AIAA SciTech Forum
6-10 January 2020, Orlando, FL
AIAA Scitech 2020 Forum
10.2514/6.2020-2130

Risk-Based Design Optimization Via Probability of Failure, Conditional Value-at-Risk, and Buffered Probability of Failure

Anirban Chaudhuri*[1], Matthew Norton†[2], and Boris Kramer‡[3]
[1] *Massachusetts Institute of Technology, Cambridge, MA, 02139, USA*
[2] *The Naval Postgraduate School, CA, 93943, USA*
[3] *University of California San Diego, CA, 92093, USA*

When designing systems, uncertainties must be dealt with at various levels. The designer must define appropriate cost and constraint functions that account for such uncertainties and capture the risk associated with unwanted system behavior. The choice of these cost and constraint functions additionally plays an important role in the convergence behavior of the optimization and, among other things, the final design. This paper studies different types of risk-based optimization problem formulations that can aid in efficient and robust design of complex engineering systems. In tutorial form, the paper describes risk-based optimization problem formulations, specifically, reliability-based design optimization, conditional-value-at-risk-based optimization, and buffered-failure-probability-based optimization. The properties of each formulation are analyzed and general guidelines for the appropriate choice of the optimization problem are provided for a given application setup. An in-depth understanding of the different optimization problems should facilitate development of future methods for designing safe engineering systems.

I. Introduction

The design of efficient aerospace systems requires quantifying and accounting for risk in the presence of uncertainties. This is not only vital to ensure safety of designs but also to safeguard against costly design alterations late in the design cycle. The traditional approach using safety factors to compensate for uncertainties in a deterministic optimization setup often ends up with overly conservative designs without a precise quantification of the involved risk. Hence, the need to understand risk-based optimization problems that take the associated risk into account in order to design safe systems that are not overly conservative is paramount. Depending on the application and the available knowledge about the application and the parameters, different optimization problem formulations are required to design an efficient and reliable system. The goal of this work is to provide various options for risk-based optimization problem formulations and analysis of each formulation to help make a choice for a particular application. This paper is meant to serve as an overview introducing and describing to the community the different available risk measures and the accompanying optimization problems in the context of complex engineering applications in a tutorial form. We provide basic algorithms for estimating the risk measures and analyze the properties of each optimization problem. We also highlight useful insights from each risk-based optimization formulation and the implications to future engineering design research.

In this work, we concentrate on three different risk measures to formulate the optimization problems: (i) probability of failure (PoF), (ii) conditional value-at-risk (CVaR), and (iii) buffered probability of failure (bPoF). The PoF risk measure is widely and successfully used as a constraint in reliability-based design optimization (RBDO) [1, 2]. Much methodological and algorithmic research exists to make the RBDO efficient and applicable to a broad range of engineering problems as we discuss later. In this work, we highlight some lesser-used methods for optimization under uncertainty that lead to alternative design problem formulations. We discuss the advantages of using these alternative ways of measuring risk in the design optimization cycle and its effect on the final design under uncertainty. To that end, we describe optimization problem formulations using CVaR [3] and bPoF [4]. Conditional value-at-risk is a risk measure that has been widely used for quantifying financial risk in portfolio optimization [5–7], and more recently in engineering design [8–11] and PDE-constrained optimization [12–15]. Lastly, we describe design optimization with bPoF, a recently introduced risk measure that possesses beneficial properties when it comes to optimization problem formulations [4, 16, 17]. CVaR and bPoF are conservative risk measures that add a buffer zone to the limiting threshold.

*Postdoctoral Associate, Department of Aeronautics and Astronautics, anirbanc@mit.edu, AIAA member.
†Assistant Professor, Department of Operations Research, mnorton@nps.edu.
‡Assistant Professor, Department of Mechanical and Aerospace Engineering, bmkramer@ucsd.edu, AIAA member.

Adding a buffer prevents excessive wear and tear of the system (especially when experimental data is required) by not needing to simulate limiting conditions at the brink of failure of the system. This is typically handled by adding safety factors to the threshold. However, it has been shown before that the probabilistic approach is more efficient than the safety factor approach. We will discuss in detail how the nature of probabilistic conservativeness introduced through CVaR and bPoF makes practical sense since it is based on the magnitude of failure. We will discuss the probabilistic approaches through some risk-based optimization formulations.

The remainder of this extended abstract is organized as follows. The problem setup is described in Section II. The different risk-based optimization problem formulations along with the risk measures used in this work are described in Section III. Section IV explains the features of different risk-based optimization formulations through numerical experiments on the widely used short column problem. Section V presents the concluding remarks.

II. Problem Setup

To define risk-based optimization formulations, we define both a cost and a constraint function. Let the quantity of interest of an engineering system be computed from the model $f : \mathcal{D} \times \Omega \mapsto \mathbb{R}$ as $f(\boldsymbol{d}, Z)$, where the inputs to the system are the n_d design variables $\boldsymbol{d} \in \mathcal{D} \subseteq \mathbb{R}^{n_d}$ and the n_z random variables $Z \in \Omega \subseteq \mathbb{R}^{n_z}$ with the probability density function π. The design variable space is denoted by $\mathcal{D}$ and the random sample space is denoted by Ω. The vector of a realization of the random variables Z is denoted by z.

The failure of the system is described by a limit state function $g : \mathcal{D} \times \Omega \mapsto \mathbb{R}$ and a critical threshold $t \in \mathbb{R}$. Without loss of generality,

$$g(\boldsymbol{d}, z) > t,$$

defines failure of the system. The limit state function in most engineering applications requires the solution of a system of equations (such as, ordinary differential equations or partial differential equations). The limit state function $g(\boldsymbol{d}, Z)$ can also be interpreted as a random variable, requiring a statistical formulation during the optimization. Risk measures are a way of converting the random variable into a single quantity that can be used in the optimization formulation. Risk-based optimization centers around the definition of risk measures and associated optimization problem formulations accounting for the risk induced onto system-level outputs by uncertainties.

To motivate the upcoming use of risk measures, we take a closer look at the limit-state function g and its use to characterize *failure events*. In the standard setting, a failure event is characterized by a realization of Z for some fixed design $\boldsymbol{d}$ that leads to $g(\boldsymbol{d}, z) > t$. This hard-threshold characterization of system failure, however, potentially ignores important information quantified by the magnitude of $g(\boldsymbol{d}, z)$. For example, there may be a large difference between the event $g(\boldsymbol{d}, z) = t + .1$ and $g(\boldsymbol{d}, z) = t + 100$, the later characterizing a catastrophic system failure. This is not captured by a hard-threshold attitude toward failure events. If the magnitude of $g(\boldsymbol{d}, z)$ has meaning, it is important to know if these events are common for a particular design. Similarly, one could also consider events $g(\boldsymbol{d}, z) = t - .1$ and $g(\boldsymbol{d}, z) = t - 100$. A hard-threshold assessment deems both of these events as *non-failure* events, even though $g(\boldsymbol{d}, z) = t - .1$ is clearly a *near-failure* event compared the later. However, a hard-threshold characterization of failure will overlook these important *near-failure* events and consider them as safe realizations of g. In reality, failure events do not usually occur using a hard-threshold rule. Even if they do, determination of the *true* threshold will also involve uncertainty, blending statistical estimation, expert knowledge, and system models. Therefore, the choice of threshold should be involved in any discussion of measures of failure risk.

In the following sections, we will start with the discussion of the most common measure of risk in PoF that uses a hard-threshold characterization of of failure. We then discuss alternative risk measures, such as CVaR and bPoF, that do not use a hard-thresholding rule. CVaR and bPoF consider both the magnitude and frequency of failure and near-failure events in their summary assessments of failure risk. We will see the pros and cons of different risk measures from a numerical perspective in the context of risk-based engineering optimization formulations.

III. Risk-Based Optimization Problem Formulations

This section describes a few risk-based optimization problem formulations which use three different risk measures: Section III. A introduces reliability-based design optimization (RBDO) with the probability of failure (PoF); Section III. B describes conditional value-at-risk in optimization (CVaR); Section III. C presents optimization based on buffered probability of failure (bPoF).

A. Reliability-Based Design Optimization

The RBDO problem uses probability of failure as the risk measure and has been the most popular choice for designing reliable engineering systems.

1. Risk Measure: Probability of Failure

The PoF is defined via the limit state function and a failure threshold t as

$$p_t(g(\boldsymbol{d}, Z)) = \mathbb{P}(g(\boldsymbol{d}, Z) > t). \tag{1}$$

Typically, Monte Carlo (MC) simulation is used to estimate the PoF when dealing with nonlinear limit state functions. The MC estimate of the PoF $\hat{p}_t(g(\boldsymbol{d}, Z))$ for a given design $\boldsymbol{d}$ is

$$\hat{p}_t(g(\boldsymbol{d}, Z)) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}_{\mathcal{G}}(\boldsymbol{d}, z_i), \tag{2}$$

where $z_1, \ldots, z_m$ are m samples distributed according to the probability density π, $\mathcal{G} = \{z \mid g(\boldsymbol{d}, z) > t\}$ is the failure set, and $\mathbb{I}_{\mathcal{G}} : \mathcal{D} \times \Omega \to \{0, 1\}$ is the indicator function defined as

$$\mathbb{I}_{\mathcal{G}}(\boldsymbol{d}, z) = \begin{cases} 1, & z \in \mathcal{G} \\ 0, & \text{else.} \end{cases} \tag{3}$$

The MC estimator is unbiased and the variance decay rate is proportional to $1/m$. Algorithm 1 describes standard MC sampling for approximating PoF.

Algorithm 1 Sampling-based estimation of PoF.

Input: m i.i.d. samples $z_1, \ldots, z_m$ of random variable, design variable $\boldsymbol{d}$, failure threshold t, and limit state function $g(\boldsymbol{d}, Z)$.
Output: $\hat{p}_t(g(\boldsymbol{d}, Z))$.
1: Evaluate limit state function at the samples to get $g(\boldsymbol{d}, z_1), \ldots, g(\boldsymbol{d}, z_m)$.
2: Compute $\mathbb{I}_{\mathcal{G}}(\boldsymbol{d}, z_i)$ for $i = 1, \ldots, m$ using Equation (3).
3: Estimate $\hat{p}_t(g(\boldsymbol{d}, Z))$ using Equation (2).

2. Optimization Problem: RBDO

The most common RBDO formulation involves the use of a PoF constraint as

$$\begin{aligned} \min_{\boldsymbol{d} \in \mathcal{D}} \quad & \mathbb{E}_\pi[f(\boldsymbol{d}, Z)] \\ \text{subject to} \quad & p_t(g(\boldsymbol{d}, Z)) \leq 1 - \alpha, \end{aligned} \tag{4}$$

where $1 - \alpha$ is the acceptable threshold on the PoF. The RBDO problem formulation designs a system with optimal characteristics, in terms of $f(\boldsymbol{d}, Z)$, such that it maintains a reliability of at least α. For our upcoming discussion, it is helpful to point out that a constraint of PoF is equivalent to a constraint on the α-quantile. The α-quantile at level α, also known as the value-at-risk (VaR), is defined as

$$\mathrm{VaR}_\alpha[g(\boldsymbol{d}, Z)] = \min_{c \in \mathbb{R}} \left\{\mathbb{P}(g(\boldsymbol{d}, Z) \leq c) = \alpha\right\}. \tag{5}$$

In the context of our optimization problem, using the same value of t and α, formulation (4) can be written equivalently as

$$\begin{aligned} \min_{\boldsymbol{d} \in \mathcal{D}} \quad & \mathbb{E}_\pi[f(\boldsymbol{d}, Z)] \\ \text{subject to} \quad & \mathrm{VaR}_\alpha[g(\boldsymbol{d}, Z)] \leq t. \end{aligned} \tag{6}$$

PoF and VaR are natural counterparts in that they are inverses of one another as functions of t and α. They are also natural measures of the tail of the distribution of $g(\boldsymbol{d}, Z)$. When one knows that the largest $100 \times (1 - \alpha)\%$ outcomes

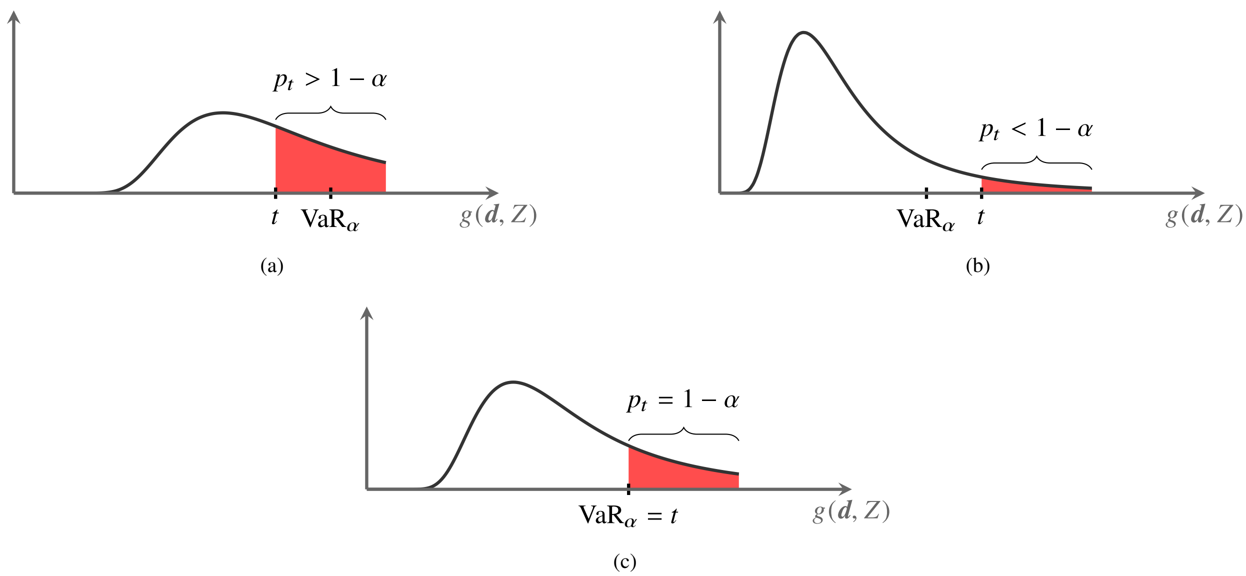


Fig. 1 Illustration of equivalence of PoF and VaR showing that the two quantities converge at the constraint threshold.

are the ones of interest, the quantile is a measure of the best-case scenario within the set of these tail events. When one knows that outcomes larger than a given threshold t are of interest, PoF provides a measure of the frequency of these "large" events. This equivalence of PoF and VaR risk constraints is illustrated in Figure 1.

In Section III. B, we will introduce CVaR, which was originally developed as an alternative for VaR with superior mathematical properties. Then, we will introduce a related quantity called bPoF in Section III. C. Just as VaR can be seen as the inverse of PoF, we have that bPoF is the inverse of CVaR.

3. Discussion about RBDO

When dealing with highly nonlinear limit state functions, the most straightforward method is to use a MC estimate for PoF. However, the PoF estimation requires sampling from the tails of the distribution, which can often make MC estimators expensive. A wealth of literature exists for methods that have been developed to deal with the computational complexity of PoF estimation. We outline a few such methods below. First, variance reduction techniques such as importance sampling, adaptive importance sampling, Markov chain Monte Carlo (see [18–20] and the references therein) offer computational advantages. While the decay rate of the MC estimate cannot be improved upon, the variance of the MC estimator can be reduced, which offers computational advantages in that fewer (suitably chosen) MC samples are needed to obtain reliable PoF estimates. Multifidelity methods have been successfully used to identify proper importance sampling densities at low-computational cost [21–23]. Note that here only the samples are chosen differently, yet the limit state function is *not* approximated. Second, a popular approach in the structural community to deal with the computational complexity of PoF estimation is to use reliability index methods (e.g., FORM, SORM, etc. [24, 25]). Reliability index methods use geometric approximations of the limit state function to reduce the computational effort. However, when the limit state function is nonlinear, the reliability index method could lead to inaccuracies in the estimate. Third, adaptive data-driven surrogates for the limit state failure boundary identification have been developed to improve computational efficiency [26, 27]. Fourth, some recent multifidelity/multi-information-source methods have led to computational savings for the PoF estimate [28–30]. In sum, these existing efficient methods for PoF estimation make the RBDO problem computationally tractable and a natural choice for risk-based optimization problem formulation. Since the goal of this work is to analyze the properties of the different risk-based optimization problem formulations, we use the standard MC estimate for probability of failure from Equation (2) when solving the RBDO problem. The reader is referred to some of the above literature for more sophisticated and efficient RBDO implementations.

While there are several advantages of the widely studied RBDO problem, there are also several potential drawbacks associated with RBDO. First, PoF is not necessarily a convex function w.r.t. design variables even when the underlying limit state function is convex. Thus, we cannot formulate a convex optimization problem, possibly based upon a sample-average-approximation (SAA), which guarantees convergence to a global optimum (even when underlying functions f and g are convex). Global optimizers can be used instead, but they generally do not provide convergence guarantees. Second, sensitivity issues can arise in the PoF gradients, which negatively affects the optimizer. During optimization, the effect of small changes in design on the PoF is analyzed and this directly relates to the PoF gradients. The accuracy of the PoF gradients estimated using approximate methods, such as, finite difference, is not always good. There are better methods for estimating the gradients, but they have been developed under potentially restrictive assumptions [31–33], which might not be easily verifiable for practical problems. Third, a qualitative drawback of PoF (and VaR) is that it ignores the magnitude of failure of the system and instead encodes a hard threshold via a binary function evaluation. Whether the failure is minor ($g(\boldsymbol{d}, \boldsymbol{z})$ slightly greater than t) or catastrophic ($g(\boldsymbol{d}, \boldsymbol{z}) \gg t$), both events are treated equally (indicator function is 1 in both cases). PoF can be viewed as an *optimistic* measure of the size and frequency of tail events. Fourth, PoF can suffer from sensitivity to the failure threshold due to it being a discontinuous function w.r.t. t. However, as mentioned before, the choice of failure threshold is often uncertain, thus, one would ideally like to have a measure of reliability that is not extremely sensitive to small changes in t.

B. Conditional-Value-at-Risk-Based Design Optimization

This section describes CVaR, also known as α-superquantile, and an associated risk-averse optimization problem formulation. CVaR emphasizes tail events, and from an engineering perspective it is important to manage such tail risks, e.g., minimizing or constraining CVaR.

1. Risk measure: CVaR

Intuitively, CVaR can be understood as simply a tail expectation, or an average over a portion of worst-case outcomes. Given a fixed design $\boldsymbol{d}$ and a distribution of potential outcomes $g(\boldsymbol{d}, Z)$, CVaR at level α is, loosely speaking, the expected value of the largest $100 \times (1-\alpha)\%$ realizations of $g(\boldsymbol{d}, Z)$.

The precise definition of CVaR is based on the α-quantile, $\text{VaR}_\alpha[g(\boldsymbol{d}, Z)]$ (see Equation (5)). In its simplest form, assuming that the cumulative distribution of $g(\boldsymbol{d}, Z)$ is continuous and strictly increasing for all $\boldsymbol{d}$, the CVaR at level α, CVaR_α, can be defined [3] as a tail expectation given by

$$\text{CVaR}_\alpha[g(\boldsymbol{d}, Z)] = \mathbb{E}_\pi\left[g(\boldsymbol{d}, Z) \mid g(\boldsymbol{d}, Z) \geq \text{VaR}_\alpha[g(\boldsymbol{d}, Z)]\right].$$

Thus, we can think of $\text{CVaR}_\alpha[g(\boldsymbol{d}, Z)]$ as the conditional expectation of $g(\boldsymbol{d}, Z)$ with the condition that $g(\boldsymbol{d}, Z)$ is not less than $\text{VaR}_\alpha[g(\boldsymbol{d}, Z)]$. It follows from the definition that $\text{CVaR}_\alpha[g(\boldsymbol{d}, Z)] > \text{VaR}_\alpha[g(\boldsymbol{d}, Z)]$. Figure 2 illustrates the CVaR risk measure.

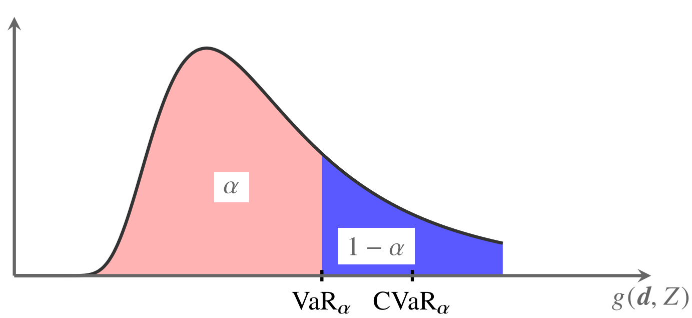


Fig. 2 Illustration for CVaR: expectation of the worst-case $1-\alpha$ outcomes shown in blue is $\text{CVaR}_\alpha[g(\boldsymbol{d}, Z)]$.

Algorithm 2 describes standard MC sampling for approximating CVaR. The second term on the right hand side in (7) is nonzero for the case $\sum_{j=1}^{k_\alpha - 1} \frac{1}{m} \neq 1-\alpha$ and is based on the idea of splitting the probability atom at $\text{VaR}_\alpha[g(\boldsymbol{d}, Z)]$ (see [5]).

2. Optimization problem: CVaR as constraint

As noted in the previous section, the PoF constraint of the RBDO problem in (4) can be viewed as a VaR constraint (as seen in (6)). The PoF constraint (and thus the VaR constraint) does not consider the magnitude of failure events, but

Algorithm 2 Sampling-based estimation of VaR_α and CVaR_α.

Input: m i.i.d. samples $z_1, \ldots, z_m$ of random variable, design variable $\boldsymbol{d}$, risk level $\alpha \in (0, 1)$, and limit state function $g(\boldsymbol{d}, Z)$.

Output: $\widehat{\text{VaR}}_\alpha[g(\boldsymbol{d}, Z)]$, $\widehat{\text{CVaR}}_\alpha[g(\boldsymbol{d}, Z)]$.

1: Evaluate limit state function at the samples to get $g(\boldsymbol{d}, z_1), \ldots, g(\boldsymbol{d}, z_m)$.

2: Sort values of limit state function in descending order and relabel the samples so that

$$g(\boldsymbol{d}, z_1) > g(\boldsymbol{d}, z_2) > \ldots > g(\boldsymbol{d}, z_m).$$

3: Find the index $k_\alpha = \lfloor m(1-\alpha) \rfloor$ to estimate $\widehat{\text{VaR}}_\alpha[g(\boldsymbol{d}, Z)] \leftarrow g(\boldsymbol{d}, z_{k_\alpha})$.

4: Estimate

$$\widehat{\text{CVaR}}_\alpha[g(\boldsymbol{d}, Z)] = \frac{1}{m(1-\alpha)} \sum_{j=1}^{k_\alpha - 1} g(\boldsymbol{d}, z_j) + \frac{1}{1-\alpha}\left(1 - \alpha - \sum_{j=1}^{k_\alpha - 1} \frac{1}{m}\right) \widehat{\text{VaR}}_\alpha[g(\boldsymbol{d}, Z)]. \tag{7}$$

only whether they are larger than the failure threshold. This could be a potential drawback for engineering applications. A CVaR constraint, on the other hand, will consider the magnitude of such events, specifically constraining the expected value of the largest $100(1-\alpha)\%$ realizations of $g(\boldsymbol{d}, Z)$. Additionally, depending upon the actual construction of $g(\boldsymbol{d}, z)$ and the accuracy of the sampling procedure, the CVaR constraint may have numerical advantages over the VaR constraint when it comes to optimization as discussed later. In particular, we have the optimization problem formulation

$$\begin{aligned} \min_{\boldsymbol{d} \in \mathcal{D}} \quad & \mathbb{E}_\pi[f(\boldsymbol{d}, Z)] \\ \text{subject to} \quad & \text{CVaR}_\alpha(g(\boldsymbol{d}, Z)) \le t, \end{aligned} \tag{8}$$

where α is the desired reliability level given the limit-state failure threshold t. The CVaR-based formulation typically leads to a more conservative design than when PoF is used. This can be observed by noting that $\text{CVaR}_\alpha(g(\boldsymbol{d}, Z)) \le t \implies \text{VaR}_\alpha(g(\boldsymbol{d}, Z)) \le 0 \implies p_t(g(\boldsymbol{d}, Z)) \le 1 - \alpha$. Therefore, if the design satisfies the CVaR constraint, then the design will also satisfy the related PoF constraint. Additionally, since the CVaR constraint assures that the average of the $(1-\alpha)$ tail is no larger than t, it is likely that the probability of exceeding t (PoF) is strictly smaller than $1-\alpha$ and is thus a conservative design if the target reliability was α. Intuitively, this conservatism comes from the fact that CVaR considers the magnitude, or degree, of the worst failure events.

The formulation with CVaR as the constraint is useful when the designer is not sure about the failure boundary location for the problem but requires a certain level of reliability from the design. For example, consider the case where the failure is defined as maximum stress of a structure not exceeding a certain value. However, the designers cannot agree on the cut-off value for stress but can agree on the desired level of reliability they want. One can use this formulation to design a structure with a given reliability $(1-\alpha)$ while constraining a conservative estimate of the cut-off value (CVaR_α) on the stress.

Remark 1 (Convexity in CVaR-based optimization) *Without any assumptions about the distribution of $g(\boldsymbol{d}, Z)$, it can be shown that CVaR can be written in the form of a convex optimization problem [3] as*

$$\text{CVaR}_\alpha[g(\boldsymbol{d}, Z)] = \min_{\gamma \in \mathbb{R}} \gamma + \frac{1}{1-\alpha} \mathbb{E}_\pi \left[g(\boldsymbol{d}, Z) - \gamma\right]^+, \tag{9}$$

where $\boldsymbol{d}$ is the given design and $[c]^+ = \max\{0, c\}$ and γ is an auxiliary variable. Using Equation (9)*, the formulation* (8) *can be reduced to an optimization problem involving only expectations as given by*

$$\begin{aligned} \min_{\gamma \in \mathbb{R},\, \boldsymbol{d} \in \mathcal{D}} \quad & \mathbb{E}_\pi[f(\boldsymbol{d}, Z)] \\ \textit{subject to} \quad & \gamma + \frac{1}{1-\alpha} \mathbb{E}_\pi \left[g(\boldsymbol{d}, Z) - \gamma\right]^+ \le t. \end{aligned} \tag{10}$$

The above formulation is a convex optimization problem when $g(\boldsymbol{d}, Z)$ is convex in $\boldsymbol{d}$ since $[\cdot]^+$ is a convex function and preserves the convexity of the limit state function. A similar convex reformulation also exists for the bPoF-based optimization formulation discussed later.

Another advantage of (10)*, as outlined in Ref. [3], is that the nonlinear CVaR constraint can be reformulated as a set of linear constraints if the density π of the random variable Z is independent of the design variable $\boldsymbol{d}$ and expectations can be estimated empirically. Specifically, consider a Monte Carlo estimate where $\boldsymbol{z}_i, i = 1, \ldots, m$ are m samples from probability density π. Then, using auxiliary variables $b_i, i = 1, \ldots, m$ to define $\boldsymbol{b} = \{b_1, \ldots, b_m\}$, we can reformulate* (10) *as*

$$\begin{aligned} \min_{\gamma \in \mathbb{R}, \boldsymbol{b} \in \mathbb{R}^m, \boldsymbol{d} \in \mathcal{D}} \quad & \mathbb{E}_\pi[f(\boldsymbol{d}, Z)] \\ \textit{subject to} \quad & \gamma + \frac{1}{m(1-\alpha)} \sum_{i=1}^{m} b_i \leq t, \\ & g(\boldsymbol{d}, z_i) - \gamma \leq b_i, i = 1, \ldots, m, \\ & b_i \geq 0, i = 1, \ldots, m. \end{aligned} \tag{11}$$

In the case where the density π depends upon $\boldsymbol{d}$, it is still possible to perform optimization. For example, [34] provide a simple sampling-based estimator for the gradient of CVaR, using these gradients in a stochastic gradient descent algorithm for optimizing a CVaR objective function. For the constrained CVaR problem, such gradient estimates can be used within stochastic optimization algorithms designed for the constrained setting, such as [35].

3. Optimization problem: CVaR as objective

CVaR naturally arises as a replacement for VaR in the constraint, but it can also be used as a cost function in the optimization problem formulation. For example, in PDE-constrained optimization, CVaR is used in the objective function [12, 13]. The optimization formulation becomes

$$\begin{aligned} \min_{\boldsymbol{d} \in \mathcal{D}} \quad & \mathrm{CVaR}_\alpha[g(\boldsymbol{d}, Z)] \\ \text{subject to} \quad & \mathrm{CVaR}_{\beta_T}[f(\boldsymbol{d}, Z)] \leq C_T, \end{aligned} \tag{12}$$

where α and β_T are the desired risk levels for g and f respectively, and C_T is a threshold on the quantity of interest f. This could be an interesting formulation for cases where it is easier to define a threshold on the quantity of interest than deciding a risk level for the limit state function. For example, if the quantity of interest is the cost of manufacturing a rocket engine, one can set a budget constraint and use the above formulation. The solution of this optimization formulation would result in the safest rocket engine design such that the expected budget is does not exceed the given budget.

4. Discussion about CVaR-based Optimization

A favorable property of CVaR is that it is a proper and coherent risk measure [36], and accordingly it is convex, monotonic, translation invariant and positive homogenous. From an optimization perspective, an important feature of CVaR is that it preserves convexity of the function it is applied to, i.e., the limit state function or cost function. CVaR also takes the magnitude of failure into account, which makes it more informative than PoF.

As noted in [37], CVaR estimators are less stable than estimators of VaR since rare, large magnitude tail samples can have large effect on the sample estimate. This is more prevalent when the distribution of the random quantity is fat-tailed. Thus, there is a need for more research to develop efficient algorithms for CVaR estimation. A drawback of CVaR is that it is non-smooth, and a direct CVaR-based optimization would require either nonsmooth optimization methods, for example variable-metric algorithms [38], or gradient-free methods, which lead to similar issues as faced by PoF-based optimization formulations. Also, smoothed approximations exist [12] which significantly improve optimization.

As noted in Remark 1, CVaR-based formulations can lead to well-behaved convex optimization problems. The formulation in (11) increases the dimensionality of the optimization problem from $n_d + 1$ to $n_d + m + 1$, where m is the number of MC samples, which poses an issue when the number of MC samples is large. However, the optimization formulation (11) has mostly linear constraints and can also be completely converted into a linear program by using a linear approximation for $g(\boldsymbol{d}, z_i)$ (following similar ideas as reliability index methods described in Section I). There are extremely efficient methods for finding solutions to linear programs even for high-dimensional problems.

C. Buffered-probability-of-failure-based Design Optimization

Buffered probability of failure was first introduced by Rockafellar and Royset [4] as an alternative to probability of failure. This section describes bPoF and the associated optimization problem formulations. As we will show, the use of bPoF and CVaR constraints are equivalent. In this case bPoF simply provides an alternative interpretation of the CVaR constraint that is, arguably, more natural for application dealing with constraints in terms of failure probability instead of constraints involving quantiles (i.e. VaR). There are, however, clear differences between CVaR and bPoF when considered as an objective function. This allows, for example, one to minimize bPoF, a conservative upper bound of PoF, subject to constraints on costs or other performance measures. This then provides a method for finding the optimally reliable design under performance constraints as opposed to the problem of finding the optimally performing design under reliability constraints.

1. Risk Measure: bPoF

The bPoF is an alternate measure of reliability which adds a buffer to the traditional PoF. The simplest way to introduce bPoF is to define it as the sum of probability of failure $\mathbb{P}(g(\boldsymbol{d}, Z) > t)$ and the probability of *near-failure* $\mathbb{P}\left(g(\boldsymbol{d}, Z) \in [\lambda, t]\right)$, where the near-failure region $[\lambda, t]$ is determined by the frequency and magnitude of tail events around t. While PoF is given by $\mathbb{P}(g(\boldsymbol{d}, Z) > t)$, we have that bPoF (under some simplifying assumptions*) is given by

$$\bar{p}_t(g(\boldsymbol{d}, Z)) = \mathbb{P}(g(\boldsymbol{d}, Z) \geq \lambda), \text{ where } \lambda \text{ is such that } \mathbb{E}_\pi[g(\boldsymbol{d}, Z) \mid g(\boldsymbol{d}, Z) \geq \lambda] = t. \tag{13}$$

In this definition, we see bPoF as the sum of failure probability $\mathbb{P}(g(\boldsymbol{d}, Z) > t)$ and *near-failure* probability $\mathbb{P}(g(\boldsymbol{d}, Z) \in [\lambda, t])$, where the size of the *near-failure* region $[\lambda, t]$ is determined by the magnitude and frequency of tail events *above and below* threshold t. Specifically, the right-hand-side condition of (13) says that λ is chosen so that the tail expectation beyond λ is equal to t. Thus, the magnitude of events in the tail will determine the value of λ and thus the size of the *near-failure* region.

The bPoF at a given design $\boldsymbol{d}$ can also be understood in the context of the CVaR measure as defined by

$$\bar{p}_t(g(\boldsymbol{d}, Z)) = \{1 - \alpha \mid \mathrm{CVaR}_\alpha[g(\boldsymbol{d}, Z)] = t\}. \tag{14}$$

This relationship can be viewed in the same way as that connecting VaR and PoF. Recall that $\mathrm{VaR}_\alpha[g(\boldsymbol{d}, Z)] = t \iff p_t(g(\boldsymbol{d}, Z)) = 1 - \alpha$. In this way, we also have that† $\mathrm{CVaR}_\alpha[g(\boldsymbol{d}, Z)] = t \iff \bar{p}_t(g(\boldsymbol{d}, Z)) = 1 - \alpha$. Bringing together (13) and (14), we can arrive at another definition of bPoF in terms of $\mathrm{VaR}_\alpha[g(\boldsymbol{d}, Z)]$ as

$$\bar{p}_t(g(\boldsymbol{d}, Z)) = \mathbb{P}[g(\boldsymbol{d}, Z) \geq \mathrm{VaR}_\alpha[g(\boldsymbol{d}, Z)]], \quad \text{where } \alpha \text{ is such that } \quad \mathrm{CVaR}_\alpha[g(\boldsymbol{d}, Z)] = t.$$

The above definition is the same as Equation (13) with $\lambda = \mathrm{VaR}_\alpha[g(\boldsymbol{d}, Z)]$ and is illustrated in Figure 3.

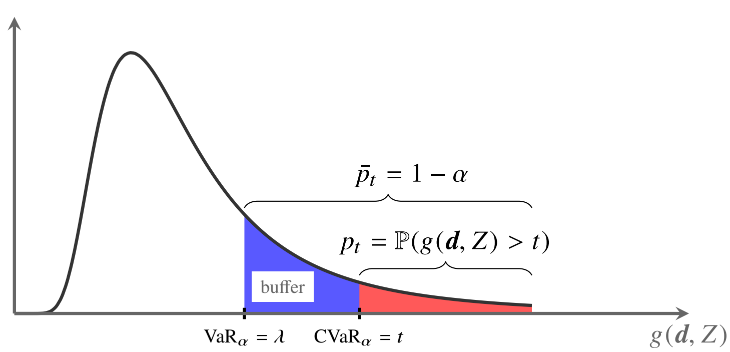


Fig. 3 Illustration for bPoF. The PoF equals $\mathbb{P}(g(\boldsymbol{d}, Z) > t)$ as shown by the area in red. For the same threshold t, bPoF equals $\bar{p}_t(g(\boldsymbol{d}, Z)) = 1 - \alpha$ the combined area in *red and blue*.

In general, we can see that for any design $\boldsymbol{d}$,

$$\bar{p}_t(g(\boldsymbol{d}, Z)) \geq p_t(g(\boldsymbol{d}, Z)). \tag{15}$$

* $g(\boldsymbol{d}, Z)$ is a continuous random variable
† assuming $0 \in (\mathbb{E}[g(\boldsymbol{d}, Z)], \sup g(\boldsymbol{d}, Z))$

Thus, the bPoF is a conservative estimate of the probability of failure for any design $\boldsymbol{d}$. However, we can see that the bPoF carries more information about failure than PoF since it takes into consideration the degree of failure. In other words, the bPoF takes the tail behavior of the $g(\boldsymbol{d}, Z)$ distribution into account. This property of bPoF is especially useful during optimization to differentiate between designs that have the same probability of failure.

Going back to interpretation (13), we see that the conservatism of bPoF comes from the data-dependent mechanism that selects the conservative threshold $\lambda < t$, which acts to establish a type of *buffer* zone. Specifically, the size of the buffer zone (λ, t) is determined by the magnitude of failure events above and below the original failure threshold of t. If realizations of $g(\boldsymbol{d}, Z)$ beyond t are very large (potentially catastrophic failures), λ will need to be smaller (making bPoF bigger) to drive the expectation beyond λ to t. Thus, the larger bPoF serves to account for not only the frequency of failure events, but also their magnitude. The bPoF also accounts for the frequency of *near-failure* events that have magnitude below, but very close to t. If there are a large number of near-failure events, bPoF will take this into account, since it will be included in the λ-tail which must have average equal to t. Algorithm 3 describes standard MC sampling for approximating bPoF.

Algorithm 3 Sampling-based estimation of bPoF.

Input: m i.i.d. samples $z_1, \ldots, z_m$ of random variable, design variable $\boldsymbol{d}$, failure threshold t, and limit state function $g(\boldsymbol{d}, Z)$.
Output: $\bar{p}_t(g(\boldsymbol{d}, Z))$.

Evaluate limit state function at the samples to get $g(\boldsymbol{d}, z_1), \ldots, g(\boldsymbol{d}, z_m)$.
Sort values of limit state function in descending order and relabel the samples so that

$$g(\boldsymbol{d}, z_1) > g(\boldsymbol{d}, z_2) > \ldots > g(\boldsymbol{d}, z_m).$$

$c = g(\boldsymbol{d}, z_1)$ ▷ Initialize CVaR estimate
$k = 1$
while c<t **do** ▷ Check if CVaR estimate equals threshold
    $k \leftarrow k + 1$
    $c = \frac{1}{k} \sum_{i=1}^{k} g(\boldsymbol{d}, z_k)$ ▷ Update CVaR estimate
end while
Estimate bPoF as $\bar{p}_t(g(\boldsymbol{d}, Z)) \approx \frac{k-1}{m}$ ▷ Estimate bPoF as $1 - \alpha$ when $c \approx t$

2. *Optimization problem: bPoF as constraint*

One of the downsides of the the CVaR-based optimization formulation given by (8) is its interpretability when compared to the widely-used PoF constraint. The bPoF concept, however, is easier to relate to PoF. While PoF gives the probability of exceeding the failure threshold, bPoF is the probability of exceeding the failure threshold *plus* the probability of being in a *near-failure* region.

In this sense, it can be more natural to consider the optimization problem with the PoF constraint replaced by the bPoF constraint as given by

$$\begin{aligned} \min_{\boldsymbol{d} \in \mathcal{D}} \quad & \mathbb{E}_{\pi}[f(\boldsymbol{d}, Z)] \\ \text{subject to} \quad & \bar{p}_t(g(\boldsymbol{d}, Z)) \leq 1 - \alpha. \end{aligned} \tag{16}$$

The power of this formulation is in its interpretation and it offers similar properties from an optimization standpoint when compared to the CVaR constraint formulation. Just as a PoF constraint is equivalent to a VaR constraint, it has been shown that a bPoF constraint is equivalent to a CVaR constraint. The optimization formulation given by (16) is equivalent[‡] to the CVaR-based optimization formulation given by (8) [39]. Therefore, this formulation simply provides another interpretation the of CVaR problem. This interpretation, however, is useful when considered in the context of the originally intended PoF reliability constraint. As noted in the previous section, bPoF can be interpreted as $P(g(\boldsymbol{d}, Z) > \lambda)$ with $\lambda < t$ and, loosely speaking, gives the probability that $g(\boldsymbol{d}, Z)$ will be *around* the zero threshold. In engineering applications, the exact failure threshold is often uncertain and chosen by a subject matter expert. Thus, it could be beneficial that bPoF measures failure probability with a data-dependent *soft* threshold. In other words,

[‡]under the condition that the optimal solution $\boldsymbol{d}^*$ does not have $\sup g(\boldsymbol{d}^*, Z) = 0$

bPoF can be viewed as a reliability constraint that is robust to uncertain or inexact choices of failure threshold. The conservatism of bPoF is data-driven and is a measure of reliability that simultaneously captures the frequency and magnitude of failure and near-failure events.

Remark 2 (Convexity in bPoF-based optimization) *Without any assumptions about the distribution of $g(\boldsymbol{d}, Z)$, it can be shown that bPoF*[§] *can be written in the form of a convex optimization problem, similar to CVaR, as [16, 40]*

$$\bar{p}_t(g(\boldsymbol{d}, Z)) = \min_{\lambda<t} \frac{\mathbb{E}_\pi\left[g(\boldsymbol{d}, Z) - \lambda\right]^+}{t - \lambda} = \min_{a\geq 0} \mathbb{E}_\pi\left[a(g(\boldsymbol{d}, Z) - t) + 1\right]^+ . \tag{17}$$

where $\boldsymbol{d}$ is the given design and $[c]^+ = \max\{0, c\}$ and λ, a is an auxiliary variable. The simplified right-most formulation comes from making the simple change of variable $a = \frac{1}{t-\lambda}$. The middle formulation using λ is important as the optimal λ^* is precisely the threshold from (13) *which provides $\mathbb{E}_\pi[g(\boldsymbol{d}, Z) \mid g(\boldsymbol{d}, Z) \geq \lambda^*] = t$.*

Using Equation (17)*, the formulation* (16) *can be reduced to an optimization problem involving only expectations as given by*

$$\begin{aligned} \min_{\lambda<t,\, \boldsymbol{d}\in\mathcal{D}} \quad & \mathbb{E}_\pi[f(\boldsymbol{d}, Z)] \\ \textit{subject to} \quad & \frac{\mathbb{E}_\pi\left[g(\boldsymbol{d}, Z) - \lambda\right]^+}{t - \lambda} \leq 1 - \alpha. \end{aligned} \tag{18}$$

One should note, however, that this can be reformulated, by a simple rearrangement of the constraint, to become equivalent to CVaR constrained problem (10)*. Thus, it is a convex problem and the same linearization trick can be performed as in* (11)*.*

3. Optimization problem: bPoF as objective

One of the novel uses of bPoF in risk-based optimization formulation is when it is used as an objective function. While its use as a constraint is equivalent to a CVaR constraint, the same can not be said about the case in which bPoF is used as an objective function. A bPoF objective provides us with an interesting optimization problem focused on optimal reliability subject to satisfaction of other design metrics.

Consider, for example, the following PoF minimization problem given by

$$\begin{aligned} \min_{\boldsymbol{d}\in\mathcal{D}} \quad & p_t(g(\boldsymbol{d}, Z)) \\ \text{subject to} \quad & \mathbb{E}_\pi[f(\boldsymbol{d}, Z)] \leq C_T. \end{aligned} \tag{19}$$

This optimization problem is normally not solved in RBDO due to the high level of difficulty caused by the objective function. As already mentioned, PoF is often nonconvex and discontinuous, making gradient calculation ill-posed or unstable. It is, however, a very desirable formulation if reliability is paramount. The formulation in (19) defines the situation where given our design specifications, characterized by $\mathbb{E}_\pi[f(\boldsymbol{d}, Z)] \leq C_T$, we desire the most reliable design achievable.

We can consider an alternative to the problem in (19) using a bPoF objective function as

$$\begin{aligned} \min_{\boldsymbol{d}\in\mathcal{D}} \quad & \bar{p}_t(g(\boldsymbol{d}, Z)) \\ \text{subject to} \quad & \mathbb{E}_\pi[f(\boldsymbol{d}, Z)] \leq C_T. \end{aligned} \tag{20}$$

Using Equation (17), the optimization problem in (20) can be reformulated in terms of expectations as

$$\begin{aligned} \min_{a\geq 0,\, \boldsymbol{d}\in\mathcal{D}} \quad & \mathbb{E}_\pi\left[a(g(\boldsymbol{d}, Z) - t) + 1\right]^+ \\ \text{subject to} \quad & \mathbb{E}_\pi[f(\boldsymbol{d}, Z)] \leq C_T. \end{aligned} \tag{21}$$

[§]assuming $t \in (\mathbb{E}[g(\boldsymbol{d}, Z)], \sup g(\boldsymbol{d}, Z))$

4. Discussion about bPoF-based optimization

There are several advantages of using the bPoF-based optimization problem described by Equation (16) as compared to the RBDO problem. Firstly, the bPoF-based optimization problem leads to a conservative design compared to RBDO and a feasible design for bPoF-based optimization problem also leads to a feasible design for RBDO. Secondly, the bPoF-based optimal design could be more desirable because it takes into account the magnitude of failure (or the tail of the distribution) that guards against more serious catastrophic failures. Thirdly, bPoF-based optimization problem preserves convexity if the underlying limit state function is convex (as compared to probability of failure which does not preserve convexity). This leads to well-behaved convex optimization problems even for the risk-based formulation and pushes us to pay more attention to devising limit state functions that are convex or nearly convex. Fourthly, under certain conditions, it is possible to calculate (quasi)-gradients for bPoF [41]. We note that non-smoothness can also be avoided by use of smoothed versions given by [42]. Lastly, if π is independent of $\boldsymbol{d}$, the same linear reformulation trick used with CVaR constraints (see (11)) can be used to transform the objective in (21) into a linear function with additional linear constraints and auxiliary variables offering similar advantages as noted in Section 4. Moreover, if the probability distribution of the limit state function, $g(\boldsymbol{d}, Z)$, could be known analytically (and be part of a special class of distributions), then the bPoF can also be derived analytically, see [17]. Statistical properties of empirical estimates of bPoF are discussed in [43].

IV. Numerical Experiment: Short Column Problem

We use the short column design problem that has been widely used in the RBDO community as a benchmark problem [2, 26] to compare some of the properties of PoF- and bPoF-based optimization formulations.

A. Short Column Problem Description

The problem consists of designing a short column with rectangular cross-section of dimensions w and h, subjected to uncertain loads (axial force F and bending moment M). The yield stress of the material, Y, is also considered to be uncertain. Table 1 shows the random variables used in the short column design. The correlation coefficient between F and M is 0.5. The random variables are $Z = [F,\ M,\ Y]^\top$. The design variables, $\boldsymbol{d} = [w,\ h]^\top$, are the length and width of the cross-section as shown in Table 2. The objective function is the cross-sectional area given by wh. Along with a failure threshold $t = 1$, the limit state function is defined as

$$g(\boldsymbol{d}, z) = \frac{4M}{wh^2Y} + \frac{F^2}{w^2h^2Y^2}. \tag{22}$$

Table 1 Random variables used in the short column application.

Random variable	Units	Distribution	Mean	Standard deviation
F	kN	Normal	500	100
M	kNm	Normal	2000	400
Y	MPa	Log-normal	5	0.5

Table 2 Design variables used in the short column application.

Design variable	Lower bound (m)	Upper bound (m)
w	5	15
h	15	25

B. Optimization Problem Formulations: RBDO and bPoF-constrained

The RBDO problem is given by

$$
\begin{aligned}
\min_{w,h} \quad & wh \\
\text{subject to} \quad & p_t(g(\boldsymbol{d}, Z)) \leq 1 - \alpha, \\
& \ell_w \leq w \leq u_w, \\
& \ell_h \leq h \leq u_h,
\end{aligned} \tag{23}
$$

where $(\ell_w, \ell_h, u_w, u_h)$ denote the lower and upper bounds on w and h as defined in Table 2.

We first show that the optimization problem with a bPoF constraint can be formulated as a convex optimization problem in this case. This will let us take advantage of convex optimization solvers and offer guarantees for the optimization problem. Let π denote the joint distribution of $Z = [F,\ M,\ Y]^\top$, with mean, variance and shape of the distributions given by Table 1. We use the formulation defined in (18) for the bPoF-based optimization problem for the short column design as

$$
\begin{aligned}
\min_{\lambda<t,w,h} \quad & wh \\
\text{subject to} \quad & \frac{\mathbb{E}_\pi\left[\frac{4M}{wh^2Y} + \frac{F^2}{w^2h^2Y^2} - \lambda\right]^+}{t - \lambda} \leq 1 - \alpha, \\
& \ell_w \leq w \leq u_w, \\
& \ell_h \leq h \leq u_h.
\end{aligned} \tag{24}
$$

To reformulate this as a more manageable convex optimization problem¶, we first rearrange the constraint to achieve the equivalent form‖,

$$
\begin{aligned}
\min_{\lambda,w,h} \quad & wh \\
\text{subject to} \quad & \lambda + \frac{1}{(1-\alpha)}\mathbb{E}_\pi\left[\frac{4M}{wh^2Y} + \frac{F^2}{w^2h^2Y^2} - \lambda\right]^+ \leq t, \\
& \ell_w \leq w \leq u_w, \\
& \ell_h \leq h \leq u_h.
\end{aligned} \tag{25}
$$

Next, we note that both w and h are nonnegative and thus we can make the change of variable $w = e^{x_1}, h = e^{x_2}$ with $x_1, x_2 \in \mathbb{R}$. Then the limit state function becomes $g(x_1, x_2, z) = \frac{4M}{Y}e^{-x_1-2x_2} + \frac{F^2}{Y^2}e^{-2x_1-2x_2}$ and objective function $wh = e^{x_1+x_2}$. These are both convex functions in the new decision variables (x_1, x_2). Thus, we can reformulate the problem as

$$
\begin{aligned}
\min_{\lambda,x_1,x_2} \quad & e^{x_1+x_2} \\
\text{subject to} \quad & \lambda + \frac{1}{(1-\alpha)}\mathbb{E}_\pi\left[\frac{4M}{Y}e^{-x_1-2x_2} + \frac{F^2}{Y^2}e^{-2x_1-2x_2} - \lambda\right]^+ \leq t, \\
& \ln \ell_w \leq x_1 \leq \ln u_w, \\
& \ln \ell_h \leq x_2 \leq \ln u_h.
\end{aligned} \tag{26}
$$

Furthermore, since the distribution of the random variables are independent of the design variables, we can empirically estimate the expectation in the constraint for any design by using a fixed set of m samples $\{z_1, \ldots, z_m\}$ that are sampled *a priori* from the distribution π. Finally, this gives us a convex sample-average-approximation (SAA) optimization

¶We were required to find a formulation that was recognized as convex by modeling language CVXpy with convex solver MOSEK

‖Note that this is in the same form as a CVaR constraint from (10).

problem for the given set of m samples as

$$\begin{aligned}
\min_{\lambda, x_1, x_2} \quad & e^{x_1+x_2} \\
\text{subject to} \quad & \lambda + \frac{1}{m(1-\alpha)} \sum_{i=1}^{m} \left[\frac{4M_i}{Y_i} e^{-x_1-2x_2} + \frac{F_i^2}{Y_i^2} e^{-2x_1-2x_2} - \lambda \right]^+ \leq t, \\
& \ln \ell_w \leq x_1 \leq \ln u_w \\
& \ln \ell_h \leq x_2 \leq \ln u_h .
\end{aligned} \tag{27}$$

Note that the constraint is convex since $[\cdot]^+$ is a convex function and preserves the convexity of the limit state function.

C. Experimental Comparison

We now compare the behavior of the RBDO formulation and the bPoF-constrained formulation. We solve the RBDO problem with various values of α using the gradient-free COBYLA optimizer. We use Algorithm 1 to estimate PoF in the RBDO problem by iteratively adding samples until the Monte Carlo error reaches below 1%. We solved the bPoF-constrained problem (Equation (27)) with various values of α, utilizing the convex optimization solver MOSEK with CVXpy as our modeling interface to the solver.

We begin by making the simple observation that, as hypothesized, PoF and bPoF are indeed natural counterparts with both measuring similar notions of failure risk. Figure 4(b) and (c) shows that both the formulations can lead to nearly identical frontiers of cross-section area for the optimal designs with similar PoF or bPoF, and they can be obtained by setting higher values for $1-\alpha$ in bPoF-based optimization as compared to PoF-based optimization. Each point in Figure 4(b) corresponds to an optimal design achieved by solving RBDO ("using PoF") or bPoF-constrained optimization for some α. The x-axis provides the estimated value of PoF of the design, estimated using a separate evaluation sample of size 5×10^6. The y-axis provides the cross-section area of the optimal design. Figure 4(c) is similar, except with the value of bPoF (estimated using 5×10^6 samples) as the x-axis. When comparing the solutions, we see that RBDO and bPoF formulations achieve nearly identical frontiers. This highlights the fact that bPoF is indeed a natural surrogate for PoF, achieving similar goals and controlling failure probability. However, an advantage of the bPoF-based formulation is that while both achieve similar design frontiers, we have a guarantee that the design given by the bPoF formulation is *globally optimal* for the given SAA problem in Equation (27). We have no such guarantee for the designs given by RBDO. Thus, even if similar designs are achieved by RBDO and bPoF-constrained optimization, we have additional guarantees about the quality of the bPoF-based design due to the underlying convexity.

Our second observation is the conservative nature of bPoF as compared to PoF. When formulated with identical levels of α in the constraint, the bPoF-based optimization achieves a more conservative design than RBDO. The conservative nature of the bPoF formulation is illustrated by Figure 4(a). Here, the x-axis provides the actual value of $1-\alpha$ used in the constraint to solve for the optimal design. We see that setting an upper bound of $1-\alpha$ on bPoF yields a more conservative design than the corresponding RBDO problem formulated with the same upper bound of $1-\alpha$ on PoF. This type of probabilistically derived conservativeness can be seen as desirable as discussed in Remark 3 below.

Remark 3 (Desirable conservativeness) *Let us take a closer look at the conservativeness induced by the bPoF-based optimization for the same desired reliability level as compared to PoF-based optimization (as shown in Figure 4(a)) and why this type of conservativeness could be desirable. There are other ways of introducing conservativeness, such as, safety factors, basis values, stricter reliability levels. Typically, using safety factors lead to overly conservative designs. When appropriate safety factor values are not used, the deterministic optimization setup could also potentially lead to unreliable designs. This is because converting to deterministic optimization using just safety factors (or basis values) to account for the uncertainty in the system does not take into account the distribution of the limit state function and lacks enough information to make good decisions. These well-known issues with safety-factor- and basis-values-based deterministic optimization formulations has progressively led us to consider risk-based optimization under uncertainty.*

One way to introduce conservativeness in risk-based optimization is by using lower values of $1-\alpha$*, which leads to stricter reliability constraints. However, this will just lead to more reliable designs than required without taking into account any information about the magnitude of failure. The bPof-based (or CVaR-based) optimization can be seen as a better way to induce conservativeness because they encode more information about the distribution of the limit state function by taking the magnitude of failure into account to arrive at the optimal design. This leads to a probabilistically informed way of getting to a conservative design, which could be seen as practically more desirable. Note that the*

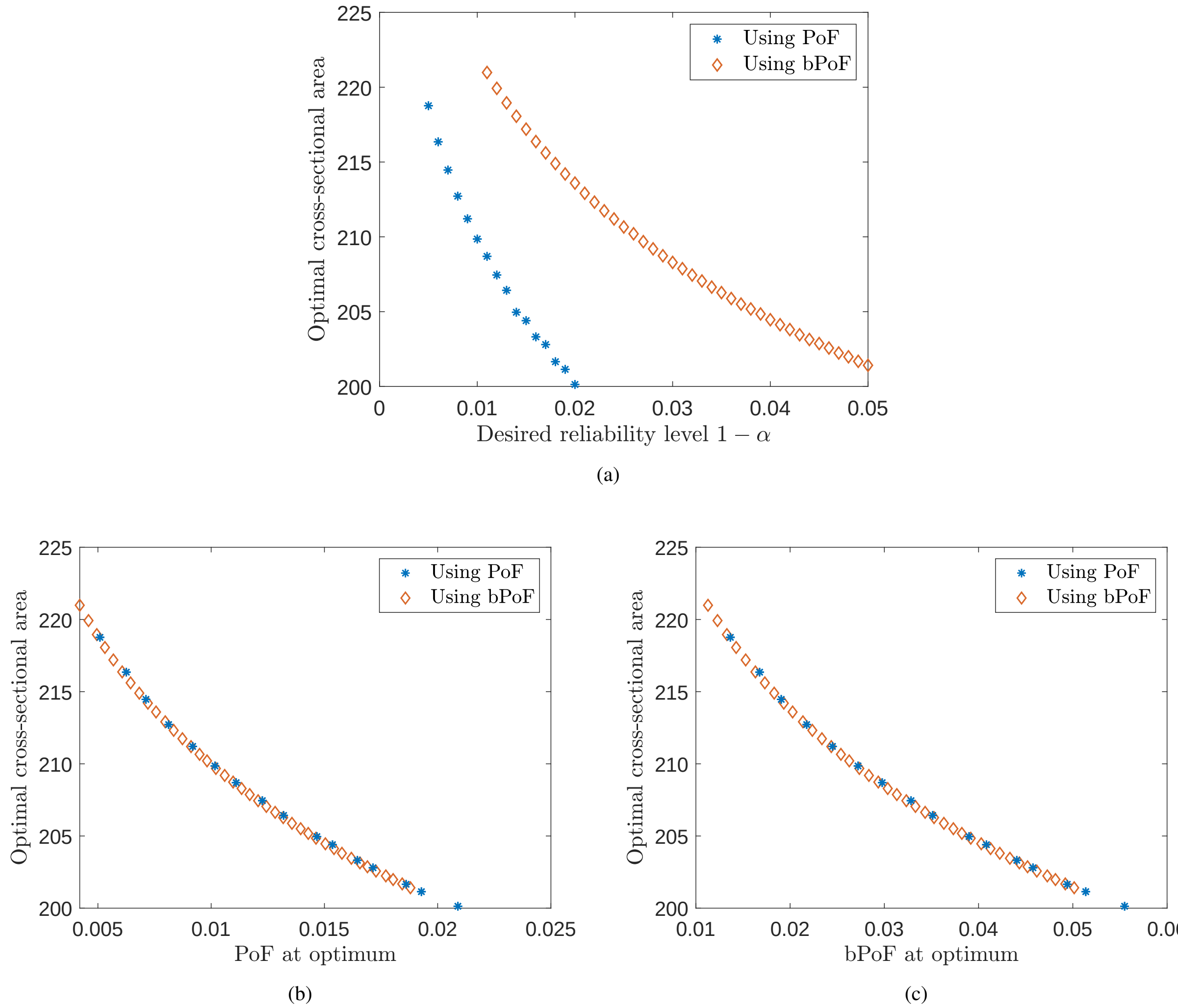


Fig. 4 Optimal cross-sectional area obtained for different levels of desired reliability

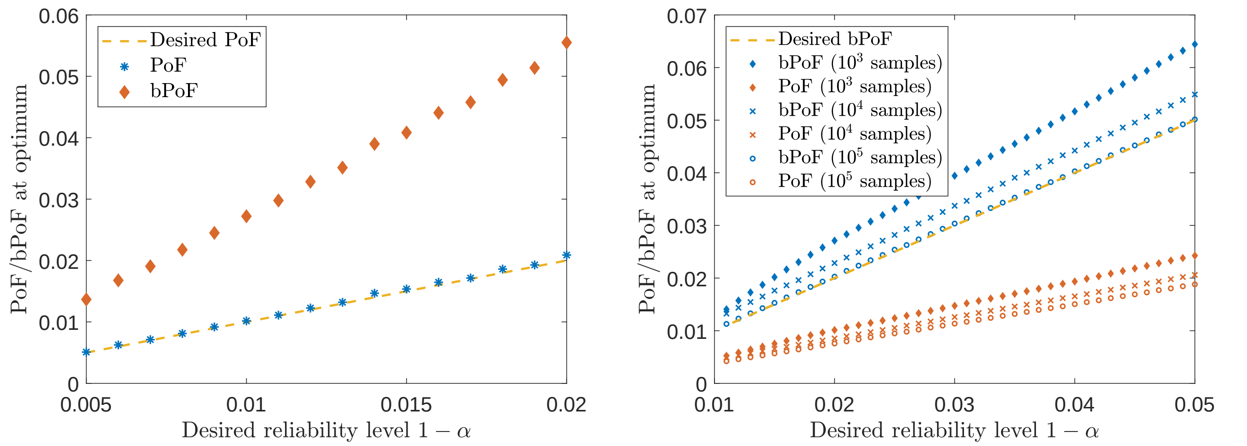
conservativeness introduced through bPoF-based optimization depends on the distribution of the limit state function as seen from Equations (13) *and* (15).

Figure 5 compares the desired reliability levels versus the estimated PoF/bPoF for the optimal designs obtained through PoF- and bPoF-based optimization. For these plots, we use 5×10^6 samples** to get accurate estimates of the PoF or bPoF at the optimum. Figure 5(a) shows the desired reliability level and the PoF/bPoF for the optimum design obtained using the RBDO problem. We can see that since the Monte Carlo error for PoF estimate in the RBDO problem was always ensured to be below 1%, the desired PoF and the PoF at the optimum overlap. The figure also shows the conservative property of bPoF for same desired reliability level.

A third key observation is also illustrated by Figure 5(b), which compares the results for bPoF-based optimization for different *a priori* sample sizes, i.e. the value of m in Equation (27). We make this comparison to analyze the effect of fixing the sample set for all optimization iterations before starting the optimization, which is required to obtain the convex optimization formulation shown in Equation (27). We can see that for lower sample sizes of 10^3 and 10^4, the bPoF at the optimum and the desired bPoF do not overlap reflecting inaccurate estimates of bPoF. However, it should be noted that the bPoF formulation is still surprisingly effective in controlling the PoF, even when sample size is small. In other words, even when small samples are used within the optimization, the conservative nature of bPoF yields a design

**These samples are not used in the optimization, but only to estimate at the optimal design after the optimization is completed.

with very small PoF. Additionally, it is important to highlight the fact that the convex optimization problem, even when formulated with a small sample size, is still very stable and can be solved reliably. One of the primary drawbacks of RBDO is the potential fragility of the optimization, particularly when sample sizes are small, where the estimates of PoF and/or gradients (if a gradient-based solver is used) are unstable and produce poor or inconsistent optimization results. The bPoF formulation does not suffer in the same way as illustrated here.



(a) RBDO (PoF-based optimization) (b) bPoF-based optimization

Fig. 5 Comparing PoF and bPoF estimates at the optimal design obtained through PoF- and bPoF-based optimization for different desired reliability levels.

V. Concluding Remarks

In this work, we compare and contrast three different risk-based optimization formulations using probability of failure, conditional value-at-risk, and buffered probability of failure. The purpose of this paper is to highlight the properties of the different risk measures and look into their potential advantages in the context of engineering design that can help outline some useful research directions for risk-based design optimization under uncertainty.

An advantage of bPoF and CVaR risk measure is the desirable conservativeness introduced by encoding extra information about the limit state function distribution in the form of the magnitude of failure. This highlights a way to get potentially better conservativeness in risk-based optimization by switching to a different optimization formulation. While PoF is an intuitive measure, it has drawbacks in that it does not consider the magnitude of failure events, but only their frequency. In addition, PoF can potentially mischaracterize the risk of *near-failure* events, which may have magnitude less than, but very near to the failure threshold. CVaR and bPoF provide alternative measures of failure risk that avoid such hard-threshold characterizations of failure events. CVaR provides a characterization of the failure risk which allows measurement of the expected magnitude of a selected portion of worst-case events. However, CVaR can be challenging to compare to PoF, since it does not provide a probabilistic characterization of failure risk relative to a failure threshold. bPoF provides a natural intuitive alternative to PoF that takes magnitude and frequency of failure and near-failure events into account.

From the computational perspective for risk-based optimization, bPoF and CVaR risk measures provide substantial benefits because they preserve convexity of the limit state function. When limit-state functions are convex w.r.t. the design variables, CVaR- and bPoF-based optimization problems can be formulated into convex optimization problems. Consequently, we can use convex optimizers to solve risk-based optimization problems and provide convergence guarantees. Using convex approximations for the limit state functions is one way of addressing non-convex problems.

The plethora of methods available for the RBDO problem and PoF estimation in engineering design reflects the ease of working with PoF. There needs to be more research efforts directed towards bPoF and CVaR estimation from an engineering design context. This will lead to flexibility in switching to appropriate and advantageous risk-based optimization formulations for designing safe engineering systems.

Acknowledgement

This work has been supported in part by the Air Force Office of Scientific Research (AFOSR) MURI on managing multiple information sources of multi-physics systems award numbers FA9550-15-1-0038 and FA9550-18-1-0023, and Air Force Center of Excellence on Multi-Fidelity Modeling of Rocket Combustor Dynamics award FA9550-17-1-0195.

References

[1] Yao, W., Chen, X., Luo, W., van Tooren, M., and Guo, J., "Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles," *Progress in Aerospace Sciences*, Vol. 47, No. 6, 2011, pp. 450–479.

[2] Aoues, Y., and Chateauneuf, A., "Benchmark study of numerical methods for reliability-based design optimization," *Structural and Multidisciplinary Optimization*, Vol. 41, No. 2, 2010, pp. 277–294.

[3] Rockafellar, R. T., Uryasev, S., et al., "Optimization of conditional value-at-risk," *Journal of Risk*, Vol. 2, 2000, pp. 21–42.

[4] Rockafellar, R. T., and Royset, J. O., "On buffered failure probability in design and optimization of structures," *Reliability Engineering & System Safety*, Vol. 95, No. 5, 2010, pp. 499–510.

[5] Rockafellar, R. T., and Uryasev, S., "Conditional value-at-risk for general loss distributions," *Journal of Banking & Finance*, Vol. 26, No. 7, 2002, pp. 1443–1471. doi:10.1016/S0378-4266(02)00271-6, URL `https://doi.org/10.1016/S0378-4266(02)00271-6`.

[6] Krokhmal, P., Palmquist, J., and Uryasev, S., "Portfolio optimization with conditional value-at-risk objective and constraints," *Journal of Risk*, Vol. 4, No. 2, 2002, pp. 11–27. doi:10.21314/JOR.2002.057, URL `http://dx.doi.org/10.21314/JOR.2002.057`.

[7] Mansini, R., Ogryczak, W., and Speranza, M. G., "Conditional value at risk and related linear programming models for portfolio optimization," *Annals of Operations Research*, Vol. 152, 2007, pp. 227–256. doi:10.1007/s10479-006-0142-4, URL `http://dx.doi.org/10.1007/s10479-006-0142-4`.

[8] Rockafellar, R. T., and Royset, J. O., "Engineering decisions under risk averseness," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, Vol. 1, No. 2, 2015, p. 04015003. doi:10.1061/AJRUA6.0000816, URL `http://dx.doi.org/10.1061/AJRUA6.0000816`.

[9] Zhang, W., Rahimian, H., and Bayraksan, G., "Decomposition algorithms for risk-averse multistage stochastic programs with application to water allocation under uncertainty," *INFORMS Journal on Computing*, Vol. 28, No. 3, 2016, pp. 385–404. doi:10.1287/ijoc.2015.0684, URL `https://doi.org/10.1287/ijoc.2015.0684`.

[10] Yang, H., and Gunzburger, M., "Algorithms and analyses for stochastic optimization for turbofan noise reduction using parallel reduced-order modeling," *Comput. Methods Appl. Mech. Engrg.*, Vol. 319, 2017, pp. 217–239. doi:10.1016/j.cma.2017.02.030, URL `https://doi.org/10.1016/j.cma.2017.02.030`.

[11] Royset, J. O., Bonfiglio, L., Vernengo, G., and Brizzolara, S., "Risk-adaptive set-based design and applications to shaping a hydrofoil," *Journal of Mechanical Design*, Vol. 139, No. 10, 2017, p. 101403. doi:10.1115/1.4037623, URL `http://dx.doi.org/10.1115/1.4037623`.

[12] Kouri, D. P., and Surowiec, T. M., "Risk-averse PDE-constrained optimization using the conditional value-at-risk," *SIAM Journal on Optimization*, Vol. 26, No. 1, 2016, pp. 365–396. doi:10.1137/140954556, URL `http://dx.doi.org/10.1137/140954556`.

[13] Zou, Z., Kouri, D. P., and Aquino, W., "A locally adapted reduced basis method for solving risk-averse PDE-constrained optimization problems," *2018 AIAA Non-Deterministic Approaches Conference, Kissimmee, Florida, AIAA SciTech Forum, (AIAA 2018-2174)*, 2018. doi:10.2514/6.2018-2174, URL `https://doi.org/10.2514/6.2018-2174`.

[14] Heinkenschloss, M., Kramer, B., Takhtaganov, T., and Willcox, K., "Conditional-value-at-risk estimation via reduced-order models," *SIAM/ASA Journal on Uncertainty Quantification*, Vol. 6, No. 4, 2018, pp. 1395–1423.

[15] Heinkenschloss, M., Kramer, B., and Takhtaganov, T., "Adaptive reduced-order model construction for conditional value-at-risk estimation," Tech. rep., Massachusetts Institute of Technology, 2019. ACDL TR-2019-02.

[16] Mafusalov, A., and Uryasev, S., "Buffered probability of exceedance: mathematical properties and optimization," *SIAM Journal on Optimization*, Vol. 28, No. 2, 2018, pp. 1077–1103.

[17] Norton, M., Khokhlov, V., and Uryasev, S., "Calculating CVaR and bPOE for common probability distributions with application to portfolio optimization and density estimation," *arXiv preprint arXiv:1811.11301*, 2018.

[18] Rubinstein, R. Y., and Kroese, D. P., *Simulation and the Monte Carlo method*, Vol. 10, John Wiley & Sons, 2016.

[19] Melchers, R., "Importance sampling in structural systems," *Structural Safety*, Vol. 6, No. 1, 1989, pp. 3–10.

[20] Owen, A. B., *Monte Carlo theory, methods and examples*, 2013.

[21] Li, J., Li, J., and Xiu, D., "An efficient surrogate-based method for computing rare failure probability," *Journal of Computational Physics*, Vol. 230, No. 24, 2011, pp. 8683–8697.

[22] Peherstorfer, B., Kramer, B., and Willcox, K., "Combining multiple surrogate models to accelerate failure probability estimation with expensive high-fidelity models," *Journal of Computational Physics*, Vol. 341, 2017, pp. 61–75. doi: https://doi.org/10.1016/j.jcp.2017.04.012.

[23] Peherstorfer, B., Kramer, B., and Willcox, K., "Multifidelity preconditioning of the cross-entropy method for rare event simulation and failure probability estimation," *SIAM/ASA Journal on Uncertainty Quantification*, Vol. 6, No. 2, 2018, pp. 737–761.

[24] Sobieszczanski-Sobieski, J., Morris, A., and van Tooren, M., *Multidisciplinary design optimization supported by knowledge based engineering*, John Wiley & Sons, 2015.

[25] Du, X., and Chen, W., "A most probable point-based method for efficient uncertainty analysis," *Journal of Design and Manufacturing Automation*, Vol. 4, No. 1, 2001, pp. 47–66.

[26] Bichon, B. J., Eldred, M. S., Mahadevan, S., and McFarland, J. M., "Efficient global surrogate modeling for reliability-based design optimization," *Journal of Mechanical Design*, Vol. 135, No. 1, 2013, p. 011009.

[27] Moustapha, M., and Sudret, B., "Surrogate-assisted reliability-based design optimization: a survey and a unified modular framework," *Structural and Multidisciplinary Optimization*, 2019, pp. 1–20.

[28] Marques, A., Lam, R., and Willcox, K., "Contour location via entropy reduction leveraging multiple information sources," *Advances in Neural Information Processing Systems*, 2018, pp. 5217–5227.

[29] Kramer, B., Marques, A., Peherstorfer, B., Villa, U., and Willcox, K., "Multifidelity probability estimation via fusion of estimators," *Journal of Computational Physics*, Vol. 392, 2019, pp. 385–402. URL `https://doi.org/10.1016/j.jcp.2019.04.071`.

[30] Chaudhuri, A., Marques, A. N., Lam, R., and Willcox, K. E., "Reusing Information for Multifidelity Active Learning in Reliability-Based Design Optimization," *AIAA Scitech 2019 Forum*, 2019, p. 1222.

[31] Uryasev, S., "Derivatives of probability functions and some applications," *Annals of Operations Research*, Vol. 56, No. 1, 1995, pp. 287–311.

[32] Royset, J., and Polak, E., "Extensions of stochastic optimization results to problems with system failure probability functions," *Journal of Optimization Theory and Applications*, Vol. 133, No. 1, 2007, pp. 1–18.

[33] Tretiakov, G. L., "Star-shaped approximation approach for stochastic programming problems with probability function," *Optimization*, Vol. 47, No. 3-4, 2000, pp. 303–317.

[34] Tamar, A., Glassner, Y., and Mannor, S., "Optimizing the CVaR via sampling," *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[35] Lan, G., and Zhou, Z., "Algorithms for stochastic optimization with expectation constraints," *arXiv preprint arXiv:1604.03887*, 2016.

[36] Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D., "Coherent measures of risk," *Mathematical finance*, Vol. 9, No. 3, 1999, pp. 203–228.

[37] Yamai, Y., Yoshiba, T., et al., "Comparative analyses of expected shortfall and value-at-risk: their estimation error, decomposition, and optimization," *Monetary and economic studies*, Vol. 20, No. 1, 2002, pp. 87–121.

[38] Uryasev, S., "New variable-metric algorithms for nondifferentiable optimization problems," *Journal of Optimization Theory and Applications*, Vol. 71, No. 2, 1991, pp. 359–388.

[39] Shang, D., Kuzmenko, V., and Uryasev, S., "Cash flow matching with risks controlled by buffered probability of exceedance and conditional value-at-risk," *Annals of Operations Research*, Vol. 260, No. 1-2, 2018, pp. 501–514.

[40] Norton, M., and Uryasev, S., "Maximization of auc and buffered auc in binary classification," *Mathematical Programming*, Vol. 174, No. 1-2, 2018, pp. 575–612.

[41] Zhang, T., Uryasev, S., and Guan, Y., "Derivatives and subderivatives of buffered probability of exceedance," *Operations Research Letters*, Vol. 47, No. 2, 2019, pp. 130–132.

[42] Kouri, D., "Higher-moment buffered probability," *Optimization Letters*, 2019, pp. 1–15.

[43] Mafusalov, A., Shapiro, A., and Uryasev, S., "Estimation and asymptotics for buffered probability of exceedance," *European Journal of Operational Research*, Vol. 270, No. 3, 2018, pp. 826–836.