}\})^i (1-\mathbb{P}\{\overline{\Delta}_{\epsilon-\nu}\})^{N-i} \\
&\leq \sum_{i=k+1}^{N} \binom{N}{i} (\epsilon-\nu)^i (1-\epsilon+\nu)^{N-i},
\end{aligned}$$

where the last inequality follows from observing that $\mathbb{P}\{\overline{\Delta}_{\epsilon-\nu}\} \leq \epsilon - \nu$.

Wrapping up the above results, we finally have

$$\begin{aligned}
\mathbb{P}^N\{B_i \cup B_{ii}\} &\leq \mathbb{P}^N\{B_i\} + \mathbb{P}^N\{B_{ii}\} \\
&\leq \binom{k+d-1}{k} \sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} + \sum_{i=k+1}^{N} \binom{N}{i} (\epsilon-\nu)^i (1-\epsilon+\nu)^{N-i} \\
&\leq \beta,
\end{aligned}$$

where the last inequality is given by (15). $\square$

Theorem 2 holds true provided that constraints are optimally removed, a nontrivial combinatorial problem in general. A brute-force approach (where one solves the optimization problems for all possible combinations of $N-k$ constraints taken from the initial set of N constraints and then choose that combination resulting in the lowest value of the objective function) requires to solve $\binom{N}{k}$ optimization problems, a truly large number in general. Along a different approach, constraints can be removed sequentially by choosing each time among active constraints only. This reduces the actual number of possible combinations of constraints to be taken into account resulting in an algorithm that solves $O(\min\{N \cdot d^k, N \cdot k^d\})$ optimization problems, see [15]. Even this way, however, the computational burden becomes rapidly prohibitive for values of k and d that are not both very small.

5.1 Existence of N and k

In Theorem 2, ν measures the performance mismatch between x^*_{opt} and the solution of CCP$_\epsilon$. For any small ν, N and k satisfying condition (15) always exist, a result shown in this section. As expected, N and k increase as ν approaches zero.

To start with, consider (15) and split β evenly between the two terms in the left-hand-side of this equation, that is impose that both terms are less than $\beta/2$. A condition for the first term to be less than $\beta/2$ has been already established in equation (6) (substitute $\beta/2$ for β in that equation). We here work on the second term.

Similarly to (15), we use a Chernoff bound for the Binomial tail, this time the right tail Chernoff bound ([9]; see also Section 2.3.1 in [27]) stating that

$$\sum_{i=k+1}^{N} \binom{N}{i} (\epsilon-\nu)^i (1-\epsilon+\nu)^{N-i} \leq e^{-\frac{((\epsilon-\nu)N-k-1)^2}{3(\epsilon-\nu)N}}, \quad \text{for } (k+1)/2 \leq (\epsilon-\nu)N \leq k+1.$$

Further, imposing that $e^{-\frac{((\epsilon-\nu)N-k-1)^2}{3(\epsilon-\nu)N}}$ is less than $\beta/2$ and solving for k yields

$$k \geq (\epsilon-\nu)N - 1 + \sqrt{3(\epsilon-\nu)N \ln \frac{2}{\beta}}. \tag{16}$$

In (6), the term linear in N has slope ϵ, while the other terms are sub-linear. Instead, in (16) the slope is $(\epsilon-\nu)$. Since $\epsilon > (\epsilon-\nu)$, for N large enough there is a gap between the bounds expressed by (6) and (16), and, consequently, an N and a k can be found that simultaneously satisfy (6) and (16).

Appendix

A An illustrative example

The extremely simple example of this appendix illustrates the nature of some results contained in this paper. It goes without saying that applying a randomized procedure to such a simple situation has no practical interest to attain solvability.

Let us consider the chance-constrained problem:

$$\min_{x \in [0,1]} x$$

$$\text{subject to: } x \geq \delta \text{ with probability } \mathbb{P} \geq 1 - \epsilon.$$

Here, $\mathcal{X} = [0,1]$, $\mathcal{X}_\delta = \{x : x \geq \delta\}$, while $\delta \in \Delta = [0,1]$. Suppose also that $\mathbb{P}$ is uniform over Δ.

In this simple setting, the CCP$_\epsilon$ optimum is achieved by removing the set $[1-\epsilon, 1]$ from Δ, leading to the optimal solution $x^*_\epsilon = 1 - \epsilon = J^*_\epsilon$ and to $V(x^*_\epsilon) = \epsilon$. Throughout we take $\epsilon = 0.2$.

Turn now to consider the randomized optimization program where k constraints are removed. Given a multi-extraction $(\delta^{(1)}, \dots, \delta^{(N)})$, x^* is obtained by removing the k largest $\delta^{(i)}$'s and by letting $x^* =$ the $(k+1)^{\text{th}}$ largest $\delta^{(i)}$ value. Also, $V(x^*) = 1 - x^*$ and $J^* = x^*$, and all these quantities are random variables as they depend on the multi-extraction $(\delta^{(1)}, \dots, \delta^{(N)})$.

A.1 The need for constraints removal

Figure 6 depicts the probability density function of $V(x^*)$ when $N = 15$ and $k = 0$, that is, according to the philosophy of papers [5, 6, 7], no constraints are removed. One sees that $V(x^*) \leq \epsilon = 0.2$ for most of the multi-extractions. On the other hand, the density concentrates near the zero value. This means that the violation of x^* will be much less than that for x^*_ϵ with high enough probability, entailing that the objective value of x^* will be poor as compared to the chance-constrained solution.

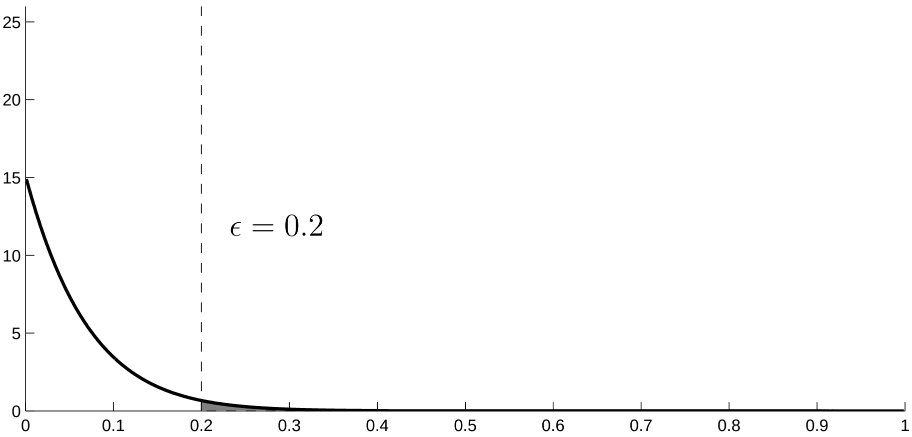

Figure 6: The probability density function of $V(x^*)$ for $N = 15$ and $k = 0$; grey area represents the probability that $V(x^*) > \epsilon$.

Selecting $N = 552$ and $k = 93$ leads instead to the probability density function in Figure 7. Constraints discarding generated solutions for which $V(x^*) > \epsilon$ has the same probability

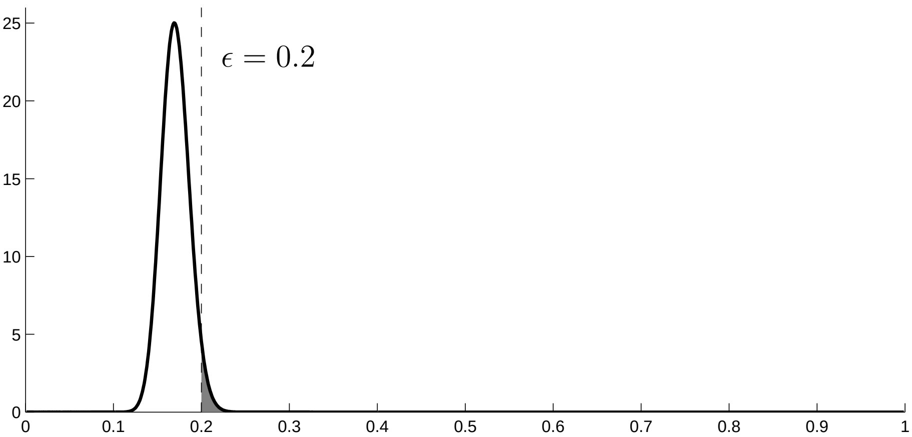

Figure 7: The probability density function of $V(x^*)$ for $N = 552$ and $k = 93$; grey area represents the probability that $V(x^*) > \epsilon$.

as before, but the violation approaches the desired violation level $\epsilon = 0.2$ of the chance-constrained problem for most multi-extractions. This is the beneficial effect of constraints removal.

A.2 Optimality results

In this subsection we illustrates the results in the optimality Theorem 2.

Observe first that, in this 1-dimensional example, the considered removal algorithm coincides with the optimal removal algorithm $\mathcal{A}_{opt}$, i.e. $x^* = x^*_{opt}$.

Again referring to $N = 552$ and $k = 93$, Figure 8 further displays the region B_i where $V(x^*) > \epsilon$ along with region B_{ii} where $J^*_{opt} = 1 - V(x^*_{opt}) > 1 - (\epsilon - \nu) = J^*_{\epsilon-\nu}$ for $\nu = 0.05$. Here, $\mathbb{P}^N\{B_i \cup B_{ii}\} = 0.1352$.

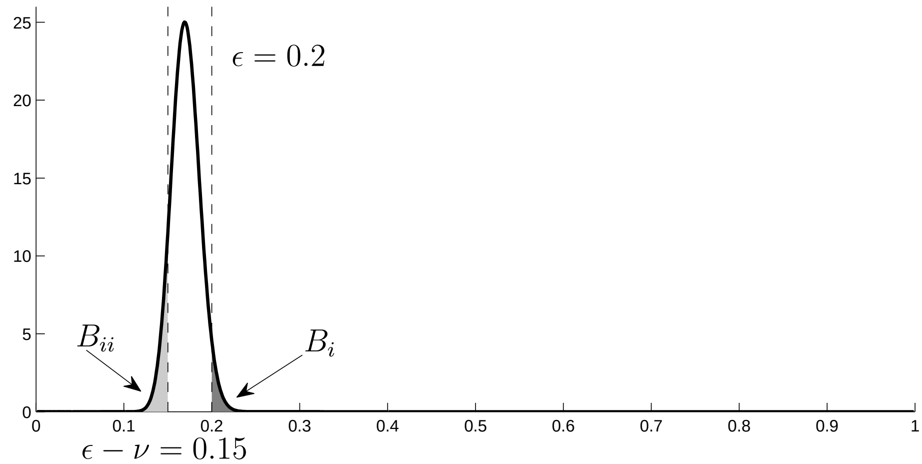

Figure 8: The probability density function of $V(x^*)$ for $N = 552$ and $k = 93$; B_i is the region where $V(x^*) > \epsilon$, while B_{ii} where $J^*_{opt} > J^*_{\epsilon-\nu}$.

Thus, $N = 552$ and $k = 93$ suffice to simultaneously guarantee that $V(x^*_{opt}) \leq 0.2$ and $J^*_{opt} \leq J^*_{0.15}$ with probability 0.8648. Interestingly enough, applying Theorem 2 provides in general upper-bounds for N and k; however, in the present 1-dimensional case, substituting $\epsilon = 0.2$, $\nu = 0.05$, and $\beta = 1 - 0.8648 = 0.1352$ in (15) just returns $N = 552$ and $k = 93$.

References

[1] G. Appa and C. Smith. On ℓ_1 and Chebyshev estimation. *Mathematical programming*, 5:73–87, 1973.

[2] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88:411–424, 2000.

[3] A. Ben-Tal and A. Nemirovski. Robust optimization – methodology and applications. *Mathematical Programming Series B*, 92:453–480, 2002.

[4] D. Bertsimas and M. Sim. The price of robustness. *Operation Research*, 52(1):35–53, 2004.

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

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

[7] M.C. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization, to appear. Available at: http://www.optimization-online.org/DB_HTML/2007/07/1727.html*, 2008.

[8] A. Charnes and W.W. Cooper. Chance constrained programming. *Management Science*, 6:73–79, 1959.

[9] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

[10] D.P. de Farias and B. Van Roy. On constraints sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478, 2004.

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

[12] E. Erdoğan and G. Iyengar. Ambiguous chance constrained problems and robust optimization. *Mathematical Programming Series B*, 107(1-2):37–61, 2006.

[13] R. Henrion and C. Strugarek. Convexity of chance constraints with independent random variables. *Computational Optimization and Applications*, 2007. published on-line.

[14] J. Luedtke and S. Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699, 2008.

[15] J. Matoušek. On geometric optimization with few violated constraints. *Discrete & Computational Geometry*, 14:365–384, 1995.

[16] L.B. Miller and H. Wagner. Chance-constrained programming with joint constraints. *Operations Research*, 13:930–945, 1965.

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

[18] A. Nemirovski and A. Shapiro. Scenario approximations of chance constraints. In G. Calafiore and F. Dabbene, editors, *Probabilistic and Randomized Methods for Design under Uncertainty*, London, 2006. Springer-Verlag.

[19] G.C. Pflug. Stochastic optimization and statistical inference. In A. Ruszczyǹski and A. Shapiro, editors, *Stochastic Programming*, volume 10 of *handbooks in operations research and managment science*, London, UK, 2003. Elsevier.

[20] A. Prèkopa. On probabilistic constrained programming. In *Proceedings of the Princeton Symposium on Mathematical Programming*, Princeton, USA, 1970. Princeton University Press.

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

[22] A. Prèkopa. Probabilistic programming. In A. Ruszczyǹski and A. Shapiro, editors, *Stochastic Programming*, volume 10 of *handbooks in operations research and managment science*, London, UK, 2003. Elsevier.

[23] A. Ruszczyǹski and A. Shapiro. Stochasting programming models. In A. Ruszczyǹski and A. Shapiro, editors, *Stochastic Programming*, volume 10 of *handbooks in operations research and managment science*, London, UK, 2003. Elsevier.

[24] A. Ruszczyǹski and A. Shapiro. *Lectures on stochastic programming.* preprint, 2008.

[25] A. Shapiro. Monte-Carlo sampling methods. In A. Ruszczyǹski and A. Shapiro, editors, *Stochastic Programming*, volume 10 of *handbooks in operations research and managment science*, London, UK, 2003. Elsevier.

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

[27] M. Vidyasagar. *A theory of learning and generalization: with applications to neural networks and control systems.* Springer-Verlag, London, UK, 1997.