G. C. Calafiore, IEEE TAC, VOL. 62, NO. 3, Mar. 2017

Repetitive Scenario Design

Paper Introduction
DOI: 10.1109/TAC.2016.2575859
Zhicheng Zhang, D 1, Fujisaki Lab
Osaka University
June 29, 2021

Preliminaries

- Let $q \in \mathbb{Q}$ denote random parameters, with $\mathbb{Q} \subseteq \mathbb{R}^{n_q}$, and let Prob be a probability measure on $\mathbb{Q}$.
- Let $\theta \in \mathbb{R}^n$ be a design variable, and consider a *spec function*, $f(\theta, q) : \mathbb{R}^n \times \mathbb{Q} \to \mathbb{R}$ defining the design constraints and parameterized by uncertainty instance q.
- In particular, for a given design vector θ and realization q of the uncertainty, the a design θ is *robust*, if $f(\theta, q) \leq 0$, $\forall q \in \mathbb{Q}$. Also, f is convex in θ for each fixed $q \in \mathbb{Q}$.
- Define $\omega \doteq \{q^{(1)}, \cdots, q^{(N)}\} \in \mathbb{Q}^N$, where $q^{(i)} \in \mathbb{Q}$, $i = 1, \cdots, N$ are independent random variables, identically distributed (i.i.d.) according to Prob, where $\mathbb{Q}^N = \mathbb{Q} \times \mathbb{Q} \cdots \mathbb{Q}$ (N times).
- Let Prob^N denote the product probability measure on $\mathbb{Q}^N$.

Random Convex Program (RCP)

Standard Scenario Program (SP)

One considers N i.i.d. random samples of the uncertainty $\{q^{(1)}, \cdots, q^{(N)}\} \doteq \omega$, and builds a *random convex program* (RCP):

$$
\begin{array}{llll}
RCP & \min\limits_{\theta \in \Theta \subset \mathbb{R}^n} & c^\top \theta & \\
 & \text{s.t.} & f(\theta, q^{(i)}) \leq 0, & i = 1, \cdots, N.
\end{array}
$$

where $\Theta \in \mathbb{R}^n$ is some given convex and compact domain, and c is the given objective direction.
An optimal solution θ^* to this problem, if it exists, is a random variable which depends on the multiextraction ω, i.e., $\theta^* = \theta^*(\omega)$.

- Worst Case Robust Design $\Rightarrow$ Chance (or Probabilistic) Constrained Optimization $\Rightarrow$ RCP (or SP)

Probability of Violation (Risk)

Given design vector $\theta \in \Theta$. The probability of violation (risk) is defined as

$$V(\theta) = \text{Prob}\{q \in \mathbb{Q} : f(\theta, q) > 0\}$$

- θ is an *ϵ-probability robust design*, if it holds that $V(\theta) \leq \epsilon$
- $V(\theta)$ is a priori, a random variable.

Assumption 1 (feasibility and uniquess)

With probability (w.p.) one with respect to the multi-extraction $\omega = \{q^{(1)}, \cdots, q^{(N)}\}$, RCP is feasible and it attains a unique optimal solution $\theta^*(\omega)$.

Definition 1 (Support constraints)

Let $J^* = c^\top \theta^*$ denote the optimal objective value of RCP. Also, for $j = 1, \cdots, N$, define

$$\begin{aligned} J_j^* = \min_{\theta \in \Theta \subset \mathbb{R}^n} \quad & c^\top \theta \\ \text{s.t.} \quad & f(\theta, q^{(i)}) \leq 0, \quad i \in \{1, \cdots, N\} \setminus j \end{aligned}$$

The j-th constraint in (2) is said to be a *support constraint* if $J_j^* < J^*$. In other words, its removal changes the solutions θ^*.

- The *complexity* of RCP is the number of support constraints.
- The number of support constraints (complexity) cannot exceed n (the number of decision variable θ).
- In case of *fully-supported (f.s.)* problems, RCP has n support constraints *w.p. one*, whenever, $N \geq n$.

Theorem

Let Assumption 1 hold. For given $\epsilon \in [0,1]$ and $N \geq n$, it holds that

$$\mathrm{Prob}^N\{\omega : V(\theta^*(\omega)) \leq \epsilon\} \geq \sum_{i=n}^{N} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \doteq 1 - \beta_\epsilon(N)$$

Random Smaple N

Given $\epsilon, \beta \in (0,1)$. If N is an integer such that

$$N \geq \frac{2}{\epsilon}(\mathrm{In}\beta^{-1} + n - 1) \text{ then } \mathrm{Prob}^N\{V(\theta^*(\omega)) \geq \epsilon\} \leq \beta$$

- Furthermore, the exact minimal value of N can be easily found numerically by searching for the least integer N such that $\sum_{i=n}^{N} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \geq 1 - \beta$.

Repetitive Scenario Design

- Repetitive Scenario Design (RSD) is a randomized approach to robust design based on iterating two phases:
  1. a standard scenario design phase that uses N scenarios (design samples), followed by
  2. a randomized feasibility test phase that uses N_o test samples on the scenario solution.
- The above steps are repeated until the desire level of probabilistic feasibility is eventually obtained.
- The plain (one-shot) scenario design becomes just one of the possibilities, sitting at one extreme of the tradeoff curve, in which one insists in finding a solution in a single repetition: this comes at the cost of possibly high N.
- Other possibilities along the tradeoff curve use lower N values, but possibly require more than one repetition.

Repetitive Scenario Design (Main Idea)

- Each iteration of RCP with N scenarios can be thought as a biased "coin toss". The toss is successful if $V(\theta_k^*) \leq \epsilon$, while it fails if $V(\theta^*) \geq \epsilon$.
- Success in one toss comes at the price of possibly high N.
- What if use a lower N (i.e., bias the coin with higher $\beta_\epsilon(N)$) and then check the resulting solution ?
- Idea: while the probability of being successful in one shot is low, if toss the coin repeatedly, the probability of being successful at some toss becomes arbitrarily high.
- Thus, an iterative approach in two stages is set up: a scenario optimization stage, and a feasibility check phase.

RSD (Idea Feasibility Oracle)

- An idea feasibility oracle, called a *deterministic ϵ-violation oracle* (ϵ-DVO), that returns as output a flag value which is `true` if $V(\theta^*) \leq \epsilon$, and `false` otherwise.

ALGO. 1 (RSD with ϵ-DVO)

Input data: integer $N \geq n$, level $\epsilon \in [0, 1]$.
Output data: solution θ^*.
Initialization: set iteration counter to $k = 1$.

1. (Scenario step) Generate N i.i.d. samples according to Prob, $\omega^{(k)} = \{q_k^{(1)}, \cdots, q_k^{(N)}\}$, and solve RCP. Let θ_k^* be the resulting optimal solution
2. (ϵ-DVO step) If $V(\theta_k^*) \leq \epsilon$, then set flag to `true`; else `false`.
3. (Exit condition) If flag is `true`, then exit and return. current solution $\theta^* \leftarrow \theta_k^*$; else set $k \leftarrow k + 1$ and goto 1.

Theorem (2)

Let Assumption 1 hold. Given $\epsilon \in [0, 1]$ and $N \geq n$, define the running time K of ALGO. 1 as the value of the iteration counter k when the algorithm exits. Then

1. *The solution θ^* returned by ALGO. 1 is an ϵ-probabilistic robust design, i.e., $V(\theta^*) \leq \epsilon$.*
2. *The expected running time of ALGO. 1 is $\leq (1 - \beta_\epsilon(N))^{-1}$, and equality holds if the SP is f.s. w.p. one.*
3. *The running time of ALGO. 1 is $\leq k$ with probability $\geq 1 - \beta_\epsilon(N)^k$, and equality holds if the SP is f.s. w.p. one.*

Such ideal oracle is hardly realizable in practice, we next introduce a *randomized ϵ'-violation oracle* (ϵ'-RVO):

Definition (Randomized ϵ'-violation oracle)

Input data: integer N, N_o, level $\epsilon' \in [0, 1]$. Output data: a logic flag, `true` or `false`.

1. Generate N_o i.i.d. samples $\omega_o \doteq \{q^{(1)}, \cdots, q^{(N_o)}\}$ according to Prob.
2. For $i = 1, \cdots, N_o$, let $v_i = 1$ if $f(\theta, q^{(i)}) > 0$ and $v_i = 0$ otherwise.
3. If $\sum_i v_i \leq \epsilon' N_o$, return `true`, else return `false`.

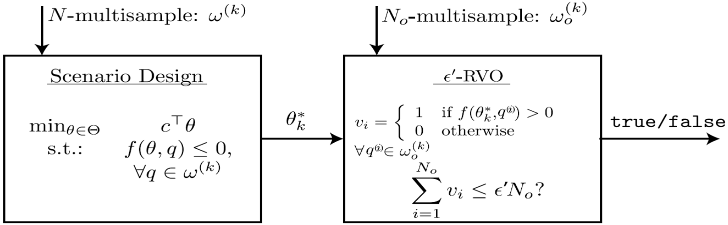

Fig. 1. Generic stage k of Algorithm 2.

ALGO. 2 (RSD with ϵ'-RVO)

Input data: integer N, N_o, level $\epsilon' \in [0, 1]$.
Output data: solution θ^*.
Initialization: set iteration counter to $k = 1$.

1. (Scenario step) Generate N i.i.d. samples according to Prob, $\omega^{(k)} = \{q_k^{(1)}, \cdots, q_k^{(N)}\}$, and solve RCP. Let θ_k^* be the resulting optimal solution.
2. (ϵ'-RVO step) Call ϵ'-RVO with current θ_k^* as input, and set flag to `true` or `false` according to the output of the ϵ'-RVO.
3. (Exit condition) If flag is `true`, then exit and return current solution $\theta^* \leftarrow \theta_k^*$; else set $k \leftarrow k + 1$ and goto 1.

Randomized Oracle

Theorem (RSD with ϵ'-RVO)

Let Assumption 1 hold. Let $\epsilon, \epsilon' \in [0,1]$, $\epsilon \leq \epsilon'$ and $N \geq n$ be given. Define the event BadExit *in which ALGO. 2 exits returning a "bad" solution θ^*:* BadExit $\doteq$ {ALGO. 2 returns $\theta^* : \mathrm{V}(\theta^*) > \epsilon$}.
The following statements holds.

1. Prob{BadExit} $\leq \frac{\mathrm{Fbeta}((1-\epsilon')N_o, \epsilon' N_o + 1; 1-\epsilon)}{1 - H_{1,\epsilon'}(N, N_o)}$
2. *The expected running time of ALGO. 2 is* $\leq 1 - H_{1,\epsilon'}(N, N_o)^{-1}$*, and equality holds if the scenario problem is f.s. w.p. one.*
3. *The running time of ALGO. 2 is* $\leq k$ *w.p.* $\geq 1 - H_{1,\epsilon'}(N, N_o)^k$*, and equality holds if the scenario problem is f.s. w.p. one.*

Here Fbeta denotes the cumulative distribution of a beta density, and $H_{1,\epsilon'}(N, N_o)$ has an explicit expression in terms of beta-Binomial distributions. (Defined in Lemma 1)

RSD (Asymptotic Bounds)

- A key quantity related to the expected running time of ALGO. 2 is $H_{1,\epsilon'}(N, N_o)$, which is the upper tail of a beta-Binomial distribution.
- It is therefore useful to have a more "manageable" albeit approximate, expression for $H_{1,\epsilon'}(N, N_o)$.

Corollary

For $N_o \to \infty$ it holds that $H_{1,\epsilon'}(N, N_o) \to \beta_{\epsilon'}(N)$.

- For large N_o, $\epsilon \leq \epsilon'$, we have $H_{1,\epsilon'}(N, N_o) \simeq \beta_{\epsilon'}(N) \geq \beta_{\epsilon}(N)$,

$$\hat{K} \doteq \frac{1}{1 - H_{1,\epsilon'}(N, N_o)} \simeq \frac{1}{1 - \beta_{\epsilon'}(N)} \geq \frac{1}{1 - \beta_{\epsilon}(N)}$$

This last equation gives us an approximate, asymptotic, expression for the upper bound $\hat{K}$ on the expected running time of ALGO. 2.

Example(Robust finite-horizon input design)

- Consider a system of the form

$$x(t+1) = A(q)x(t) + Bu(t), \quad t = 0, 1, \cdots ; \; x(0) = 0$$

where $u(t)$ is a scalar input signal, and $A(q) \in \mathbb{R}^{n_a \times n_a}$ is an interval uncertain matrix of the form

$$A(q) = A_0 + \sum_{i,j=1}^{n_a} q_{ij} e_i e_j^\top, \; |q_{ij}| \leq \rho, \; \rho \geq 0,$$

e_i is a vector of all zeros, except for a one in the i-th entry.

- Given a final time $T \geq 1$ and a target state $\bar{x}$, the problem is to find an input sequence $\{u(0), \cdots, u(T-1)\}$ such that

1. the state $x(T)$ is robustly contained in a small ball around the target state $\bar{x}$;
2. the input energy $\sum_k u(k)^2$ is not too large.

- Let $x(T) = x(T;q) = \mathcal{R}(q)u$, where $\mathcal{R}(q)$ is the T-reachability matrix of the system (for a given q), and $u \doteq (u(0), \cdots, u(T-1))$.
- The design goals in the form of minimization of a level γ such that
$$\|x(T;q) - \bar{x}\|_2^2 + \lambda \sum_{t=0}^{T-1} u(t)^2 \leq \gamma,$$
where $\lambda \leq 0$ is a tradeoff parameter. Letting $\theta = (u, \gamma)$, the problem is formally stated in our framework by setting
$$f(\theta, q) \leq 0, \text{ where } f(\theta, q) \doteq \|\mathcal{R}(q)u - \bar{x}\|_2^2 + \lambda \|u\|_2^2 - \gamma.$$
- Robust finite-horizon input design
$$\begin{aligned} \min_{\theta=(u,\gamma)} \quad & \gamma \\ \text{s.t.} \quad & f(\theta, q^{(i)}) \leq 0, \quad i = 1, \cdots, N. \end{aligned}$$
where the uncertain parameter q is random and uniformly distributed in the hypercube $\mathbb{Q} = [-\rho, \rho]^{n_a \times n_a}$.

Example

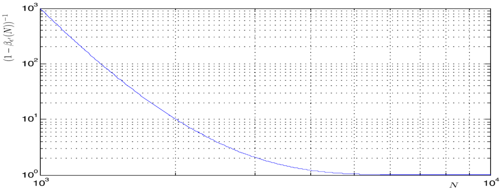

Fig. 2. Example in Section III-A: Log-log plot of $(1-\beta_{\epsilon'}(N))^{-1}$ vs. N.

- We set $T = 10$, thus the size of the decision variable $\theta = (u, \gamma)$ of the scenario problem is $n = 11$.
- We set the desired level of probabilistic robustness to $1 - \epsilon = 0.995$, i.e., $\epsilon = 0.005$, and require a level of failure of randomized method below $\beta = 10^{-12}$.
- Using a plain (one-shot) scenario approach, imposing $\beta_\epsilon(N) \leq \beta$ would require $N \geq 10440$ scenarios.
- Reduce this N figure via RSD.

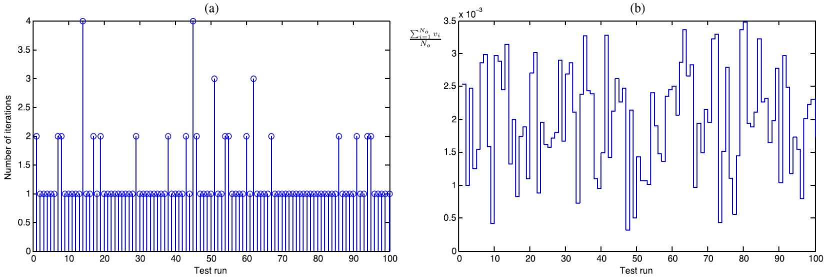

Fig. 3. Example in Section III-A: (a) Repetitions of Algorithm 2 in the 100 test runs. (b) Levels of empirical violation probability evaluated by the oracle upon exit, in the 100 test runs.

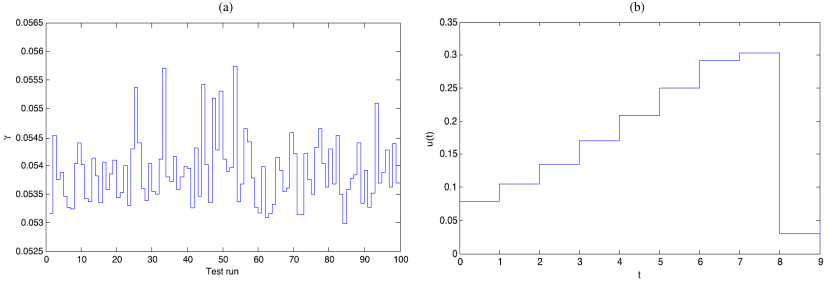

Fig. 4. Example in Section III-A: (a) Optimal γ level returned by Algorithm 2 in the 100 test runs. (b) Average over the 100 test runs of the optimal input $u(t)$ returned by Algorithm 2.

Conclusion

- RSD aims to reduce the sample size with two stage: a scenario optimization stage, and a feasibility check phase
- A clear tradeoff between the number N of design samples and the ensuing expected number of repetitions required by the RSD algorithm.
- Extension : how about sparse input design via RSD ?

$$\begin{aligned} \min_{u \in \mathbb{R}^m} \quad & \ell(x, u) \\ \text{s.t.} \quad & x(t+1) = A(q)x(t) + Bu(t), \ x(0) = \xi \\ & \|u\|_0 \leq s \\ & f(u, x(q^{(i)})) \leq 0, \quad i = 1, \cdots, N \end{aligned}$$

where $f(u, x(q^{(i)})) = \|x(T; q) - 0\|_2 - \gamma = \|\mathcal{R}(q)u\|_2 - \gamma$