Proceedings of the 18th World Congress
The International Federation of Automatic Control
Milano (Italy) August 28 - September 2, 2011

FAST: an algorithm for the scenario approach with reduced sample complexity

A. Carè [*] **S. Garatti** [**] **M.C. Campi** [*]

[*] *Dipartimento di Ingegneria dell'Informazione - Università di Brescia, via Branze 38, 25123 Brescia, Italia. (E-mail:* algo.care@ing.unibs.it, marco.campi@ing.unibs.it*; web-site:* http://bsing.ing.unibs.it/∼algo.care/, http://bsing.ing.unibs.it/∼campi/*)*

[**] *Dipartimento di Elettronica e Informazione - Politecnico di Milano, piazza Leonardo da Vinci 32, 20133 Milano, Italia. (E-mail:* sgaratti@elet.polimi.it*, web-site:* http://home.dei.polimi.it/sgaratti/*)*

Abstract: In previous contributions, Calafiore and Campi (2006) and Campi et al. (2009b), it has been shown that the scenario approach is a handy methodology to design control systems having robustness properties that are otherwise difficult to design along more standard methods. On the other hand, it has also been noted that the sample complexity of the scenario approach rapidly increases with the number of optimization variables and this may pose a hurdle to its applicability to medium and large scale problems. We here introduce FAST (Fast Algorithm for the Scenario Technology), a variant of the scenario optimization algorithm with reduced sample complexity. The price one pays with FAST is that the obtained solution may be suboptimal. However, practical evidence shows that this suboptimality is often negligible and thus FAST offers a practical alternative to the standard scenario approach.

Keywords: scenario approach, control system design, convex optimization, randomized method, sample complexity.

1. INTRODUCTION

Many problems in systems and control with uncertain knowledge of the environment can be formulated as *uncertain convex-optimization programs*, that is programs in which the constraint is imprecisely known:

$$\begin{aligned} \text{UP} : &\min_x c^T x \\ &\text{subject to: } x \in \mathcal{X}_\delta, \quad \mathcal{X}_\delta \subseteq \mathbb{R}^d. \end{aligned} \tag{1}$$

In (1), x is the design parameter, δ is the uncertain parameter belonging to an uncertainty domain Δ, and $\mathcal{X}_\delta$ is a convex and closed set for every $\delta \in \Delta$. Linearity in the objective function is without loss of generality since any problem with generic convex objective can be rewritten as in (1), see Boyd and Vandenberghe (2004). A particular instance of (1), which often arises in systems and control problems, is the following (Calafiore and Campi (2006) and Campi et al. (2009b)):

$$\begin{aligned} &\min_{\gamma,h} h \\ &\text{subject to: } \ell(\gamma,\delta) \leq h, \end{aligned}$$

where γ is e.g. the controller parameter, δ the unknown plant parameter, and $\ell(\gamma,\delta)$ a cost measuring the performance of γ when applied to δ. h upper-bounds the performance and the optimization problem is such that γ is chosen so as to squeeze h down as much as possible. Uncertain programs UP cannot be seen as a complete formalization of the design problem, since there is no indication on how to deal with the uncertainty. Different choices are possible. A *robust* solution to (1) is a solution that satisfies all the constraints obtained for all possible values of δ, i.e. it belongs to the set $\bigcap_{\delta\in\Delta} \mathcal{X}_\delta$, see e.g. Ben-Tal et al. (2010). Alternatively, one can view Δ as a probability space endowed with a σ-algebra $\mathcal{D}$ and a probability measure $\mathbb{P}$ (i.e. δ is a random element) and be interested in finding a chance-constrained solution satisfying the constraint $x \in \mathcal{X}_\delta$ with "high probability", see Prékopa (1995, 2003), Vajda (1972). Both robust and chance-constrained optimization are notoriously hard to solve in general, even though there are notable exceptions where the solution can actually be computed, see Ben-Tal and Nemirovski (1998, 1999), El Ghaoui and Lebret (1998), El Ghaoui and Niculescu (2000), Prékopa (1995, 2003), Henrion and Strugarek (2008).

The scenario optimization approach, introduced in Calafiore and Campi (2005, 2006), is an innovative technology to find chance-constrained solutions to problem (1), and has been successfully applied to a number of system and control problems in Calafiore and Campi (2006) and Campi et al. (2009b). Along this approach, a *finite* number N of uncertainty instances $\delta^{(1)}, \ldots, \delta^{(N)}$ are randomly sampled according to $\mathbb{P}$ independently one of another[1], and only the constraints $x \in \mathcal{X}_{\delta^{(i)}}$ are enforced, corresponding to the *sample program*:

[1] See Calafiore et al. (2000), Calafiore and Dabbene (2002), and Tempo et al. (2005) for algorithms to perform random sampling.

978-3-902661-93-7/11/$20.00 © 2011 IFAC

10.3182/20110828-6-IT-1002.02457

$$\mathrm{SP}_N : \min_x c^T x$$
$$\text{subject to: } x \in \bigcap_{i\in\{1,\ldots,N\}} \mathcal{X}_{\delta^{(i)}}. \qquad (2)$$

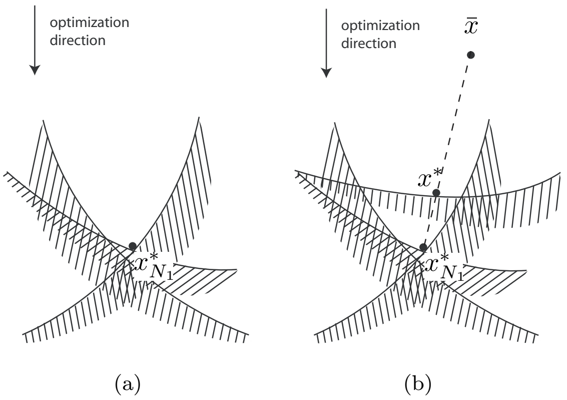
SP_N is a standard convex optimization program with a *finite* number of constraints and its solution x_N^* can be obtained by resorting to standard optimization tools. Moreover, as shown in Calafiore and Campi (2005, 2006) and Campi and Garatti (2008), x_N^* satisfies the vast majority of the constraints in Δ, even those that have not been sampled and that therefore have had no role in computing x_N^*, provided that N is suitably chosen. More precisely, the scenario solution x_N^* satisfies constraints with high probability $1-\epsilon$, that is, it is chance-constrained feasible at level ϵ.

The issue of evaluating the sample size N of sample programs is of great importance, as discussed in many contributions, Calafiore and Campi (2005, 2006), Campi and Garatti (2008), Alamo et al. (2009). In the context of (2) the fundamental result has been established in Campi and Garatti (2008), where a necessary and sufficient condition on N for x_N^* to be feasible at level ϵ was given. It turns out that the required N is inversely proportional to ϵ and is proportional to d, the number of optimization variables, i.e. N scales as $\frac{1}{\epsilon}\cdot d$, see Calafiore (2009) and Alamo et al. (2010) which provide explicit expression for N. In Nemirovski and Shapiro (2006) and Oishi (2007), it has been observed that the dependence of N on $\frac{d}{\epsilon}$ may result in too many constraints to sample for large scale problems having large d, thus posing a difficulty for the practical use of the method. The present paper proposes a new sample-based algorithm called FAST (Fast Algorithm for the Scenario Technology) to overcome this difficulty. The main advantage of FAST is that the dependence of N on $\frac{1}{\epsilon}$ and d becomes additive, i.e. N scales as $\frac{1}{\epsilon}+d$. This significantly improves the applicability of the scenario approach.

1.1 New Idea behind FAST

The idea behind FAST is as follows. Suppose that one a priori knows a point $\bar{x}$ that is robustly feasible, i.e. $\bar{x} \in \bigcap_{\delta\in\Delta}\mathcal{X}_\delta$. This assumption is verified in many situations of interest [2]. It is perhaps worth stating explicitly that *there are no requirements on $\bar{x}$ other than it is robustly feasible, in particular there are no requirements on the performance value $c^T\bar{x}$.*

Based on $\bar{x}$, FAST constructs a quasi-optimal solution in two steps. First, a moderate number N_1 of constraints are sampled and the optimal solution $x_{N_1}^*$ which satisfies these N_1 constraints is determined, see Fig. 1(a). This first step is accomplished at low computational effort due to the moderate number of constraints involved; on the other hand, $x_{N_1}^*$ is not guaranteed to meet the desired violation level ϵ since N_1 is too low. Then, a *detuning* step is started: N_2 additional constraints are sampled and $x_{N_1}^*$ is updated by moving it along the line segment connecting $x_{N_1}^*$ to $\bar{x}$ until the updated solution x^* satisfies all the N_2 newly sampled constraints, see Fig. 1(b).



(a) (b)

Figure 1. Illustration of FAST.

In this construction, N_1 and N_2 scale as d and $\frac{1}{\epsilon}$ respectively, leading to an overall number of constraints $N = N_1+N_2$ which is typically much smaller than that required by the "classical" scenario approach. Moreover, choosing a small ϵ does not affect N_1 and only results in a large N_2 value which corresponds to having many constraints in the detuning step, which is a *one-dimensional* program and can therefore be efficiently solved even for large values of N_2.

1.2 Organization of the paper

The remainder of the paper is organized as follows. Section 2 provides some mathematical background on the "classical" scenario approach which is required to correctly formalize the FAST algorithm. Such a formalization and a discussion on the properties of FAST are given in Section 3. The proofs are provided in Section 4, while an illustrative simulation example is given in Section 5.

2. BACKGROUND MATERIAL ON THE SCENARIO APPROACH

Throughout, we make the following assumption.

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

$$\min_x c^T x$$
$$\text{subject to: } x \in \bigcap_{\delta\in F\subseteq\Delta} \mathcal{X}_\delta, \qquad (3)$$

is feasible, and its feasibility domain has a nonempty interior. Moreover, the solution of (3) exists and is unique. *

Although this assumption can be relaxed, see Campi and Garatti (2008), it is here made to avoid technical complications that have little conceptual importance.

The following notion of violation probability is fundamental.

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

[2] In e.g. robust feedback controller synthesis, as in Campi et al. (2009b), one can take $\bar{x}$ corresponding to zero control. Similarly, a suitable $\bar{x}$ can be easily determined in applications as IPMs (Interval Predictor Models), see Campi et al. (2009a), and robust Chebyshev FIR equalization, see Mutapcic et al. (2007). In other more general contexts, one can resort to sequential randomized algorithms, see e.g. Polyak and Tempo (2001), Fujisaki et al. (2003), Oishi (2007).

One issue of major interest is whether the violation $V(x_N^*)$ of the scenario solution x_N^* of problem (2) is below a user-chosen level ϵ. Note that $V(x_N^*)$ is a random variable since x_N^* depends on the random extractions $\delta^{(1)}, \dots, \delta^{(N)}$, so that the statement $V(x_N^*) \le \epsilon$ has a probabilistic nature, i.e. $V(x_N^*) \le \epsilon$ holds with a certain probability. An exact quantification of the probability of "bad" extractions of $\delta^{(1)}, \dots, \delta^{(N)}$ such that $V(x_N^*) > \epsilon$ is given in Campi and Garatti (2008), where the following relation is proved:

$$\mathbb{P}^N\{V(x_N^*) > \epsilon\} \le \sum_{i=0}^{d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}, \qquad (4)$$

where $\mathbb{P}^N$ indicates the product probability measure on Δ^N according to which multi-samples $(\delta^{(1)}, \dots, \delta^{(N)})$ are extracted. This quantification is "exact" in that the inequality $\le$ in (4) is an equality $=$ for a whole class of problems - the so-called *fully-supported problems*, see Definition 3 in Campi and Garatti (2008) - so that the result in (4) cannot be improved. The right hand side of (4) is the so-called incomplete Beta function ratio, see e.g. Gupta and Nadarajah (2004). For brevity, in the sequel we will use the notation

$$B_\epsilon^{N,d} = \sum_{i=0}^{d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}. \qquad (5)$$

One can fix an arbitrarily small confidence parameter β and find the smallest integer N such that $B_\epsilon^{N,d} \le \beta$. Due to (4), this N entails $\mathbb{P}^N\{V(x_N^*) > \epsilon\} \le \beta$, that is solving SP_N returns a chance-constrained feasible solution at level ϵ with high confidence $1-\beta$.

In practice, one can find the number of constraints N by numerically solving the inequality $B_\epsilon^{N,d} \le \beta$, e.g. by means of the bisection method. To this end, in e.g. MATLAB$^{\mathrm{TM}}$ the value of $B_\epsilon^{N,d}$ can be efficiently computed by calling the betainc function as follows: betainc$(1-\epsilon, N-d+1, d)$. In alternative, one can use an explicit formula for N, at the price of introducing some conservatism: in Calafiore (2009) it is shown that $B_\epsilon^{N,d} \le \beta$ provided that

$$N \ge \frac{2}{\epsilon}\left(d + \ln\frac{1}{\beta}\right). \qquad (6)$$

This formula reveals that N bears a logarithmic dependence on β so that β can be made very small, say 10^{-9}, without increasing N too much. That is, confidence has little practical importance. Once β is fixed, the dependence on d and ϵ given by (6) is linear in $\frac{d}{\epsilon}$; indeed this is provably the correct dependence for relation $B_\epsilon^{N,d} \le \beta$ to hold.

3. FAST

The FAST algorithm is given first, followed by a discussion.

3.1 FAST algorithm

- INPUT:
  - $\epsilon \in (0,1)$, violation parameter;
  - $\beta \in (0,1)$, confidence parameter;
  - $\bar{x} \in \bigcap_{\delta\in\Delta} \mathcal{X}_\delta$, robustly feasible point;
  - N_1, an integer $\ge d$.

(1) Compute the smallest integer N_2 such that

$$N_2 \ge \frac{\ln\beta - \ln B_\epsilon^{N_1,d}}{\ln(1-\epsilon)}, \qquad (7)$$

where $B_\epsilon^{N_1,d}$ is as in (5).

(2) Sample N_1+N_2 independent constraints $\delta^{(1)}, \dots, \delta^{(N_1)}, \delta^{(N_1+1)}, \dots, \delta^{(N_1+N_2)}$, according to $\mathbb{P}$.

(3) Solve the SP_{N_1} problem in (2) with $N = N_1$; let $x_{N_1}^*$ be the optimal solution.

(4) Let $\hat{x}[\alpha] := (1-\alpha)x_{N_1}^* + \alpha\bar{x}$, $\alpha \in [0,1]$, i.e. $\hat{x}[\alpha]$ describes the line segment connecting $x_{N_1}^*$ with $\bar{x}$.
Solve the following problem

$$\mathrm{DETUNING}_{N_2}: \quad \min_{\alpha\in[0,1]} c^T\hat{x}[\alpha] \quad \text{subject to: } \hat{x}[\alpha] \in \bigcap_{i=N_1+1}^{N_1+N_2} \mathcal{X}_{\delta^{(i)}}; \qquad (8)$$

let α^* be the optimal solution.

- OUTPUT:
  - $x^* := \hat{x}[\alpha^*]$.

The following theorem states that x^* satisfies all constraints in Δ but an ϵ-fraction at most, with probability no smaller than $1-\beta$. That is, x^* is a chance-constrained ϵ-feasible solution with high confidence $1-\beta$.

Theorem 3. Under the current assumptions, it holds that

$$\mathbb{P}^{N_1+N_2}\{V(x^*) > \epsilon\} \le \beta.$$

*

The proof of the Theorem 3 is given in Section 4.

3.2 Discussion

In FAST, the user first solves SP_{N_1} with N_1 constraints and then computes N_2 through (7). N_1 is decided by the user, while N_2 depends on N_1, ϵ, and β. In this section, we discuss how to choose N_1, how to compute N_2, and provide arguments on why FAST is a sensible optimization algorithm.

Selection of N_1

N_1 should be chosen so as to achieve a fast computation of $x_{N_1}^*$: if N_1 is too large, finding $x_{N_1}^*$ for medium or large dimensional problems can be expensive and one loses the advantages of using FAST. On the other hand, how large N_1 is impacts on the quality of the solution $x_{N_1}^*$, and if N_1 is too small, the final solution x^* resulting from the detuning phase may exhibit a poor objective value. As a rule of thumb out of empirical experience, we suggest to take $N_1 = 20d$.

Computing N_2

N_2 is a function of N_1, ϵ, and β. In many practical cases, however, one can just use, in place of N_2, the smallest integer $\widetilde{N_2}$ satisfying

$$\widetilde{N_2} \ge \frac{\ln\beta}{\ln(1-\epsilon)}, \qquad (9)$$

obtained from (7) by neglecting the term $-\frac{\ln B_\epsilon^{N_1,d}}{\ln(1-\epsilon)}$ which is ≥ 0 so that $\widetilde{N_2} \ge N_2$. (9) can be further simplified to

$$\widetilde{N_2} \ge \frac{1}{\epsilon}\ln\frac{1}{\beta}, \qquad (10)$$

obtained from (9) using relation $-\ln(1-\epsilon) \geq \epsilon$. The practical value of expressions (9) and (10) relies on that the contribution of $B_\epsilon^{N_1,d}$ in (7) is negligible when N_1 is smaller than $\frac{d}{\epsilon}$. Thus, the first N_1 constraints primarily serve the purpose of obtaining a good initial solution $x^*_{N_1}$, while the other N_2 are responsible for securing the feasibility guarantees of the solution.

Advantages with FAST

- **Reduced sample size requirements**
  The FAST algorithm provides a cheaper way to find solutions to medium and large dimensional problems than the classical scenario approach. Indeed, one can choose $N_1 = Kd$, where K is a user-selected number normally set to 20, while N_2 can be approximated by $\frac{1}{\epsilon}\ln\frac{1}{\beta}$, as already seen. Hence, a handy formula to estimate the *overall number of constraints needed* with FAST is
  $$Kd + \frac{1}{\epsilon}\ln\frac{1}{\beta}.$$
  A comparison with
  $$\frac{2d}{\epsilon} + \frac{2}{\epsilon}\ln\frac{1}{\beta},$$
  the number of constraints needed with the classical scenario approach, see (6), shows the key point that, with FAST, the critical multiplicative dependence on $\frac{1}{\epsilon}\cdot d$ is replaced by an additive dependence on $\frac{1}{\epsilon}$ and d.
- **Possibility to reduce ϵ to small values**
  The increasing of the number N_2 of constraints to be used in the DETUNING$_{N_2}$ problem yields a moderate increasing of the computational complexity, because DETUNING$_{N_2}$ is a 1-decision-variable problem which can e.g. be solved by means of bisection. Thus, N_2 can be large and, correspondingly, ϵ can be reduced to values much smaller than with the classical scenario approach.

Suboptimality

The solution obtained through the classical scenario approach is superoptimal when compared to the optimal *robust* solution, the solution that satisfies $x \in \bigcap_{\delta\in\Delta}\mathcal{X}_\delta$, indeed the former is less constrained than the latter. This property does not carry over to the solution of FAST, as illustrated in Fig. 2. Nevertheless, empirical evidence shows that in many situations [3] the constraints tend to concentrate and cluster in a way that, by only slightly moving away from $x^*_{N_1}$ in the direction of $\bar{x}$, one soon "emerges" above the newly sampled N_2 constraints. Consequently, x^* is only little away from $x^*_{N_1}$ and the suboptimality is minor, often negligible. Moreover, the algorithm provides us with a handy instrument to confirm in hindsight whether this is indeed the case: the difference $c^T x^* - c^T x^*_{N_1}$ is an upper bound to the possible suboptimality with respect to the robust solution.

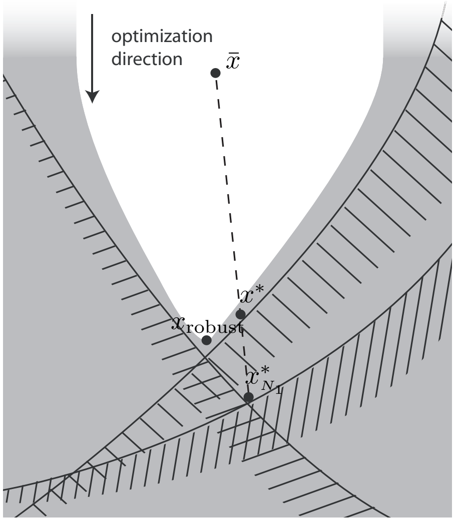


Figure 2. Blank region: feasibility domain for the robust problem. The detuning step updates the solution $x^*_{N_1}$ (always superoptimal) by moving it towards $\bar{x}$. The final solution x^* can be suboptimal for the robust problem.

4. PROOF OF THEOREM 3

Define for brevity $\boldsymbol{\delta}_m^n := (\delta^{(m)}, \delta^{(m+1)}, \ldots, \delta^{(n)})$.
We aim to compute the probability of the event where $V(x^*) > \epsilon$, i.e. we are interested in the probability of the set B of "bad" extractions formally defined as follows:
$$B = \left\{\boldsymbol{\delta}_1^{N_1+N_2} \in \Delta^{N_1+N_2} : V(x^*) > \epsilon\right\}.$$
According to the current notation, we indicate with $\mathbb{P}^{N_1+N_2}\{B\}$ this probability.
Given $\bar{x}$ and $x^*_{N_1}$, consider the half-line in $\mathbb{R}^d$ defined as $\hat{x}[\alpha] := (1-\alpha)x^*_{N_1} + \alpha\bar{x}$, $\alpha \in (-\infty, 1]$ (this extends the line segment at point 4 of the FAST algorithm in Section 3 beyond $x^*_{N_1}$). Now, consider the set Z of points on this half-line with a violation bigger than ϵ:
$$Z = \{\hat{x}[\alpha] : \alpha \in (-\infty, 1] \text{ and } V(\hat{x}[\alpha]) > \epsilon\}.$$
Note that Z is a random set, depending on $\boldsymbol{\delta}_1^{N_1}$ through $x^*_{N_1}$. We need the following lemma.

Lemma 4. The function $V(\hat{x}[\alpha])$ is nonincreasing in $\alpha \in (-\infty, 1]$. $*$

Proof. Let α_1 and α_2 be two reals in the interval $(-\infty, 1]$ with $\alpha_1 < \alpha_2$. Letting $\gamma = \frac{\alpha_2-\alpha_1}{1-\alpha_1}$ one can easily check by substitution that $\hat{x}[\alpha_2] = (1-\gamma)\hat{x}[\alpha_1] + \gamma\bar{x}$, i.e. $\hat{x}[\alpha_2]$ is a convex combination of $\hat{x}[\alpha_1]$ and $\bar{x}$. Take a δ such that $\hat{x}[\alpha_1] \in \mathcal{X}_\delta$. Since also $\bar{x} \in \mathcal{X}_\delta$ and $\mathcal{X}_\delta$ is convex, it follows that $\hat{x}[\alpha_2] \in \mathcal{X}_\delta$. Therefore, a constraint satisfied by $\hat{x}[\alpha_1]$ is also satisfied by $\hat{x}[\alpha_2]$ and this leads to the conclusion that $V(\hat{x}[\alpha_1]) \geq V(\hat{x}[\alpha_2])$. □

Lemma 4 implies that Z is a half-line (see Fig. 3). Next, we state a property which fully characterizes the set B.

Property 5. $\boldsymbol{\delta}_1^{N_1+N_2} \in B$ if and only if $V(x^*_{N_1}) > \epsilon$ and $Z \cap \mathcal{X}_{\delta^{(i)}} \neq \emptyset, \forall i \in \{N_1+1, \ldots, N_1+N_2\}$. $*$

Proof. Looking at Fig. 3, the DETUNING$_{N_2}$ problem in step 4 of the FAST algorithm looks for the point

[3] See e.g. the case of IPMs, Campi et al. (2009a), or Campi et al. (2009b) for problems in control.

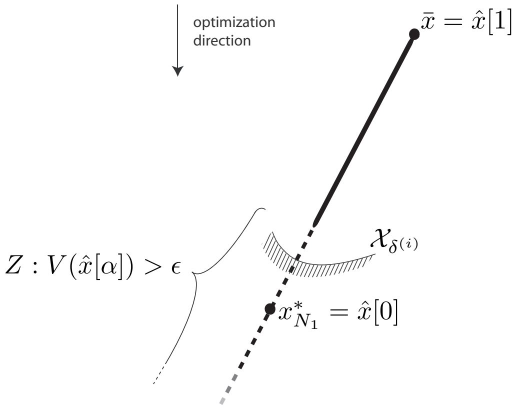


Figure 3. Optimization domain for DETUNING$_{N_2}$.

$x^* \in \overline{x^*_{N_1}\,\bar{x}}$ closest to $x^*_{N_1}$ such that $x^* \in \mathcal{X}_{\delta^{(i)}}$, $\forall i \in \{N_1 + 1, \dots, N_1 + N_2\}$. If $V(x^*_{N_1}) > \epsilon$ and $Z \cap \mathcal{X}_{\delta^{(i)}} \neq \emptyset$, $\forall i \in \{N_1 + 1, \dots, N_1 + N_2\}$, this point falls in Z so that $\boldsymbol{\delta}_1^{N_1+N_2}$ is in B. Viceversa, if either $V(x^*_{N_1}) \leq \epsilon$ or $Z \cap \mathcal{X}_{\delta^{(\bar{i})}} = \emptyset$ for some $\bar{i} \in \{N_1 + 1, \dots, N_1 + N_2\}$, then $x^* \notin Z$ and $\boldsymbol{\delta}_1^{N_1+N_2}$ is not in B, and Property 5 remains proven. □

Based on Property 5 we proceed now to bound the probability of B:

$$\begin{aligned}
&\mathbb{P}^{N_1+N_2}\{B\}\\
&= [\mathbb{1} = \text{indicator function}]\\
&= \int_{\Delta^{N_1+N_2}} \mathbb{1}\{V(x^*_{N_1}) > \epsilon \text{ and } Z \cap \mathcal{X}_{\delta^{(i)}} \neq \emptyset,\\
&\quad \forall i \in \{N_1+1,\dots,N_1+N_2\}\}\mathbb{P}^{N_1+N_2}\{\mathrm{d}\boldsymbol{\delta}_1^{N_1+N_2}\}\\
&= \int_{\Delta^{N_1+N_2}} \mathbb{1}\{V(x^*_{N_1}) > \epsilon\}\mathbb{1}\{Z \cap \mathcal{X}_{\delta^{(i)}} \neq \emptyset, \forall i \in\\
&\quad \{N_1+1,\dots,N_1+N_2\}\}\mathbb{P}^{N_1}\{\mathrm{d}\boldsymbol{\delta}_1^{N_1}\}\mathbb{P}^{N_2}\{\mathrm{d}\boldsymbol{\delta}_{N_1+1}^{N_1+N_2}\}\\
&= [\text{using Fubini's theorem}]\\
&= \int_{\Delta^{N_1}} \mathbb{1}\{V(x^*_{N_1}) > \epsilon\} \left[\int_{\Delta^{N_2}} \mathbb{1}\{Z \cap \mathcal{X}_{\delta^{(i)}} \neq \emptyset, \forall i \in \right.\\
&\quad \left. \{N_1+1,\dots,N_1+N_2\}\}\mathbb{P}^{N_2}\{\mathrm{d}\boldsymbol{\delta}_{N_1+1}^{N_1+N_2}\}\right] \mathbb{P}^{N_1}\{\mathrm{d}\boldsymbol{\delta}_1^{N_1}\}.
\end{aligned} \tag{11}$$

As we are going to show, the inner integral in this latter equation is bounded by $(1-\epsilon)^{N_2}$ for every $\boldsymbol{\delta}_1^{N_1}$, so that

$$\mathbb{P}^{N_1+N_2}\{B\} \leq (1-\epsilon)^{N_2} \int_{\Delta_{N_1}} \mathbb{1}\{V(x^*_{N_1}) > \epsilon\}\mathbb{P}^{N_1}\{\mathrm{d}\boldsymbol{\delta}_1^{N_1}\}. \tag{12}$$

The integral in (12) is exactly $\mathbb{P}^{N_1}\{V(x^*_{N_1}) > \epsilon\}$, a quantity that can be bounded by $B_\epsilon^{N_1,d}$, according to the classical theory of the scenario approach reminded in Section 2, see Theorem 1 in Campi and Garatti (2008). Hence,

$$\mathbb{P}^{N_1+N_2}\{B\} \leq (1-\epsilon)^{N_2} \cdot B_\epsilon^{N_1,d}. \tag{13}$$

From (13) the thesis follows by observing that formula (7) implies $(1-\epsilon)^{N_2} \cdot B_\epsilon^{N_1,d} \leq \beta$.

Thus, to complete the proof we have to show that the inner integral in (11) is bounded by $(1-\epsilon)^{N_2}$. In what follows, we take a fixed $\boldsymbol{\delta}_1^{N_1}$ - so that $\hat{x}[\alpha]$, $\alpha \in (-\infty, 1]$, has to be thought of as a fixed half-line - and the result is proven by working conditionally with respect to $\boldsymbol{\delta}_1^{N_1}$.

By the independence of extractions,

$$\begin{aligned}
&\int_{\Delta^{N_2}} \mathbb{1}\{Z \cap \mathcal{X}_{\delta^{(i)}} \neq \emptyset,\\
&\quad \forall i \in \{N_1+1,\dots,N_1+N_2\}\}\mathbb{P}^{N_2}\{\mathrm{d}\boldsymbol{\delta}_{N_1+1}^{N_1+N_2}\}\\
&= \left(\int_{\Delta} \mathbb{1}\{Z \cap \mathcal{X}_\delta \neq \emptyset\}\mathbb{P}\{\mathrm{d}\delta\}\right)^{N_2}\\
&= (\mathbb{P}\{Z \cap \mathcal{X}_\delta \neq \emptyset\ \})^{N_2}.
\end{aligned}$$

Let

$$\alpha_\epsilon = \sup_{\alpha \in (-\infty,1]} \{\alpha : V(\hat{x}[\alpha]) > \epsilon\}, \tag{14}$$

and note that Z can be written as

$$Z = \{\hat{x}[\alpha] : \alpha \in (-\infty, \alpha_\epsilon)\}.$$

Also, introduce

$$Z_n = \left\{\hat{x}[\alpha] : \alpha \in \left(-\infty, \alpha_\epsilon - \tfrac{1}{n}\right]\right\},$$

a sequence of sets closed to the right and such that $Z_n \uparrow Z$. Clearly, $\{\delta \in \Delta : Z_n \cap \mathcal{X}_\delta \neq \emptyset\} = \{\delta \in \Delta : \hat{x}[\alpha_\epsilon - \frac{1}{n}] \in \mathcal{X}_\delta\}$, that is for $Z_n \cap \mathcal{X}_\delta$ to be non empty, the extreme point $\hat{x}[\alpha_\epsilon - \frac{1}{n}]$ of Z_n must be in $\mathcal{X}_\delta$. Now, by the Definition 2 of violation probability, $\mathbb{P}\{\delta \in \Delta : \hat{x}[\alpha_\epsilon - \frac{1}{n}] \in \mathcal{X}_\delta\} = 1 - V(\hat{x}[\alpha_\epsilon - \frac{1}{n}])$, and applying the property of continuity of probability measures, we conclude that

$$\begin{aligned}
\mathbb{P}\{Z \cap \mathcal{X}_\delta \neq \emptyset\} &= \mathbb{P}\left\{\bigcup_{n=1}^{\infty} \{Z_n \cap \mathcal{X}_\delta \neq \emptyset\}\right\}\\
&= \lim_{n \to \infty} \left(1 - V\left(\hat{x}\left[\alpha_\epsilon - \tfrac{1}{n}\right]\right)\right) \leq 1 - \epsilon,
\end{aligned}$$

where the last inequality follows from the fact that $V(\hat{x}[\alpha_\epsilon - \frac{1}{n}]) \geq \epsilon, \forall n$, see (14). Thus, $(\mathbb{P}\{Z \cap \mathcal{X}_\delta \neq \emptyset\})^{N_2} \leq (1-\epsilon)^{N_2}$ and since this holds for any $\boldsymbol{\delta}_1^{N_1}$ the proof is complete. □

5. AN EXAMPLE

5.1 Uncertain program

The following UP with 200 optimization variables and LMI (Linear Matrix Inequality) constraints resembles problems arising in robust control, see Boyd and Vandenberghe (2004), and well illustrates the procedure developed in this paper.

$$\begin{aligned}
\text{UP} : \min_{x \in \mathbb{R}^{200}} &\sum_{j=1}^{200} x_j\\
\text{subject to: } &\sum_{j=1}^{200} R_j(\delta)B(\delta)R_j(\delta)^T x_j \preceq I\\
&\delta \in \Delta = [0,1]^4;
\end{aligned}$$

where

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; \quad B(\delta) = \begin{pmatrix} \delta_1 & \delta_2 \\ \delta_2 & \delta_3 \end{pmatrix};$$

$$R_j(\delta) = \begin{pmatrix} \cos\left(2\pi \dfrac{j-1}{T(\delta)}\right) & -\sin\left(2\pi \dfrac{j-1}{T(\delta)}\right) \\ \sin\left(2\pi \dfrac{j-1}{T(\delta)}\right) & \cos\left(2\pi \dfrac{j-1}{T(\delta)}\right) \end{pmatrix},$$

for $j = 1, \dots, 200$, and $T(\delta) = 200 + 200^{2\delta_4}$.

$B(\delta)$ is a stochastic matrix and $R_j(\delta)$ is a rotation matrix whose period $T(\delta)$ is also stochastic.

5.2 Classical Scenario Approach vs FAST

Consider $\mathbb{P}$ uniform in $[0,1]^4$ and take $\epsilon = 0.01$ and $\beta = 10^{-9}$. In the classical scenario approach using (4) we write $\sum_{i=0}^{199} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq 10^{-9}$ which yields $N = 29631$, leading to the following sample program

$$\begin{aligned} \mathrm{SP_N} : & \min_{x \in \mathbb{R}^{200}} \sum_{j=1}^{200} x_j \\ & \text{subject to: } \sum_{j=1}^{200} R_j(\delta^{(i)}) B(\delta^{(i)}) R_j(\delta^{(i)})^T x_j \preceq I, \\ & i = 1, \dots, 29631. \end{aligned} \tag{15}$$

Turning to FAST, we take $N_1 = 20 \cdot d = 4000$, as suggested in Section 3.2, and, according to (7), we obtain $N_2 = 2062$.

5.3 Results

Running SP_{N_1} we obtained a solution $x^*_{N_1}$ with objective value $\sum_{j=1}^{200} x^*_{N_1,j} = -1.076$. Next, we selected $\bar{x} = 0$, so that $\hat{x}[\alpha] = (1-\alpha)x^*_{N_1}$, and solved the $\mathrm{DETUNING}_{N_2}$ problem:

$$\begin{aligned} & \mathrm{DETUNING}_{N_2} : \\ & \min_{\alpha \in [0,1]} (1-\alpha) \sum_{j=1}^{200} x^*_{N_1,j} \\ & \text{subject to: } (1-\alpha) \sum_{j=1}^{200} R_j(\delta^{(i)}) B(\delta^{(i)}) R_j(\delta^{(i)})^T x^*_{N_1,j} \preceq I, \\ & i = N_1 + 1, \dots, N_1 + N_2. \end{aligned}$$

The optimal detuning value was $\alpha^* = 0.048$, yielding the final solution $x^* = (1-\alpha^*)x^*_{N_1} = 0.952 x^*_{N_1}$ with objective value $0.952 \cdot (-1.076) = -1.024$.

Going back to the classical scenario approach, we solved (15) which took an execution time about 20 times longer than with FAST and yielded an objective value -1.052.

With smaller value of ϵ, the comparison between the execution times is further unbalanced in favor of FAST: FAST continues to offer a viable approach while the classical scenario approach becomes rapidly impractical.

REFERENCES

Alamo, T., Tempo, R., and Camacho, E. (2009). Randomized strategies for probabilistic solutions of uncertain feasibility and optimization problems. *IEEE Trans. Aut. Control*, 54(11), 2545–2559.

Alamo, T., Tempo, R., and Luque, A. (2010). On the sample complexity of probabilistic analysis and design methods. In J. Willems, S. Hara, Y. Ohta, and H. Fujioka (eds.), *Perspectives in Mathematical System Theory, Control and Signal Processing*, volume 398 of *Lecture Notes in Control and Information Sciences*, 39–50. Springer.

Ben-Tal, A. and Nemirovski, A. (1998). Robust convex optimization. *Mathematics of Operations Research*, 23(4), 769–805.

Ben-Tal, A. and Nemirovski, A. (1999). Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1), 1–13.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2010). *Robust Optimization.* Princeton University Press.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization.* Cambridge University Press.

Calafiore, G. (2009). On the expected probability of constraint violation in sampled convex programs. *Journal of Optimization Theory and Applications*, 143(2), 405–412.

Calafiore, G. and Dabbene, F. (2002). A probabilistic framework for problems with real structured uncertainty in systems and control. *Automatica*, 38(8), 1265–1276.

Calafiore, G. and Campi, M. (2005). Uncertain convex programs: randomized solutions and confidence levels. *Math. Program.*, 102(1), 25–46.

Calafiore, G. and Campi, M. (2006). The scenario approach to robust control design. *IEEE Trans. Aut. Control*, 51(5), 742–753.

Calafiore, G., Dabbene, F., and Tempo, R. (2000). Randomized algorithms for probabilistic robustness with real and complex structured uncertainty. *IEEE Trans. Aut. Control*, 45(12), 2218–2235.

Campi, M., Calafiore, G., and Garatti, S. (2009a). Interval predictor models: identification and reliability. *Automatica*, 45(2), 382–392.

Campi, M. and Garatti, S. (2008). The exact feasibility of randomized solutions of uncertain convex programs. *SIAM J. on Optimization*, 19(3), 1211–1230.

Campi, M., Garatti, S., and Prandini, M. (2009b). The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2), 149 – 157.

El Ghaoui, L., and Lebret, H. (1998). Robust solutions to uncertain semidefinite programs. *SIAM J. on Optimization*, 9(1), 33–52.

El Ghaoui, L., and Niculescu, S.I. (2000). Robust decision problems in engineering: a Linear Matrix Inequality approach. In L. El Ghaoui and S.I. Niculescu (eds.), *Advances in Linear Matrix Inequality Methods in Control.* SIAM.

Fujisaki, Y., Dabbene, F., and Tempo, R. (2003). Probabilistic robust design of LPV control systems. *Automatica*, 39:1323–1337, 2003.

Gupta, A. and Nadarajah, S. (eds.) (2004). *Handbook of Beta Distribution and Its Applications.* Mercel Dekker, Inc.

Henrion, H. and Strugarek, C. (2008). Convexity of chance constraints with independent random variables. *Computational Optimization and Applications*, 41(2), 263–276.

Mutapcic, A., Kim, S., and Boyd, S. (2007). Robust Chebyshev FIR equalization. In *Proceedings of the 50th IEEE Global Communication Conference (GLOBECOM '07).*

Nemirovski, A. and Shapiro, A. (2006). Convex approximations of chance constrained programs. *SIAM J. on Optimization*, 17(4), 969–996.

Oishi, Y. (2007). Polynomial-time algorithms for probabilistic solutions of parameter-dependent Linear Matrix Inequalities. *Automatica*, 43(3), 538 – 545.

Polyak, B.T., and Tempo R. (2001). Probabilistic robust design with linear quadratic regulators. *Systems & Control Letters*, 43:343–353, 2001.

Prékopa, A. (1995). *Stochastic Programming.* Kluwer.

Prékopa, A. (2003). Probabilistic programming. In A. Ruszczyński and A. Shapiro (eds.), *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science.* Elsevier.

Tempo, R., Calafiore, G., and Dabbene, F. (2005). *Randomized Algorithms for Analysis and Control of Uncertain Systems.* Springer.

Vajda, S. (1972). *Probabilistic Programming.* Academic Press.