Computational Management Science manuscript No.
(will be inserted by the editor)

The Wait-and-Judge Scenario Approach applied to Antenna Array Design

A. Carè · S. Garatti · M. C. Campi

Received: date / Accepted: date

Abstract The scenario optimisation approach is a methodology for finding solutions to uncertain convex problems by resorting to a sample of data, which are called "scenarios". In a min-max set-up, the solution delivered by the scenario approach comes with tight probabilistic guarantees on its risk defined as the probability that an empirical cost threshold will be exceeded when the scenario-based solution is adopted. While the standard theory of scenario optimisation has related the risk of the data-based solution to the number of optimisation variables, a more recent approach, called the "wait-and-judge" scenario approach, enables the user to assess the risk of the solution in a data-dependent way, based on the number of decisive scenarios ("support scenarios"). The aim of this paper is to illustrate the potentials of the wait-and-judge approach for min-max sample-based design and we shall consider to this purpose an antenna array design problem.

Keywords Scenario approach · Data-driven optimisation · Min-max design

A. Carè
University of Brescia
via Branze 38, 25123 - Brescia, Italy
E-mail: algo.care@unibs.it

S. Garatti
Politecnico di Milano
P.zza L. da Vinci 32, 20133 - Milano, Italy
Tel.: +39 - 02 - 2399 - 3650
Fax: +39 - 02 - 2399 - 3412
E-mail: simone.garatti@polimi.it

M. C. Campi
University of Brescia
via Branze 38, 25123 - Brescia, Italy
Tel.: +39 - 030 - 3715458
Fax: +39 - 030 - 380014
E-mail: marco.campi@unibs.it

1 Introduction

A fundamental approach to make decisions consists in choosing the values of the decision variables so as to minimise a cost function. Although sometimes the value of the cost function can be considered to be fully determined by the decision variables, in most of the situations it is more realistic to assume that the cost function depends also on an uncertain variable. In this case, optimising the cost function only with respect to one fixed value of the uncertain variable generates decisions that are fragile with respect to real world situations.

Traditional ways to deal with an uncertain variable require modelling the uncertainty: as a preliminary step in the decision problem, one models either the set of possible values that the uncertain variable can take or the probability according to which these values are distributed. Once a model of the uncertainty is available, the decision variables are chosen by re-casting the decision problem as a suitable optimisation problem. For example, one can i) minimise the value of the cost function by averaging over the distribution of the uncertain variable, see e.g. [18,32]; ii) solve a robust optimisation problem, where the maximum of the cost function with respect to all the values of the uncertain variable is minimised, see e.g. [33,3,20,8,7]; iii) solve a chance-constrained problem, see e.g. [17,30,19,32,21,22]; etc.

In this paper we focus on the so-called "scenario approach" which is a *direct* method to make a decision in the presence of uncertainty. The scenario approach is a direct method in the sense that it does not require any modelling of the uncertain variable. As a preliminary step, the scenario approach prescribes only to collect a set of N instances of the uncertain variable, which are called "scenarios". By instantiating the cost function with the N realisations of the uncertain variable, N different instances of the cost function are obtained, and the prescribed decision (scenario solution) is the one that minimises the worst-case cost, i.e., the highest cost value among the N available costs. Said more briefly, the scenario approach prescribes to adopt a solution that is robust with respect to the data that have been observed. In general, the data-based solution might perform bad on new instances of the uncertain variable. In order to characterise this eventuality, we define the *risk* of the scenario solution as the probability that the maximum observed cost value (that is computed with respect to the N observed scenarios) is exceeded when a new realisation of the uncertain variable is considered. A reliable assessment of the risk is crucial in whether to accept and adopt the decision or to further refine it. In a little bit more sophisticated decision processes, the assessment of the risk can also guide the pricing of data-based designed products or even the subscription of quality-of-service agreements on the outcome of the design process, in such a way that the probability of refunding a client is kept under control and no loss is incurred on average over the sold design solutions.

We show that, from the mathematical point of view, the event that the empirical cost of the scenario solution is exceeded is the event that a random constraint (suitably parametrised in the uncertain variable) is violated by the scenario solution. Characterising the probability of such "constraint violation" is the main focus of the *theory of the scenario approach*. This theory was initiated in [10], and has ever since attracted an increasing interest, see e.g. [28,9, 13,34,35,11,31,16,38,1]. We refer also to [9,12,32,34,27,29,24] for a comparison of scenario optimisation with other methods in stochastic optimisation. This paper aims at illustrating the potentials of a new assessment methodology introduced in [14] that is called the "wait-and-judge" scenario approach.

Building upon the results in [14], we show in this paper how the "wait-and-judge" theory can be used in practice to characterise the risk of the data-based decision in a data-dependent way, without any knowledge on the probability distribution of the uncertain variable. The risk is evaluated using the same data (scenarios) that are used during the optimisation process, that is, no independent validation set is required to assess the value of the risk. This is of great importance in data-driven design when scenarios are a *costly and limited resource*. The punchline of the "wait-and-judge" approach is that the risk depends on how many scenarios among those that have been observed were really decisive in determining the scenario solution: when the solution depends only on a few scenarios its risk is likely (in a sense that is rigorously defined) to be small.

Although a quick reference to data-driven decision-making can be found in [14], no further explanations and no examples were provided therein. This paper aims at complementing the mathematical theory of [14] with an illustration of its potentials for safe decision-making, where real data are used to robustify a decision. The approach is illustrated on an *antenna array design* problem. Antenna array design is a traditional testbed domain for uncertain optimisation methods, in particular for robust optimisation, see [5,6,3], and has recently gained popularity for its role in modern wireless communication systems, see e.g. [26].

In the following Section 2, the main concepts of antenna array design are introduced and made concrete by resorting to a numerical instance of the problem. This set-up works as an ongoing example throughout the paper. In the same section, we show that neglecting the uncertainty on the actuation errors leads to dramatic design failures. Section 3 is devoted to the scenario approach. First, we illustrate a simple data-based optimisation procedure that takes uncertainty into account by resorting to a collection of scenarios. Then, the main results from the theory of the scenario approach, and in particular the recent "wait-and-judge" approach, are recalled and reformulated for convex min-max problems. The scenario-based optimisation procedure is revisited in the light of the theory, and guarantees are issued on the quality of our scenario-based design for the antenna array. Finally, the guaranteed bounds on the risk computed according to the "wait-and-judge" approach are validated by resorting to Monte Carlo simulations. Conclusions are drawn in Section 4.

2 Antenna Array Design

In this paper, we follow the formulation of the problem provided in [6]. The reader is also referred to [3] for more details.

An antenna is characterised by its *sensitivity diagram*, which describes the capability of the antenna to receive signals from different directions; the importance of the sensitivity diagram can also be understood in relation to a relevant theorem in electromagnetism, the reciprocity theorem, according to which the sensitivity diagram is equivalent to the far-field radiation pattern of the antenna when used for transmitting.

Mathematically, the sensitivity diagram can be written as a function of a vector v, which represents a direction in the three-dimensional space, and is proportional to $|D(v)|^2$, where $D(v)$ is a complex function of v, simply called the *diagram* of the antenna. $D(v)$, for each v, returns the antenna receiving gain and phase shift for an electromagnetic wave propagating along the direction v.

An *antenna array* is just a set of multiple antennas that work together. From the electromagnetic point of view, the antenna array can also be thought of as a single antenna with its own sensitivity diagram. Since the diagram of the overall array can vary considerably depending on the locations of the individual component antennas and their exciting currents, combining antennas into an array is an easy and flexible way to design an antenna with a desired sensitivity diagram.

For the sake of concreteness and simplicity of presentation, we consider here antenna arrays consisting of n coplanar and concentric ring-shaped antennas. Due to circular geometry, the diagram of a ring-shaped antenna depends only on the latitude angle θ between a given direction in the 3D space and the plane to which the antenna belongs. Moreover, such a diagram turns out to be real-valued (which means that the phase shift is either 0 or π) and can be written as

$$D_\ell(\theta) = \frac{1}{2}\int_0^{2\pi} \cos(2\pi\nu_\ell \cos(\theta)\cos(\phi))\mathrm{d}\phi, \tag{1}$$

where $\ell = 1, 2, \ldots, n$ denotes the identification number of the antenna, and ν_ℓ is a technical parameter representing the ratio of the ring radius over the wavelength. The diagram of the antenna array, $D_a(\theta)$, can be obtained by a weighted sum of the individual components according to the formula $D_a(\theta) = \sum_{\ell=1}^{n} x_\ell D_\ell(\theta)$, where each x_ℓ is a user-chosen weight (in general, a complex number) that determines the actuation that excites the ℓ-th antenna in the array. These weights are our *design parameters*.

The objective of antenna array design is that of choosing the weights $x_1, \dots, x_n$ in such a way that the diagram of the array $D_a(\theta)$ is as close as possible to a desired diagram $T(\theta)$.

In what follows, the array design problem is cast as an optimisation problem, and is illustrated on a numerical instance. Although the weights $x_1, \dots, x_n$ can be complex numbers in general, following [6] we restrict ourselves to the case where they are real. As we shall notice, this is without loss of generality for the problem that we consider in this paper.

2.1 Naive set-up – without uncertainty

Given a target diagram $T(\theta)$, a naive solution to the antenna array design problem can be found by minimising the *cost function*

$$\max_{\theta \in [0,\pi/2]} \left| \sum_{\ell=1}^{n} x_\ell D_\ell(\theta) - T(\theta) \right| \tag{2}$$

that penalises the uniform distance between the target diagram and the designed diagram. More precisely, we solve the problem

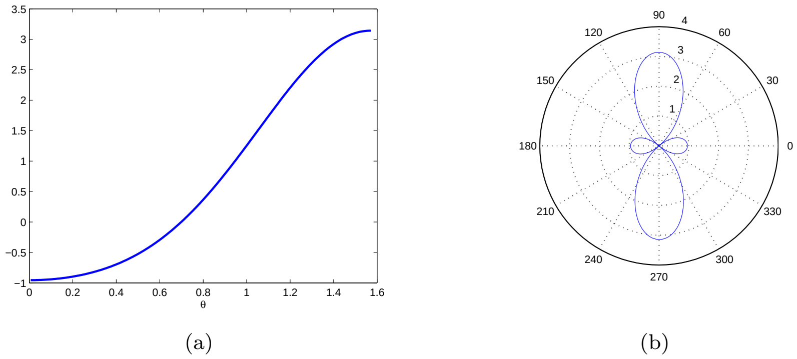
$$\begin{aligned} &\min_{h, x_1, \dots, x_n} \; h \\ &\text{subject to: } \max_{\theta \in [0,\pi/2]} \left| \sum_{\ell=1}^{n} x_\ell D_\ell(\theta) - T(\theta) \right| \leq h, \\ &\qquad |x_\ell| \leq c, \; \ell = 1, \dots, n, \end{aligned} \tag{3}$$

where c is a bound on the weights, which is introduced to take into account the physical actuation limits. Denoting by $(h^*, x_1^*, \dots, x_n^*)$ the solution to (3), we note that h^* is the cost incurred when the optimal design $(x_1^*, \dots, x_n^*)$ replaces the generic vector $(x_1, \dots, x_n)$ in (2).

2.1.1 Numerical Instance

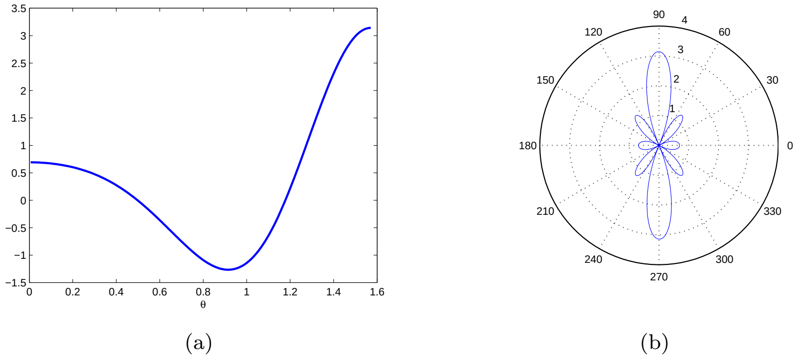
Concretely, we work under the following specifications.

Individual Antennas: We have at our disposal $n = 100$ individual antennas with diagrams as in (1), with parameter $\nu_\ell = \frac{\ell}{10}$ (this can be phrased by saying that our antennas are equally spaced). In Figures 1a, 2a, 3a the values of the diagram functions $D_\ell(\theta)$ for $\ell = 5, 10, 15$ are shown. The corresponding diagrams of $|D_\ell(\theta)|$ in a polar coordinate frame are in Figures 1b, 2b, 3b. These diagrams represent how the antenna radiates in different directions, where 0 corresponds to the plane of the array and 90 to the direction orthogonal to the plane of the array. It should be noted that the diagrams of the individual antennas exhibit various extra “lobes” (maxima) besides the main one, showing that radiation tends to be dispersed in different directions or, equivalently, that these antennas will be sensitive to signals coming from secondary directions and are prone to interferences.



(a) (b)

Fig. 1: The diagram $D_5(\theta)$, (a), and the corresponding polar diagram ($|D_5(\theta)|$), (b).



(a) (b)

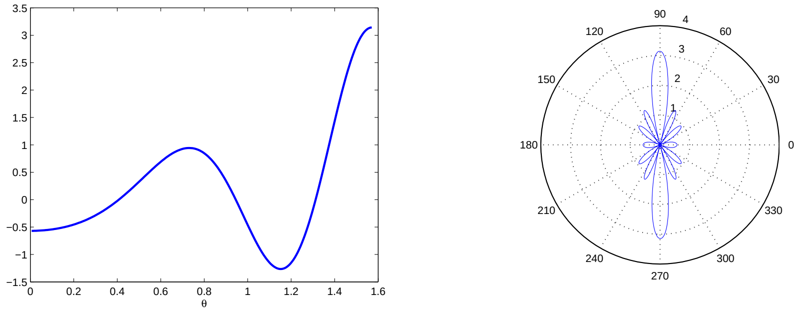
Fig. 2: The diagram $D_{10}(\theta)$, (a), and the corresponding polar diagram ($|D_{10}(\theta)|$), (b).

Target diagram $T(\theta)$: as for the target diagram, we take

$$T(\theta) = \begin{cases} 0, & \text{if } \theta \in [0, \frac{\pi}{2} - \frac{\pi}{12}] \\ \cos\left(6\left(\theta - \frac{\pi}{2}\right)\right), & \text{otherwise;} \end{cases} \tag{4}$$

see Figure 4. Note that, differently from the diagrams of individual antennas in Figures 1b, 2b, 3b, the target diagram has no side lobes, and is concentrated in the small cone $\frac{\pi}{2} - \frac{\pi}{12} \leq \theta \leq \frac{\pi}{2} + \frac{\pi}{12}$, that is, we have a *directional* target diagram. Directionality has several advantages, as it reduces the power consumption by avoiding dispersion of the signal in directions that are not necessary, and minimises interferences with other devices.

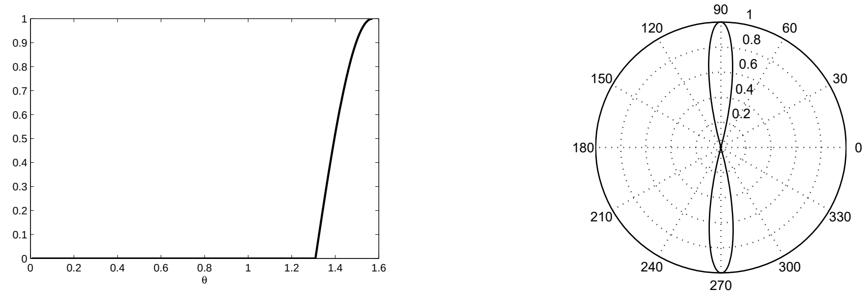
Weights: Weights are subject to actuation constraints as represented by a bound $c = 5$ in (3). We also observe that restricting the weights to real values



(a)

(b)

Fig. 3: The diagram $D_{15}(\theta)$, (a), and the corresponding polar diagram $(|D_{15}(\theta)|)$, (b).



(a)

(b)

Fig. 4: The target diagram $T(\theta)$, (a), and the corresponding polar diagram $(|T(\theta)|)$, (b).

is a natural choice in the context of the above specifications. In fact, the target function and the individual antennas are real-valued, so that adding imaginary parts to the weights can only increase the value of the cost function.

2.1.2 Results and comments

Problem (3) was solved numerically by using the CPLEX [23] function cplexlp in MATLAB [25], where the maximum over $[0, \pi/2]$ was approximated by the maximum over a grid of 240 equally spaced points on the same interval.[1] After solving Problem (3), we used the obtained weights $x_1^*, \dots, x_n^*$ to compute the diagram of the antenna array according to the formula $D_a(\theta) = \sum_{\ell=1}^{n} x_\ell^* D_\ell(\theta)$, and we obtained the diagram in Figure 5. In the same figure, the target diagram

[1] The same approximation was used for solving all the other numerical instances throughout the paper.

is also shown. As it can be seen, the approximation is very good: the cost h^*, which measures the distance between $D_a(\theta)$ and the target diagram $T(\theta)$, turns out to be $h^* = 0.0138$. On the other hand, this design procedure is naive because in real world situations there are always *actuation errors* that modify the effect of the designed weights $x_1^*, \dots, x_n^*$ on the array diagram. One might hope that small actuation error do not affect the performance significantly. However, this is *not* the case, and the fragility of the naive design is a well-known problem. In the following, we show that our naive design is no exception, which implies that uncertainty has to be taken into account.

2.1.3 Actuation uncertainty

We describe actuation errors as multiplicative errors so that each term

$$x_\ell D_\ell(\theta) \text{ is replaced by } (1+\delta_\ell)x_\ell D_\ell(\theta)$$

in the definition of the antenna array diagram. Corresponding to the naive solution $x_1^*, \dots, x_{100}^*$, the array diagram can therefore be written as $\tilde{D}_a(\theta) = \sum_{\ell=1}^n (1+\delta_\ell)x_\ell^* D_\ell(\theta)$. In Figure 6 we show the diagrams of the antenna array in the presence of actuation errors. It is worth remarking that, in a data-based perspective, the actuation errors $(1+\delta_1), \dots, (1+\delta_{100})$ have to be *measured* on the field. Here, however, for the sake of reproducibility of our numerical results, we have simulated their values by generating artificially 500 instances of the vector $\delta = (\delta_1, \dots, \delta_{100})$, according to the formula $\delta = (0.05r^{105} \cdot u_1, \dots, 0.05r^{105} \cdot u_{100})$ where each u_ℓ, $\ell = 1, \dots, 100$ was independently uniformly sampled from $[-1, 1]$ and r was independently uniformly sampled from $[0, 1]$. Note that r is a common factor that affects all the actuation errors in a vector δ, thus accounting for common environmental phenomena, coupling, etc., and the exponent 105 makes small values of r highly probable. From Figure 6, the conclusion is clear: neglecting actuation errors results in a dramatic design failure, a fact previously noted by other authors, see, e.g., [4].

3 Data-driven design through the Scenario Approach

The scenario approach requires that we make our design decision based on a set of experiments that represent possible uncertainty instances. These uncertainty instances are called *scenarios*. In the antenna array design of Section 2, a scenario is a vector of $n = 100$ actuation errors, i.e., an instance of the 100-valued vector $\delta = (\delta_1, \dots, \delta_{100})$. We collected $N = 500$ scenarios, and we denoted the i-th scenario by $\delta^{(i)} = (\delta_1^{(i)}, \dots, \delta_{100}^{(i)})$. The scenario-based design is obtained by solving the min-max scenario program (in epigraphic form)

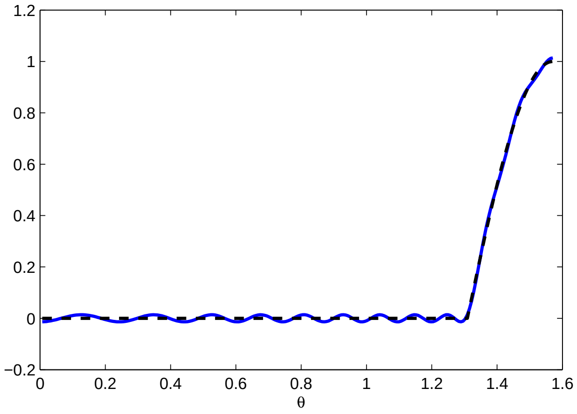


Fig. 5: Diagram of the antenna (continuous line) obtained by solving (3) under the specifications provided in Section 2.1.1. The target diagram is also shown (dashed line).

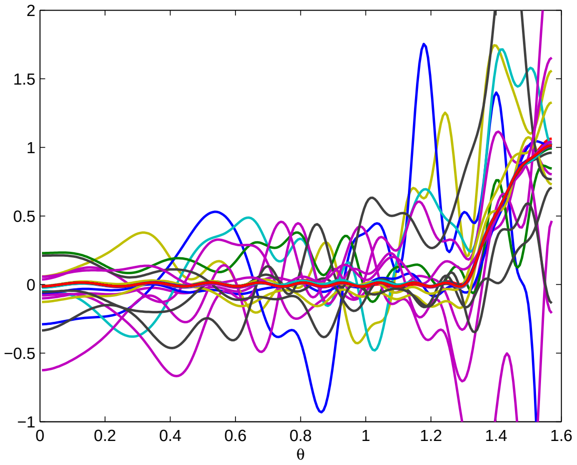


Fig. 6: Instances of the diagram $\tilde{D}_a(\theta)$ when the naive design is applied in the presence of actuation errors (500 instances).

$$\begin{aligned} &\min_{h,x_1,\dots,x_n} \ h \\ &\text{subject to: } \max_{\theta\in[0,\pi/2]} \left| \sum_{\ell=1}^{n} x_\ell(1+\delta_\ell^{(i)})D_\ell(\theta) - T(\theta) \right| \leq h, \ i=1,\dots,N \\ &\qquad\qquad\quad |x_\ell| \leq c, \ \ell = 1,\dots,n. \end{aligned} \tag{5}$$

Using the specifications in Section 2.1.1 and the $N = 500$ scenarios that were used in Section 2.1.3 for testing the robustness of the naive design, see Figure 6, we solved (5) and found the corresponding scenario-based design $x_1^*, \dots, x_{100}^*$. We obtained $h^* = 0.0144$. The idea is that we chose the design variables in a way that is robust over data, by minimising (w.r.t. $x = (x_1, \dots, x_n)$) the cost function

$$f(x, \delta) = \max_{\theta \in [0, \pi/2]} \left| \sum_{\ell=1}^{n} x_\ell (1 + \delta_\ell) D_\ell(\theta) - T(\theta) \right| \tag{6}$$

in the worst-case with respect to the 500 observed instances (scenarios) of the vector $(\delta_1, \dots, \delta_{100})$. Note that the interpretation of the optimal value h^* of h in the solution of the scenario program (5) is different from h^* in the solution of (3), where it was simply the value of the cost function (6) evaluated at the naive design variables with no actuation errors ($\delta = 0$). In the scenario program, instead, h^* is the *highest* value incurred by the cost functions $f(x^*, \delta^{(1)}), \dots, f(x^*, \delta^{(500)})$, where $x^* = (x_1^*, \dots, x_n^*)$ is the scenario design and $\delta^{(1)}, \dots, \delta^{(500)}$ are the observed uncertainty instances. It is appropriate to call the scenario-based h^* an *empirical* cost as it has been obtained from data. So, in short, we can say that h^* in the context of (5) is the empirical worst-case cost of the scenario solution.

Considering the way in which it was obtained, the h^* value delivered by the scenario program is certainly expected to be higher than the h^* value in the solution of the naive program (3). However, the former turns out to be only 4% higher than the latter, which was 0.0138. This small increase in the cost suggests that the price of robustness is low in this design problem.

The diagram of the antenna array obtained using the scenario approach and evaluated in the absence of actuation errors is in Figure 7.

Figure 8 confirms that the performance of the scenario solution is excellent simultaneously for all the observed scenarios that have been taken into account during optimisation. The striking difference between Figure 6 and Figure 8 can be made quantitative by comparing the values of $f(x^*, \delta^{(i)})$ for $i = 1, \dots, 500$ in the case where x^* is the naive solution to the case where x^* is the scenario solution. In the first case (naive design), the cost function $f(x^*, \delta)$ has a sample average value of 0.035 and a sample standard deviation of 0.16; in the second case (scenario design) the sample average is 0.0144 and the sample standard deviation is in the order of 10^{-6}.

However, since these latter values were computed using the same 500 scenarios that were used in optimising the decision, they might not be representative of what will happen when instances of the actuation errors that have not been measured so far occur. Therefore, it is crucial to evaluate the probability that the diagram of the designed antenna degrades and deviates from the target diagram more than the *empirical* worst-case cost $h^* = 0.0144$ for new realisations of the actuation uncertainty.

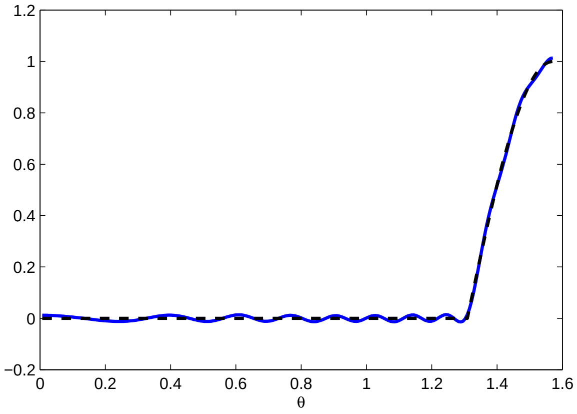


Fig. 7: Diagram of the scenario-based antenna array (solid line) $\sum_{\ell=1}^{n} x_\ell^* D_\ell(\theta)$ with the weights obtained by solving (5) with 500 scenarios, under the specifications provided in Section 2.1.1. The target diagram is also shown (dashed line).

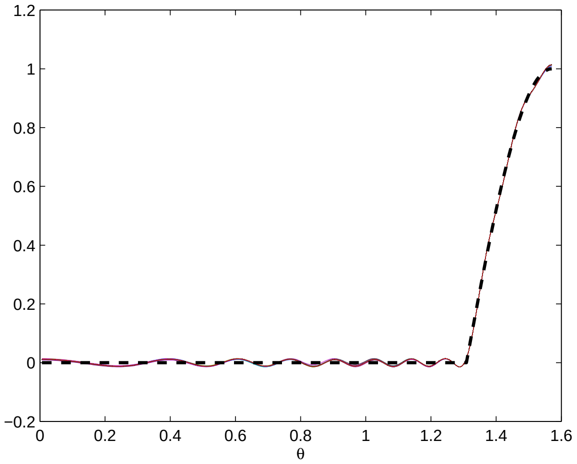


Fig. 8: Instances of the diagram of the scenario-based array when it is applied to the 500 scenarios that were used for computing the weights x_ℓ^* according to (5) and under the specifications provided in Section 2.1.1. The scenarios are the same that were used to test the naive design in Figure 6. Here, the 500 instances of the diagram are shown simultaneously by using thin lines, but are still very hardly distinguishable. The target diagram is also shown (dashed line).

The theory of the scenario approach allows us to assess rigorously the reliability of a scenario-based design without resorting to any other data besides those that have been considered during the optimisation step. The fundamental facts about this theory are recalled, and then applied to our scenario-solution, in the following section.

3.1 The Theory of the Scenario Approach

In this section we introduce the mathematical theory that allows us to evaluate the performance of the scenario-based design with respect to the infinite amount of unseen uncertainty instances, without availing of extra experiments or knowledge of the data-generating distribution. This theory is obtained by adapting to the present set-up the results from [13,14]. The theory works under the condition that the cost function $f(x,\delta)$ is convex in x for every given value of δ, and that the scenarios are independently sampled. Precisely, the essential working *assumption* for our purposes is that

> there is a probability measure $\mathbb{P}$ over the space of the uncertain actuation error vectors δ, and each vector δ is generated independently according to $\mathbb{P}$.

We remark that independence is postulated among sampled vectors $\delta^{(1)},\ldots,\delta^{(N)}$, but the components $\delta_1^{(i)},\ldots,\delta_n^{(i)}$ of a vector $\delta^{(i)}$ can be (even strongly) dependent: no assumption is made with respect to the distribution $\mathbb{P}$ according to which single scenarios are generated. Note also that,

> while the *existence* of $\mathbb{P}$ is postulated, no knowledge of $\mathbb{P}$ is required in what follows.

This is a crucial point for the general applicability of the scenario approach to data-driven designs, where assuming knowledge of $\mathbb{P}$ is usually unrealistic. We remark also that although in this paper the uncertain element δ is a random vector of size n, in general δ is allowed to be a generic random element, e.g., a process belonging to an infinite-dimensional space.

Denoting by

$$z_N^* = (h^*, x^*) = (h^*, x_1^*, \ldots, x_n^*)$$

the solution[2] to problem (5) with N scenarios, we can now define the *risk* of z_N^*, denoted by $R(z_N^*)$, as the probability that a new instance of the uncertain actuation errors will lead to an antenna array diagram that differs from the target diagram more than the empirical cost h^*, that is,

$$R(z_N^*) = \mathbb{P}\left\{\delta : f(x^*,\delta) > h^*\right\}. \tag{7}$$

In (7), $\mathbb{P}$ refers to the occurrence of δ only, so that the risk $R(z_N^*)$ is a function of z_N^*, which in turn depends on the scenarios $\delta^{(1)},\delta^{(2)},\ldots,\delta^{(N)}$. Hence, $R(z_N^*)$

[2] We assume that the solution is unique. Otherwise, a convex tie-break rule can be used, [14].

can be also interpreted as the *conditional probability* given $\delta^{(1)}, \delta^{(2)}, \dots, \delta^{(N)}$ that a new instance of the uncertain actuation error δ incurs a cost higher than h^*, that is, it holds that

$$R(z_N^*) = \mathbb{P}^{N+1}\{f(x^*, \delta) > h^* | \delta^{(1)}, \delta^{(2)}, \dots, \delta^{(N)}\},$$

where $\mathbb{P}^{N+1} = \mathbb{P} \times \mathbb{P} \times \cdots \times \mathbb{P}$ is the probability distribution of $(\delta^{(1)}, \delta^{(2)}, \dots, \delta^{(N)}, \delta)$, which is a product probability due to independence.

In a real problem, the value of $R(z_N^*)$ is unknown because the probability measure $\mathbb{P}$ is not available to the user, who has access only to a limited amount of samples from $\mathbb{P}$. However, as a function of the random scenarios $\delta^{(1)}, \dots, \delta^{(N)}$, the risk $R(z_N^*)$ is a random variable, and a fundamental theorem of the scenario approach states that the distribution of the random variable $R(z_N^*)$ is always *dominated* by a Beta probability distribution function whose parameters depend only on N and the number of design variables, which is n.[3] Precisely, it holds that

$$\mathbb{P}^N\{R(z_N^*) > \epsilon\} \leq \sum_{i=0}^{n} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}, \tag{8}$$

where the probability $\mathbb{P}^N$ is with respect to the N scenarios $\delta^{(1)}, \dots, \delta^{(N)}$.[4] It is a fact that the upper tail of the Beta distribution (i.e., the right-hand side of (8)) decays exponentially fast with N, so that the risk can be kept under control by choosing N large enough; for example, it can be proven that if

$$N \geq \frac{1}{\epsilon}\left(n + \ln\frac{1}{\beta} + \sqrt{2n \ln\frac{1}{\beta}}\right) \tag{9}$$

then

$$\mathbb{P}^N\{R(z_N^*) > \epsilon\} \leq \beta,$$

where $\epsilon \in (0,1)$ is an upper-bound to the risk, and $\beta \in (0,1)$ is a confidence parameter that can be easily made very small as it affects N only through logarithmic dependence (see [13,2]). In [13] it is further proved that the distribution of the probability of $R(z_N^*)$ is invariant, and indeed *completely determined* by N and n, for a class of problems that is called the class of *fully supported* problems. In order to characterise this important class of problems, we need the following definition.

[3] By "design variables" we mean "*real* design variables". In this paper, $x_1, \dots, x_n$ are real, therefore there are n (real) design variables. On the other hand, if the weights $x_1, \dots, x_n$ were allowed to take complex values, then, in order to apply the theoretical results in this section, problem (5) should be first reformulated as an optimization problem over $(\mathrm{Re}(x_1), \mathrm{Im}(x_1), \dots, \mathrm{Re}(x_n), \mathrm{Im}(x_n))$, that is, as a problem with $2n$ (real) design variables.

[4] This result holds true as a direct consequence of interpreting the *risk* of z_N^* as the *violation probability* of z_N^* according to the terminology of [14]. The interpretation of the *violation probability* as the *risk* that the cost h^* is exceeded is possible here because of the separation between design variables $(x_1, \dots, x_n)$ and the additional variable h in the min-max scenario program (5). See also [16] for more details.

Definition 1 (support scenario) A scenario in the program (5) is a *support scenario* if its removal changes the solution z_N^*. □

It is a fact that

the number of support scenarios in (5) is always bounded by $n+1$.

A problem is *fully supported* if, for every $N \geq n+1$, the number of support scenarios is $n+1$ with probability 1. In this case, the inequality in (8) is actually an equality, and there is no way to improve the upper-bound on the risk (at a given confidence level).

However, it is not rare that scenario programs with many variables are *not* fully supported, and way fewer support scenarios are found than there are optimization variables, see e.g. [37,36,31,15]. In these cases, if it were known a priori that the number of support scenarios is smaller than $n+1$, a smaller value than n could be used in formulae (8) and (9), and, as a consequence, a tighter upper-bound on the risk could be used. However, investigating the existence of tighter bounds on the number of support scenarios requires an ad-hoc analysis of the specific problem at hand. Moreover, even assuming that we can find a tighter bound, the actual number of support scenarios might still be smaller than the upper-bound, thus making our mathematical efforts practically useless. In [14], a new approach, called *wait-and-judge*, has been introduced, which allows the user to issue claims about the risk of the scenario solution that are made *a-posteriori*, based on the number of support scenarios that have been counted in the scenario program at hand.[5]

Concretely, the user is provided before any data is observed with *a set of statements* about the risk of the solution. These statements are of the kind "$R(z_N^*) \leq \epsilon_k$", for $k = 0, 1, \ldots, n+1$, where $\epsilon_0, \ldots, \epsilon_{n+1}$ is a sequence of increasing numbers between 0 and 1. After data have been used to compute the scenario solution z_N^* and the number of support scenarios s_N^* has been counted, the user issues the statement

"$R(z_N^*) \leq \epsilon_{s_N^*}$",

which depends on s_N^*, the number of support scenarios. The theory in [14] explains how to choose $\epsilon_0, \ldots, \epsilon_{n+1}$ in such a way that the overall probability that the user observes N scenarios, computes z_N^*, s_N^* and issues a false claim is practically negligible. In short, the theory explains how to issue reliable data-dependent claims.

A valid rule to select $\epsilon_0, \ldots, \epsilon_{n+1}$ is provided by Theorem 1 below. The theorem holds true under two technical conditions, which will be made precise and discussed for the antenna array design problem in Section 3.3. As we shall see, the evaluations of the risk that are provided by Theorem 1 turn out to be very tight in spite of the distribution-free nature (the value of the bound does not depend on $\mathbb{P}$) of the theorem.

[5] Computing the number of support scenarios requires removing one by one the active constraints and verifying whether the solution changes, an operation that can be carried out at reasonably low computational cost.

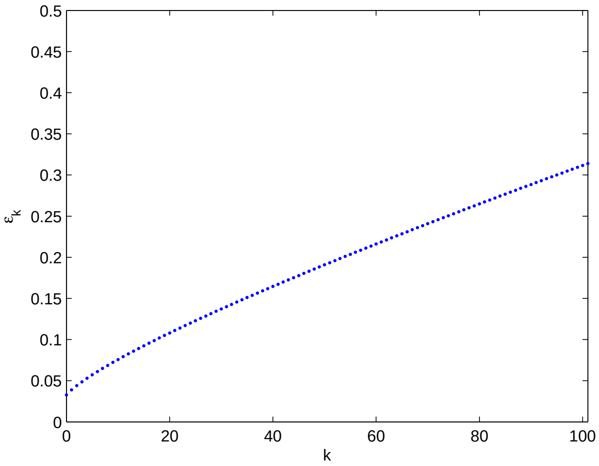


Fig. 9: $\epsilon_0, \dots, \epsilon_{101}$ when $N = 500$, $n = 100$ and for $\beta = 10^{-6}$

Theorem 1 **([14])** *Given $\beta \in (0,1)$, for any $k = 0, 1, \dots, n+1$, let ϵ_k be the unique solution in $(0,1)$ of the polynomial equation in the v variable*

$$\frac{\beta}{N+1} \sum_{m=k}^{N} \binom{m}{k} (1-v)^{m-k} - \binom{N}{k} (1-v)^{N-k} = 0. \tag{10}$$

It holds that

$$\mathbb{P}^N \{ R(z_N^*) > \epsilon_{s_N^*} \} \le \beta.$$

Figure 9 shows the values of $\epsilon_0, \dots, \epsilon_{101}$ computed for $\beta = 10^{-6}$, $N = 500$ and $n = 100$ according to Theorem 1.

If the values in Figure 9 are used as data-dependent bounds on the risk, the probability of issuing a false post-optimisation claim on the risk of the solution is no bigger than 10^{-6}. In other words, if many designs are performed using the scenario approach on independently sampled scenarios, and the corresponding post-optimisation claims are issued by selecting the s_N^*-th ϵ-value from Figure 9, only one claim out of 10^6 issued claims is expected to be wrong (at most).

In concluding, the standard theory of the scenario approach provides a way to compute an upper-bound on the risk that is tight (not improvable) for the class of fully-supported problems, but that can be conservative for general problems. In fact, in order to guarantee that the risk is below a certain threshold, one has to collect a number of scenarios that depends on the number of decision variables and is often larger than necessary. This is penalising especially when scenarios represent a costly and limited resource. The "wait-and-judge" approach, on the other hand, allows one to assess the risk of a data-based solution in a data-dependent way, and the message is that the number of support scenarios among the observed scenarios is a solid indicator of the risk, no matter how many decision variables are used in the problem.

3.2 Application to the Antenna Array Scenario-Based Design

We are now ready to examine our scenario-based design in the light of the "wait-and-judge" theory. We consider the solution to (5) and we count how many scenarios among the 500 that were used to compute the solution are support scenarios. The number of support scenarios, s_N^*, turns out to be equal to 17. We retrieve the value of the corresponding ϵ_{17} in Figure 9, and we get $\epsilon_{17} = 0.099$. Therefore, we can claim that

$$R(z_N^*) \leq 9.9\%,$$

that is, we can be practically sure that for more than the 90% of the possible actuation errors the diagram of our scenario-based antenna array will not differ more than $h^* = 0.0144$ from the desired, target diagram.

3.2.1 Validation of the bound on the risk

In principle, the risk of a scenario-based solution can always be estimated by Monte Carlo methods, by resorting to a (very) large number of new realisations of the actuation errors. However, unless a lot of data are available (e.g., a reliable model of the uncertainty is available and simulations can be run based on it in a reasonable amount of time), this is impractical. When data are expensive measurements, they should be used to robustify the decision rather than used for estimating the risk: this is the reason why data-dependent bounds computed according to the "wait-and-judge" approach are so important.

In this paper, contrary to what happens in real world situations, we can use the same simulation set-up that was used to compute the numerical solutions above (see Section 2.1.3) in order to estimate the true risk of our data-based design and validate the upper-bound of 9.9%. Hence, we simulated 10^6 new instances of the actuation error vector δ and we evaluated the cost function $f(x^*, \delta)$ for all of them. The empirical cost threshold $h^* = 0.0144$ was exceeded in the 4.3% of the cases, so that the computed threshold of 9.9% was satisfied, as expected from the application of Theorem 1. On average, the value of $f(x^*, \delta)$ was 0.0145 with a sample standard deviation of 0.0013. The gap between the actual value of the risk (4.3%) and the estimated value (9.9%) is motivated by the stochastic fluctuation of the risk and the fact that the bound was required to be valid with very high confidence $1 - 10^{-6}$.

3.2.2 Experiments for different values of N

So far, we have considered the scenario program (5) with $N = 500$ scenarios. Here we report the results of other simulations for other values of N.

Table 1: Results of the wait-and-judge approach for different N. The first row ($N = 0$) refers to the nominal program (3).

N	h^*	s_N^*	$\epsilon_{s_N^*}$	$R(z_N^*)$	μ_f	σ_f
0	0.0138	N.A.	N.A.	100%	0.0519	0.3098
100	0.0140	9	32.2%	10.2%	0.0364	0.1876
250	0.0142	15	14.0%	5.9%	0.0144	0.0019
500	0.0144	17	9.9%	4.3%	0.0145	0.0013
1000	0.0148	28	6.7%	2.7%	0.0147	$8 \cdot 10^{-4}$
2500	0.0153	42	3.5%	1.6%	0.0148	$6 \cdot 10^{-4}$

In particular, we solved an instance of the scenario program (5) for $N = 100, 250, 1000, 2500$. In Table 1, we give for each case the values of the worst-case empirical cost h^*, the number of support constraints s_N^*, and the guaranteed theoretical bound on the risk $\epsilon_{s_N^*}$, which is computed using (1) for the corresponding value of N and $\beta = 10^{-6}$. Finally, by using a Monte Carlo sample of new 10^6 instances of the actuation errors δ, we estimated the true risk $R(z_N^*)$ of exceeding h^*. We computed also the sample average value of $f(x^*, \delta)$ (denoted by μ_f) and its sample average standard deviation (denoted by σ_f). These three values, $R(z_N^*), \mu_f, \sigma_f$, which were obtained by using the additional Monte Carlo samples, are reported in the right-hand side of the table. The value of μ_f initially decreases when N is increased because with a small N the value of $f(x^*, \delta)$ is subject to a large uncertainty and its erratic nature inflates its mean; later, for larger values of N, μ_f slightly increases, while the variability as described by σ_f becomes progressively smaller. We remark again that while the left side of the table was computed by resorting only to the observed scenarios that were used in optimising the decision, the right side of the table was obtained by using the knowledge of the distribution of the uncertain variable δ in order to simulate a large amount of new validation instances. Therefore, the right side is *not available* in real world situations.

The results reported in each row of Table 1 characterise the outcome obtained by applying the wait-and-judge approach to a single set of data $\delta^{(1)}, \ldots, \delta^{(N)}$. The next Table 2 shows statistics that are obtained by applying the wait-and-judge approach to 3000 different sets of randomly sampled data $\delta^{(1)}, \ldots, \delta^{(N)}$. In particular, the table shows the minimum, the maximum and the average number of support scenarios over the 3000 runs (denoted respectively by $\min_{\{s_N^*\}}$, $\max_{\{s_N^*\}}$ and $\mu_{\{s_N^*\}}$); the sample average of the upper-bound $\epsilon_{s_N^*}$ (denoted by $\mu_{\{\epsilon_{s_N^*}\}}$) and its sample standard deviation (denoted by $\sigma_{\{\epsilon_{s_N^*}\}}$); finally, in the last two columns we have reported results related to the difference between the upper-bound $\epsilon_{s_N^*}$ and the true risk $R(z_N^*)$: the penultimate column gives the sample average ($\mu_{\{\epsilon_{s_N^*} - R(z_N^*)\}}$) and the last column gives the standard deviation ($\sigma_{\{\epsilon_{s_N^*} - R(z_N^*)\}}$). We also remark that in all the experiments we had $\epsilon_{s_N^*} > R(z_N^*)$ as is expected because the probability of

Table 2: Results of the wait-and-judge approach from 3000 samples of data (for $N = 250, 500, 1000$)

N	$\min_{\{s^*_N\}}$	$\max_{\{s^*_N\}}$	$\mu_{\{s^*_N\}}$	$\mu_{\{\epsilon_{s^*_N}\}}$	$\sigma_{\{\epsilon_{s^*_N}\}}$	$\mu_{\{\epsilon_{s^*_N} - R(z^*_N)\}}$	$\sigma_{\{\epsilon_{s^*_N} - R(z^*_N)\}}$
250	9	23	16	18.2%	1.2%	12.1%	1.9%
500	14	30	21	11.2%	0.7%	6.9%	1.1%
1000	20	39	29	6.8%	0.4%	3.9%	0.7%

underestimating the true risk was set to a very small value 10^{-6}. This choice explains also the presence of an average “safety gap” between $\epsilon_{s^*_N}$ and $R(z^*_N)$.

3.3 Technical assumptions for the validity of Theorem 1

Theorem 1 holds true under two technical assumptions. The first is the existence and the uniqueness of the solution z^*_N, with probability 1 with respect to the sample $\delta^{(i)}$, $i = 1, \ldots, N$, for all N. In the present context of antenna array design, existence is ensured by the fact that the scenario program (5) is a min-max problem with closed optimisation domain, while uniqueness can always be ensured by introducing a convex tie-break rule, e.g., by choosing the solution that minimises the sum $\sum_{\ell=1}^{n} x_\ell^2$. The second assumption under which the theorem is stated in [14] is a non-degeneracy condition: for every N, with probability 1 with respect to the sample $\delta^{(i)}$, $i = 1, \ldots, N$, the solution to program (5) with all the constraints in place coincides with the solution to the program where only the support constraints (which are the constraints in (5) that correspond to the scenarios $\delta^{(i)}$ that are support scenarios) are kept. In the present context of antenna array design, this corresponds to requiring that all the active constraints in (5) be support constraints. However, we here anticipate that more recent results have shown that this condition can be dropped and that the statement of Theorem 1 holds true in general, provided that s^*_N is redefined as the number of *active constraints*.

4 Conclusions

In this paper, we have shown how the recently developed “wait-and-judge” approach for scenario optimisation can be applied to scenario-based min-max design problems in order to issue certificates about the cost incurred by the output of the designing process. The approach has been illustrated and validated on an antenna array design problem.

References

1. Alamo, T., Tempo, R., Luque, A., Ramirez, D.R.: Randomized methods for design of uncertain systems: sample complexity and sequential algorithms. Automatica **51**, 160–172 (2015)

2. Alamo, T., Tempo, R., Luque, A., Ramirez, D.R.: Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms. Automatica **52**, 160–172 (2015)
3. Ben-Tal, A., El Ghaoui, L., Nemirovski, A.: Robust optimization. Princeton University Press (2009)
4. Ben-Tal, A., Nemirovski, A.: Robust convex optimization. Mathematics of Operations Research **23**(4), 769–805 (1998)
5. Ben-Tal, A., Nemirovski, A.: Lectures on modern convex optimization. SIAM (2001)
6. Ben-Tal, A., Nemirovski, A.: Robust optimization–methodology and applications. Mathematical Programming **92**(3), 453–480 (2002)
7. Bertsimas, D., Gupta, V., Kallus, N.: Data-driven robust optimization. Mathematical Programming (2017). DOI 10.1007/s10107-017-1125-8
8. Bertsimas, D., Sim, M.: The price of robustness. Operations Research **52**(1), 35–53 (2004). DOI 10.1287/opre.1030.0065
9. Calafiore, G., Campi, M.C.: The scenario approach to robust control design. IEEE Transactions on Automatic Control **51**(5), 742–753 (2006)
10. Calafiore, G.C., Campi, M.C.: 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
11. Campi, M.C., Carè, A.: Random convex programs with L1-regularization: sparsity and generalization. SIAM Journal on Control and Optimization **51**(5), 3532–3557 (2013)
12. Campi, M.C., Garatti, S., Prandini, M.: The scenario approach for systems and control design. Annual Reviews in Control **33**(2), 149–157 (2009). DOI 10.1016/j.arcontrol.2009.07.001
13. Campi, M.C., 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
14. Campi, M.C., Garatti, S.: Wait-and-judge scenario optimization. Mathematical Programming **167**(1), 155–189 (2018). DOI 10.1007/s10107-016-1056-9
15. Carè, A., Garatti, S., Campi, M.C.: FAST - Fast Algorithm for the Scenario Technique. Operations Research **62**(3), 662–671 (2014)
16. Carè, A., Garatti, S., Campi, M.C.: Scenario min-max optimization and the risk of empirical costs. SIAM Journal on Optimization **25**(4), 2061–2080 (2015)
17. Charnes, A., Cooper, W.: Chance constrained programming. Management Science **6**(1), 73–79 (1959)
18. Dantzig, G.: Linear programming under uncertainty. Management Science **1**(3-4), 197–206 (1955)
19. Dentcheva, D.: Optimization models with probabilistic constraints. In: G. Calafiore, F. Dabbene (eds.) Probabilistic and Randomized Methods for Design under Uncertainty. Springer-Verlag, London, UK (2006)
20. El Ghaoui, L., Lebret, H.: Robust solutions to uncertain semidefinite programs. SIAM Journal on Optimization **9**(1), 33–52 (1998)
21. Hanasusanto, G.A., Roitch, V., Kuhn, D., Wiesemann, W.: A distributionally robust perspective on uncertainty quantification and chance constrained programming. Mathematical Programming **151**(1), 35–62 (2015). DOI 10.1007/s10107-015-0896-z
22. Hanasusanto, G.A., Roitch, V., Kuhn, D., Wiesemann, W.: Ambiguous joint chance constraints under mean and dispersion information. Operations Research **65**(3), 751–767 (2017). DOI 10.1287/opre.2016.1583
23. IBM-ILOG: V12.4: Users manual for CPLEX (2012)
24. Margellos, K., Goulart, P., Lygeros, J.: On the road between robust optimization and the scenario approach for chance constrained optimization problems. IEEE Transactions on Automatic Control **59**(8), 2258–2263 (2014)
25. MATLAB: version 8.1.0.604 (R2013a). The MathWorks Inc., Natick, Massachusetts (2013)
26. Matyjas, J.D., Kumar, S., Hu, F.: Wireless network performance enhancement via directional antennas: models, protocols, and systems. CRC Press (2015)
27. 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)

28. 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)
29. Petersen, I.R., Tempo, R.: Robust control of uncertain systems: classical results and recent developments. Automatica **50**, 1315–1335 (2014)
30. Prékopa, A.: Probabilistic programming. In: A. Ruszczyński, A. Shapiro (eds.) Stochastic Programming, *Handbooks in Operations Research and Management Science*, vol. 10. Elsevier, London, New York and Amsterdam (2003)
31. Schildbach, G., Fagiano, L., Morari, M.: Randomized solutions to convex programs with multiple chance constraints. SIAM Journal on Optimization **23**(4), 2479–2501 (2013)
32. Shapiro, A., Dentcheva, D., Ruszczyński, A.: Lectures on stochastic programming: modeling and theory. MPS-SIAM, Philadelphia, Pennsylvania, USA (2009). URL http://www2.isye.gatech.edu/people/faculty/Alex_Shapiro/SPbook.pdf
33. Soyster, A.: Convex programming with set-inclusive constraints and applications to inexact linear programming. Operations Research **21**(5), 1154–1157 (1973)
34. Tempo, R., Calafiore, G., Dabbene, F.: Randomized Algorithms for Analysis and Control of Uncertain Systems – 2nd Edition. Springer, London, UK (2013)
35. Vayanos, P., Kuhn, D., Rustem, B.: A constraint sampling approach for multistage robust optimization. Automatica **48**(3), 459–471 (2012)
36. Welsh, J., Kong, H.: Robust experiment design through randomisation with chance constraints. In: Proceedings of the 18th IFAC World Congress. Milan, Italy (2011)
37. 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)
38. 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)