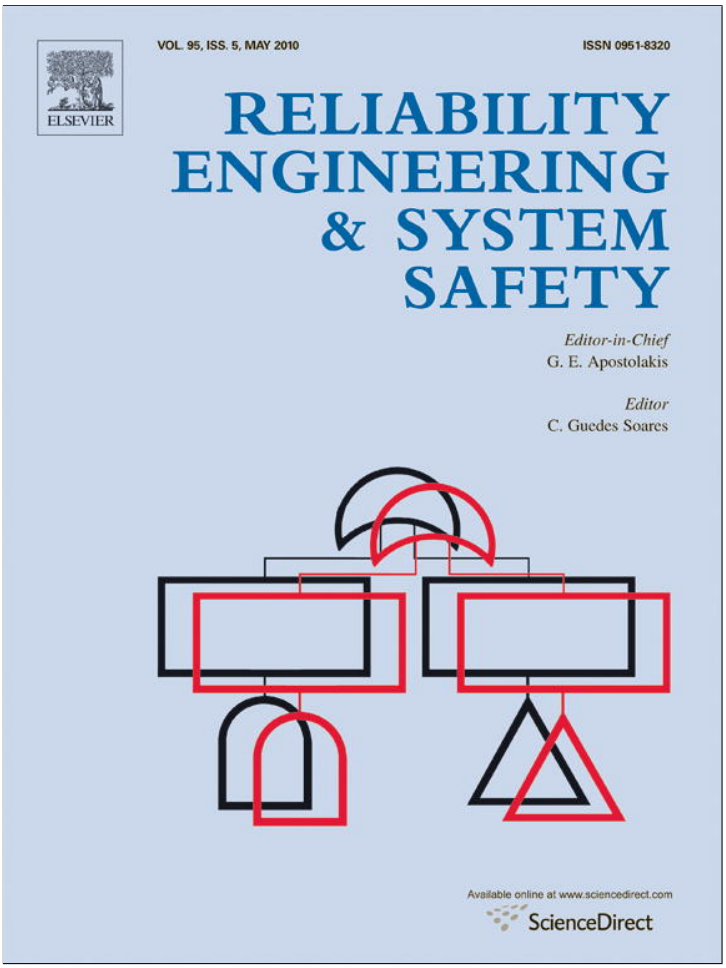

Reliability Engineering and System Safety 95 (2010) 499–510

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

On buffered failure probability in design and optimization of structures

R.T. Rockafellar [a,b], J.O. Royset [c,*]

[a] Department of Mathematics, University of Washington, Seattle, WA 98195-4350, USA
[b] Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611-6595, USA
[c] Operations Research Department, Naval Postgraduate School, Monterey, CA 93943, USA

ARTICLE INFO

Article history:
Received 26 June 2009
Received in revised form
19 December 2009
Accepted 7 January 2010
Available online 13 January 2010

Keywords:
Failure probability
Structural reliability
Reliability-based design optimization

ABSTRACT

In reliability engineering focused on the design and optimization of structures, the typical measure of reliability is the probability of failure of the structure or its individual components relative to specific limit states. However, the failure probability has troublesome properties that raise several theoretical, practical, and computational issues. This paper explains the seriousness of these issues in the context of design optimization and goes on to propose a new alternative measure, the buffered failure probability, which offers significant advantages. The buffered failure probability is handled with relative ease in design optimization problems, accounts for the degree of violation of a performance threshold, and is more conservative than the failure probability.

Published by Elsevier Ltd.

1. Introduction

Civil, mechanical, naval, and aeronautical structures such as bridges, building, offshore platforms, vehicle frames, ship hulls, and aircraft wings are subject to uncertain loads, environmental conditions, material properties, and geometry. It is widely recognized that these uncertainties must be accounted for in the design, maintenance, and retrofit of such structures. The theory of structural reliability, see, e.g. [5], provides an analytic framework for assessing the reliability of a structure as measured by its *failure probability* to be defined precisely below. The failure probability is widely promoted to designers and building code developers as a tool for assessing and comparing designs and has successfully been applied to many applications, see, e.g. [5]. While the failure probability is of significant importance, it also possesses troublesome properties that raise several theoretical, practical, and computational issues. In particular, these issues surface when the failure probability is used in design optimization of structures and may lead to poor numerical performance of standard nonlinear optimization algorithms such as SNOPT [9], LANCELOT [4], and NLPQL [28]. In this paper, we discuss these issues and propose an alternative measure of reliability that we call the *buffered failure probability*. The buffered failure probability is handled with relative ease in design optimization problems, accounts for the degree of violation of a performance threshold, and is more conservative than the failure probability.

The failure probability and the buffered failure probability are defined in terms a limit-state function $g(\mathbf{x},\mathbf{v})$ that is a function of a vector $\mathbf{x}=(x_1,x_2,\ldots,x_n)'$ of design variables (with prime $'$ denoting the transpose of a vector), which may represent member sizes, material type and quality, amount of steel reinforcement, and geometric layout selected by the designer, and a vector $\mathbf{v}=(v_1,v_2,\ldots,v_m)'$ of quantities, which may describe loads, environmental conditions, material properties, and other factors the designer cannot directly control. The quantities $\mathbf{v}$ are usually subject to uncertainty and their values are therefore not known a priori. The limit-state function represents the performance of the structure with respect to a specific criterion referred to as a limit state. As commonly done, we describe these quantities by random variables $\mathbf{V}=(V_1,V_2,\ldots,V_m)'$ with a joint probability distribution which is regarded as known, although it might need to be estimated empirically. To distinguish between the random variables and their realizations, we denote the former by capital letters and the latter by lower case letters. For a given design $\mathbf{x}$, $g(\mathbf{x},\mathbf{V})$ is a random variable describing the (random) performance of the structure. We refer to this random variable as the state of the structure.

By convention, $g(\mathbf{x},\mathbf{v})>0$ represents unsatisfactory performance of the structure with respect to the limit-state function and, consequently, the event $\{g(\mathbf{x},\mathbf{V})>0\}$ is the set of realizations of the random vector $\mathbf{V}$ corresponding to "failure." We refer to this set as the failure domain. We note that failure may not necessarily imply total collapse of the structure, but may simply mean the violation of a prespecified threshold for crack width, deflection, vibration, etc.

The current approach to structural reliability defines the failure probability of a structure with limit-state function $g(\mathbf{x},\mathbf{v})$

* Corresponding author. Tel.: + 1 831 656 2578; fax: +1 831 656 2595.
E-mail addresses: rtr@math.washington.edu (R.T. Rockafellar), joroyset@nps.edu (J.O. Royset).

0951-8320/$ - see front matter Published by Elsevier Ltd.
doi:10.1016/j.ress.2010.01.001

as the probability that the state of the structure takes on a positive value. As the failure probability depends on the design $\mathbf{x}$, we denote it by $p(\mathbf{x})$. That is,

$$p(\mathbf{x}) = P[g(\mathbf{x},\mathbf{V}) > 0] = \int \cdots \int I(g(\mathbf{x},\mathbf{v}) > 0) f_{\mathbf{V}}(\mathbf{v})\, dv_1 \ldots dv_m, \qquad (1)$$

where $f_{\mathbf{V}}(\mathbf{v})$ is the joint probability density function for $\mathbf{V}$ and $I(g(\mathbf{x},\mathbf{v}) > 0)$ is the indicator function defined to be one if $g(\mathbf{x},\mathbf{v}) > 0$ and zero otherwise.

Our definitions of unsatisfactory performance and the failure domain deviate in two minor ways from those of some other authors, see, e.g. [5]. First, we exclude the realizations $\mathbf{v}$ corresponding to $g(\mathbf{x},\mathbf{v}) = 0$ from the failure domain. Of course, if the probability of the event $\{g(\mathbf{x},\mathbf{V}) = 0\}$ is zero, as is typically the case when $\mathbf{V}$ are continuous random variables, then this exclusion does not change the failure probability. Our convention, however, facilitates easy transfer of the results in [20] to the framework of the present paper and therefore allows general forms of the limit-state function and a wide range of probability distributions. Second, while we define $g(\mathbf{x},\mathbf{v}) > 0$ as failure, some authors adopt the opposite convention where $g(\mathbf{x},\mathbf{v}) < 0$ represents failure. Obviously, it is trivial to switch between the two conventions by multiplying the limit-state function with -1. In this paper, we use the convention $g(\mathbf{x},\mathbf{v}) > 0$ to indicate failure as our derivations appear simpler in that case.

In Section 2, we discuss the properties of the failure probability in detail. Section 3 presents the buffered failure probability and shows that it is more conservative than the failure probability, accounts for unlikely but possible realizations of the state of the structure, and has significant computational advantages. Section 4 generalizes the discussion to structural systems with multiple limit-state functions. Section 5 illustrates the use of the buffered failure probability in design optimization of a truss structure and a vehicle frame. We end the paper with concluding remarks in Section 6.

2. Properties of the failure probability

While the definition of the failure probability is appealing due to its relative simplicity, it exhibits several undesirable properties resulting in significant theoretical and practical difficulties. We discuss these in turn next.

2.1. Simplistic characterization of structures as failed or safe

The current approach to structural reliability effectively characterizes a structure to be in only one of two possible states: failed, i.e., $g(\mathbf{x},\mathbf{v}) > 0$, or safe, i.e., $g(\mathbf{x},\mathbf{v}) \le 0$. Consequently, the "degree" of failure is not important. For example, the event $\{g(\mathbf{x},\mathbf{V}) = 100\}$ is no worse than the event $\{g(\mathbf{x},\mathbf{V}) = 0.01\}$ as they both are subsets of the failure domain and contribute to the failure probability. However, a designer would most likely prefer the event $\{g(\mathbf{x},\mathbf{V}) = 0.01\}$ as it represents only a minor violation of a threshold, possibly somewhat arbitrarily set. On the other hand, the event $\{g(\mathbf{x},\mathbf{V}) = 100\}$ may be catastrophic. The theory of structural reliability does not account for the designer's preference in this case. This preference may become important when a designer compares two candidate designs as the following example illustrates.

Example 1. Consider the design of a structure that is characterized by the limit-state function

$$g(\mathbf{x},\mathbf{v}) = 100\text{-}x_1 v_1\text{-}(1\text{-}x_1)v_2, \qquad (2)$$

where 100 is a deterministic load on the structure and x_1 is a design variable to be chosen by the designer. Only $x_1 = 0$ and 1 are allowable choices. Moreover, let V_1 be a normally distributed random variable with mean 150 and standard deviation 15 representing the strength of the structure when design $x_1 = 1$. When design $x_1 = 0$, the strength of the structure is V_2 which is a random variable with mean 150 and a triangular probability density function in the range [98.40, 175.8] with values near 175.8 being the most likely outcomes. Fig. 1 illustrates the probability density functions of $g(0,\mathbf{V})$ and $g(1,\mathbf{V})$. For both designs, the probability of failure is 4.29×10^{-4}. However, as seen from Fig. 2, which depicts the upper tails of the probability density functions in Fig. 1, the probability of an "extreme event" is substantial in case of design $x_1 = 1$, but nonexistent for design $x_1 = 0$. For example, the probability of the event $\{g(\mathbf{x},\mathbf{V}) > 2\}$ is 2.63×10^{-4} for design $x_1 = 1$ but for design $x_1 = 0$ that probability is of course zero. While this is obviously an artificial example, it illustrates that two designs with the same failure probability may have significantly different characteristics. If the designer only computes the failure probability, this difference may not be revealed.

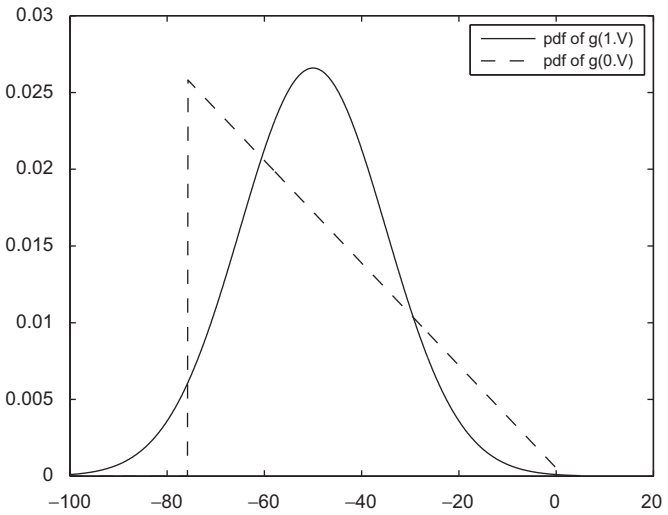

Fig. 1. Example 1: probability density functions (pdf) of $g(1,\mathbf{V})$ and $g(0,\mathbf{V})$.

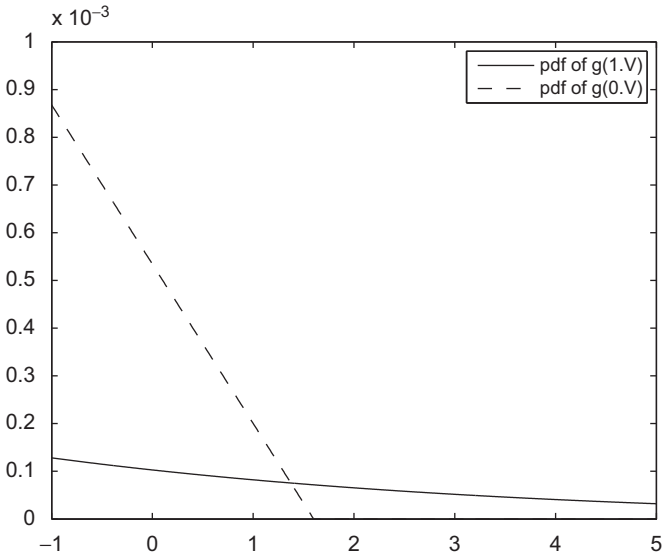

Fig. 2. Example 1: tails of probability density functions (pdf) of $g(1,\mathbf{V})$ and $g(0,\mathbf{V})$.

2.2. Inaccurate or computationally costly approximations

Since the uncertainty in a structure often needs to be characterized by many (hundreds of) random variables, the computation of the failure probability for a given design $\mathbf{x}$ requires the evaluation of a high-dimensional integral, see (1). As that evaluation is usually impossible to carry out analytically and computationally expensive to carry out by numerical integration, approximations based on Monte-Carlo simulation and geometric considerations are typically used.

2.2.1. Monte-Carlo simulation

For a given design $\mathbf{x}$, Monte-Carlo simulation estimates the failure probability $p(\mathbf{x})$ by generating N independent realizations $\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^N$ of the random vector $\mathbf{V}$ and computing the fraction of the realizations in the failure domain, i.e., the estimate of the failure probability

$$p_N(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N} I(g(\mathbf{x},\mathbf{v}^j) > 0). \tag{3}$$

The corresponding estimator is unbiased and, from the central limit theorem, we know that the standard deviation of the estimator decays proportional to $1/\sqrt{N}$, as $N\to\infty$. While this decay rate cannot be improved upon, the standard deviation of the estimator can often be much improved by the use of variance reduction techniques such as importance sampling, directional sampling, and Markov-chain Monte-Carlo sampling, see, e.g. [26]. Since the standard deviation decays only proportional to $1/\sqrt{N}$ and the effort required to compute $p_N(\mathbf{x})$ grows linearly in N, Monte Carlo simulation is usually computationally costly.

2.2.2. Geometric approximations

If the random vector $\mathbf{V}$ consists of independent standard normal random variables and the limit-state function is affine in $\mathbf{v}$, i.e., $g(\mathbf{x},\mathbf{v}) = \mathbf{a}(\mathbf{x})'\mathbf{v} + b(\mathbf{x})$ for some m-valued function $\mathbf{a}(\mathbf{x})$ and real-valued function $b(\mathbf{x})$, then the failure probability $p(\mathbf{x}) = \Phi(-\beta(\mathbf{x}))$ whenever $p(\mathbf{x}) \le 0.5$, see, e.g., [5, Chapters 4–6]. Here, $\Phi(\cdot)$ is the cumulative distribution function of a standard normal random variable and $\beta(\mathbf{x})$ is the shortest distance from the origin in $\mathbb{R}^m$ (i.e., the space of realizations of $\mathbf{V}$) to the surface $\{\mathbf{v}|g(\mathbf{x},\mathbf{v}) = 0\}$, see Fig. 3, where $g_1(\mathbf{x},\mathbf{v})$ is an example of an affine limit-state function. We refer to $\beta(\mathbf{x})$ as the reliability index of design $\mathbf{x}$. It can be shown that in this case

$$\beta(\mathbf{x}) = -b(\mathbf{x})/\|\mathbf{a}(\mathbf{x})\|. \tag{4}$$

If $g(\mathbf{x},\mathbf{v})$ is not affine, see, e.g., $g_2(\mathbf{x},\mathbf{v})$ in Fig. 3, then $\Phi(-\beta(\mathbf{x}))$ is an approximation of the failure probability. In this case, there is no explicit expression for $\beta(\mathbf{x})$ and it must be computed by solving the optimization problem

$$\beta(\mathbf{x}) = \min_{\mathbf{v}} \|\mathbf{v}\| \quad \text{s.t.} \quad g(\mathbf{x},\mathbf{v}) \ge 0. \tag{5}$$

There is empirical evidence that the approximation $\Phi(-\beta(\mathbf{x}))$ of the failure probability is quite accurate on classes of applications arising in structural engineering; see for example [36] and references therein. However, the approach may also lead to inaccuracy as discussed below.

When $g(\mathbf{x},\mathbf{v})$ is not concave[1] in $\mathbf{v}$, this optimization problem may have points satisfying the Karush–Kuhn–Tucker (KKT) first-order necessary conditions for a local minimum but that are not global minima. For example, limit-state function $g_3(\mathbf{x},\mathbf{v})$ in Fig. 3 results in a line $\{\mathbf{v}|g_3(\mathbf{x},\mathbf{v}) = 0\}$ with many points that are locally, but not globally, the closest point to the origin. Since standard nonlinear optimization algorithms such as SNOPT [9], LANCELOT [4], and NLPQL [28] only guarantee convergence to such a KKT point, it may be difficult to compute the globally optimal solution of (5) in this situation, let alone prove that an obtained point is globally optimal. The same holds for algorithms specialized for solving (5) such as the iHLRF algorithm [13]. Hence, $\beta(\mathbf{x})$ could be significantly overestimated, thereby leaving serious design risks undetected. For example, a standard nonlinear programming algorithm may return the same value for the three limit-state functions in Fig. 3 when applied to (5). The value would be correct for $g_1(\mathbf{x},\mathbf{v})$ and $g_2(\mathbf{x},\mathbf{v})$, but severely overestimate the reliability index for $g_3(\mathbf{x},\mathbf{v})$. Even if the global minimum is found in (5), we see from Fig. 3 that $\Phi(-\beta(\mathbf{x}))$ may overestimate $p(\mathbf{x})$, as in the case of $g_2(\mathbf{x},\mathbf{v})$, or underestimate it as in the case of $g_3(\mathbf{x},\mathbf{v})$. In general, it is difficult to know how close $\Phi(-\beta(\mathbf{x}))$ is to $p(\mathbf{x})$.

In practice, $\mathbf{V}$ is essentially never a vector of independent standard normal random variables. Hence, to apply the above approximation one typically needs to carry out a probability transformation, see, e.g., [5, Chapter 7]. Random vectors governed by distributions such as the multivariate normal (possibly with correlation) and lognormal distributions can be transformed into a standard normal vector using a smooth bijective mapping. Other transformations can also be carried out at least approximately. A transformation can make the limit-state function highly nonlinear and nonconcave as function of the independent standard normal random variables, which makes it difficult to determine the global minimum of (5).

The method of estimating $p(\mathbf{x})$ by $\Phi(-\beta(\mathbf{x}))$ is referred to as the first-order reliability method as it effectively linearizes a transformed limit-state function. An extension of this method is the second-order reliability method where the transformed limit-state function is approximated by a quadratic function, see [5, Chapter 6]. However, the second-order reliability method suffers from the same difficulties as the first-order method, though its accuracy may be better. An alternative method is to attempt, after a transformation to independent standard normal random variables, to determine the largest ball in $\mathbb{R}^m$, centered at the origin, with $g(\mathbf{x},\mathbf{v}) \le 0$ for all $\mathbf{v}$ in the ball. Using the chi-square distribution, this leads to an upper bound on the failure probability $p(\mathbf{x})$. However, the bound is usually overly conservative and of little practical use.

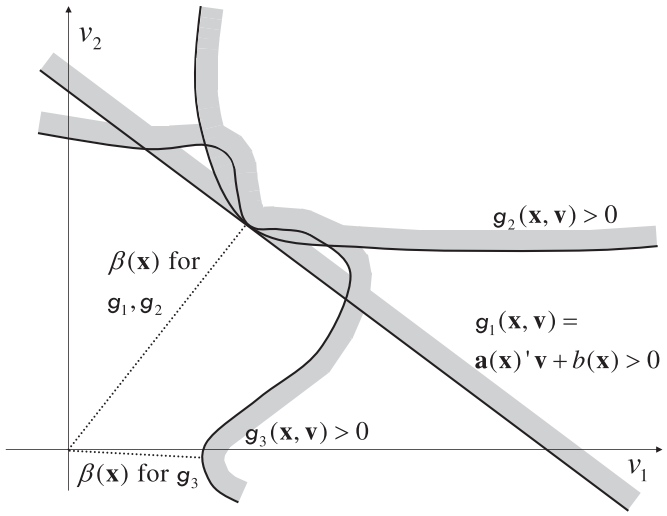

Fig. 3. Reliability indices $\beta(\mathbf{x})$ for three limit-state functions and a given design $\mathbf{x}$. The shaded areas indicate failure domains.

[1] See, e.g. [2,3] for definitions of concavity and convexity.

2.3. Poorly behaving sensitivities of failure probability and its approximations

In sensitivity analysis and design optimization, we examine the effect on the failure probability (or its approximation) of infinitesimal changes in the design. Hence, differentiability of the failure probability and its approximations with respect to design $\mathbf{x}$ as well as computable formulae for the corresponding gradient become important. Specifically, standard nonlinear optimization algorithms require all functions in an optimization problem to be continuously differentiable. If this condition is not satisfied, the algorithms may break down without returning an optimized design.

2.3.1. Gradient of the failure probability

The issue of differentiability of the failure probability is nontrivial as the integrand in (1) is not differentiable (The indicator function makes a jump from 1 to 0 as the condition $g(\mathbf{x},\mathbf{v})>0$ goes from being satisfied to not satisfied.) Hence, we cannot simply compute the derivative of an integral by integrating the derivative of the integrand which is allowed under weak assumptions when the integrand is differentiable.

Despite this situation, the failure probability is actually continuously differentiable with respect to the design $\mathbf{x}$ under rather general conditions when the failure domain is bounded and the limit-state function is continuously differentiable with respect to the design [33]. However, the gradient formula in [33] is difficult to use in estimation because it may involve surface integrals. In [14] (see also [15]), an integral transformation is presented, which, when it exists, leads to a simple formula for the gradient of the failure probability. However, it is not clear under what conditions the transformation exists. As in [33], the study [32] assumes that the failure domain is bounded. With this restriction as well as the assumption that the failure domain is "star-shaped," a formula for the gradient of the failure probability involving integration over a simplex is derived. In principle, this integral can be evaluated by Monte Carlo simulation. However, to the authors' knowledge, there is no computational experience with estimation of failure probabilities for highly reliable mechanical structures using this formula.

In [5, Section 9.2], with generalizations and proofs in [24], we find convenient expressions for the gradient of the failure probability under similar assumptions to those in [32]. The expressions can be estimated using Monte Carlo simulation with good accuracy at moderate computational expense when the star-shaped assumption is satisfied and the number of random variables is moderate. However, it becomes increasingly costly to estimate the expression using Monte Carlo simulation when the number of random variables grows. Moreover, in practice, it is difficult to verify the star-shape assumption. An alternative formula for the gradient of the failure probability is presented in [23,22] that can also be estimated using Monte Carlo simulation. However, the formula relies on the implicit function theorem applied to the equation $g(\mathbf{x},\mathbf{v})=0$ that may not always be applicable.

2.3.2. Gradient of the reliability index

As described in Section 2.2, the failure probability $p(\mathbf{x})$ can rarely be computed exactly and the approximation $\Phi(-\beta(\mathbf{x}))$ is often used, where the reliability index $\beta(\mathbf{x})$ is defined in (5). Since the cumulative distribution function $\Phi(\cdot)$ is continuously differentiable, differentiability of this approximation depends on the properties of $\beta(\mathbf{x})$. We find expressions for the gradient of $\beta(\mathbf{x})$ in [5, Chapter 8], but those cannot hold for all $\mathbf{x}$ as the following simple example illustrates.

Example 2. Consider the limit-state function

$$g(\mathbf{x},\mathbf{v})=\frac{v_1^2}{x_1^2}+v_2^2-1, \tag{6}$$

let V_1 and V_2 be independent standard normal random variables, and let $x_1>0$ be a design variable, see Fig. 4. As $\beta(\mathbf{x})$ is defined as the distance to the closest point on the surface $\{\mathbf{v}|g(\mathbf{x},\mathbf{v})=0\}$, see (5), we find that $\beta(\mathbf{x})=x_1$ if $0<x_1<1$ and 1 otherwise. Hence, $\partial\beta(\mathbf{x})/\partial x_1=1$ if $0<x_1<1$, $\partial\beta(\mathbf{x})/\partial x_1=0$ if $x_1>1$, and the derivative is not defined when $x_1=1$, see Fig. 5. As we see from this figure, $\beta(\mathbf{x})$ is not continuously differentiable and the derivative at $x_1=1$ is not defined.

As Example 2 illustrates, $\beta(\mathbf{x})$ may not be continuously differentiable and, hence, standard nonlinear optimization algorithms may stall at points that are not KKT points when applied to design optimization models involving $\beta(\mathbf{x})$.

In view of the above discussion, we see that the differentiability of the failure probability as well as the existence of tractable formulae for its gradient rely on assumptions that may not hold and that are difficult to verify in practice. Moreover, the frequently used reliability index provides an approximation of the failure probability $\Phi(-\beta(\mathbf{x}))$ that may not be continuously differentiable. Hence, even if the limit-state function is a continuously differentiable function in the design variables, the failure probability and $\Phi(-\beta(\mathbf{x}))$ may not be.

2.4. Lack of convexity of the failure probability

As stated above, standard nonlinear optimization algorithms typically only guarantee convergence to a KKT point. However, if a design optimization problem has a convex objective function, which we would like to minimize, and the constraints form a convex feasible region, then a KKT point must be a global optimal design for the problem. Absent convexity, it may be difficult to compute a globally optimal design, let alone prove that an

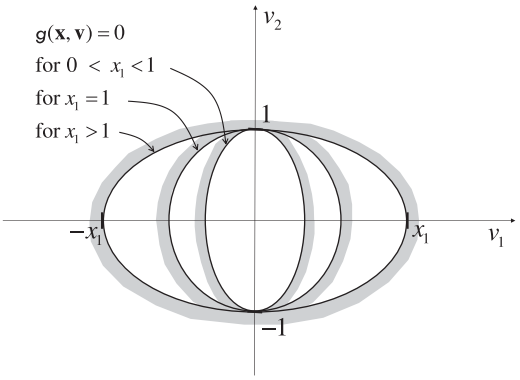

Fig. 4. Example 2. Equation $g(\mathbf{x},\mathbf{v})=v_1^2/x_1^2+v_2^2-1=0$ illustrated for three different values of x_1. The shaded areas indicate the failure domain.

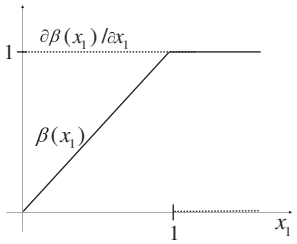

Fig. 5. Example 2. Illustration of reliability index $\beta(\mathbf{x})$ (solid line) and its derivative (dashed line).

obtained design is globally optimal. Therefore we would like to formulate convex design optimization models if possible. We refer to [3] for an introduction to convex optimization.

From this discussion we conclude that the convexity of the failure probability $p(\mathbf{x})$ would be valuable when solving a design optimization problem. Unfortunately, it is unknown whether $p(\mathbf{x})$ is convex even if $g(\mathbf{x},\mathbf{v})$ is convex in $\mathbf{x}$ for all $\mathbf{v}$. Hence, $p(\mathbf{x})$ does not "preserve" convexity as a convex limit-state function may result in a nonconvex failure probability. The same situation holds when the reliability index $\beta(\mathbf{x})$ is used to approximate the failure probability. For this reason, we expect that design optimization problems involving the failure probability or the reliability index may have many local minima that are not globally optimal. Standard nonlinear optimization algorithms are unlikely to find the globally optimal design and may return, at best, locally optimal designs. Consequently, it may be necessary to apply computationally expensive global optimization algorithms, see, e.g. [10].

3. Buffered failure probability

As reviewed in Section 2, the failure probability has several troublesome properties. In this section, we discuss an alternative probability, which we call the *buffered* failure probability, that has several advantages over the failure probability. The buffered failure probability relates to the conditional value-at-risk [19,20], which is now widely used in the area of financial engineering to assess investment portfolios. The tutorial paper [18] provides an overview including relation to safety margins and potential replacements for failure probability constraints. However, *buffered failure probability* is directly introduced and explained here for the first time.

3.1. Definition

We first recall that for any probability level α, the α-quantile of the distribution of a random variable is the value of the inverse of the corresponding cumulative distribution function at α. For simplicity in presentation, we assume here and throughout this paper that the cumulative distribution function of $g(\mathbf{x},\mathbf{V})$ is continuous and strictly increasing for all $\mathbf{x}$. For definitions which serves to fully generalize beyond this case, we refer to [20]. We consider especially the random variable $g(\mathbf{x},\mathbf{V})$ for a given design $\mathbf{x}$ and denote the α-quantile of $g(\mathbf{x},\mathbf{V})$ by $q_\alpha(\mathbf{x})$. As indicated by the notation, $q_\alpha(\mathbf{x})$ depends on the design $\mathbf{x}$ as the probability distribution of $g(\mathbf{x},\mathbf{V})$ changes with $\mathbf{x}$. Figs. 6 and 7 illustrate $q_\alpha(\mathbf{x})$ for the case when $g(\mathbf{x},\mathbf{V})$ is normally distributed with mean -1 and standard deviation 1. Fig. 6 shows the cumulative distribution function of $g(\mathbf{x},\mathbf{V})$ in this case and quantiles corresponding to probability levels $\alpha=0.60$ and $\alpha_0=0.84$. Fig. 7 illustrates the same information using the probability density function of $g(\mathbf{x},\mathbf{V})$ and, hence, probabilities correspond to areas under that function. In view of Figs. 6 and 7 and (1), we find that the failure probability is equal to one minus the probability level that results in the quantile being zero. For example, in Fig. 6 we find that $\alpha_0=0.84$ gives $q_{\alpha_0}(\mathbf{x})=0$. Hence, $p(\mathbf{x})=1\text{-}\alpha_0=1\text{-}0.84=0.16$.

Before we define the buffered failure probability, we introduce a quantity that is closely related to the quantile. For any probability level α, we define the α- superquantile as

$$\overline{q}_\alpha(\mathbf{x})=E[g(\mathbf{x},\mathbf{V})|g(\mathbf{x},\mathbf{V})\geq q_\alpha(\mathbf{x})], \tag{7}$$

where the vertical bar indicates a conditional expectation. That is, the α- superquantile is the average value of $g(\mathbf{x},\mathbf{V})$, conditional on the event that $g(\mathbf{x},\mathbf{V})$ is no less than the α- quantile. This quantity

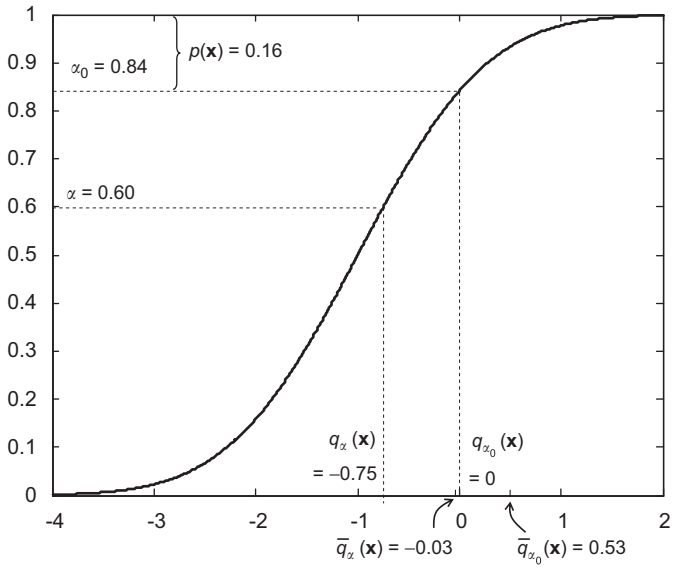

Fig. 6. Cumulative distribution function (cdf) of $g(\mathbf{x},\mathbf{V})$ with examples of α-quantile $q_\alpha(\mathbf{x})$ and α- superquantile $\overline{q}_\alpha(\mathbf{x})$ when normally distributed with mean -1 and standard deviation 1.

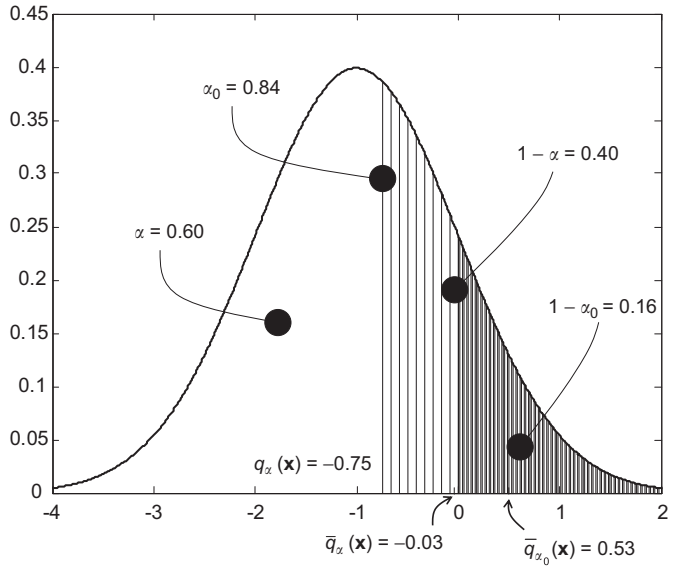

Fig. 7. Probability density function (pdf) of $g(\mathbf{x},\mathbf{V})$ with examples of α- quantile $q_\alpha(\mathbf{x})$ and α- superquantile $\overline{q}_\alpha(\mathbf{x})$ when normally distributed with mean -1 and standard deviation 1.

is called conditional value-at-risk in financial engineering, but we here propose and adopt the application-independent name *superquantile*. Figs. 6 and 7 illustrate the superquantiles of $g(\mathbf{x},\mathbf{V})$ for probability levels $\alpha=0.60$ and $\alpha_0=0.84$. Since $g(\mathbf{x},\mathbf{V})$ is normally distributed, it is trivial to compute the superquantiles using the well-known conditional expectation formula (see, e.g. [35]),

$$\overline{q}_\alpha(\mathbf{x})=\mu+\frac{\sigma\phi(q_\alpha)}{1\text{-}\alpha}, \tag{8}$$

for a normally distributed $g(\mathbf{x},\mathbf{V})$ with mean μ, standard deviation σ, and truncation level q_α, where $\phi(\cdot)$ is the standard normal probability density function and q_α is the α- quantile of the standard normal distribution. When $g(\mathbf{x},\mathbf{V})$ is not normally distributed, the calculation of the superquantile appears much

more difficult. As seen in the next subsection, however, it can be computed in a remarkably efficient manner.

Fig. 7 highlights the definition of a superquantile as a conditional expectation. As seen for probability level $\alpha = 0.60$, the corresponding quantile is -0.75. The corresponding superquantile is, roughly speaking, the value that splits the interval $[-0.75, \infty)$ into two "balancing" parts. The area under the probability density function between -0.75 and the value (the lightly shaded area in Fig. 7) "balances" the area under the function above the value (the heavily shaded area). In this case, that value is -0.03 as computed by (8). Similarly, for probability level 0.84, the corresponding quantile is 0 and the corresponding superquantile is 0.53. That is, the area under the probability density function between 0 and 0.53 "balances" the area under the function above 0.53.

We note that in general $q_\alpha(\mathbf{x}) \le \overline{q}_\alpha(\mathbf{x})$ for any probability level α and design $\mathbf{x}$. In [18] we also find the following equivalent formula for the superquantile:

$$\overline{q}_\alpha(\mathbf{x}) = \frac{1}{1\text{-}\alpha}\int_\alpha^1 q_{\alpha'}(\mathbf{x})\, d\alpha'. \tag{9}$$

We do not repeat the derivation of this expression here, but note that the expression essentially averages the quantiles for probability levels larger than α.

We now define the buffered failure probability $\overline{p}(\mathbf{x})$ to be equal to 1-α where α is selected such that the superquantile

$$\overline{q}_\alpha(\mathbf{x}) = 0. \tag{10}$$

That is,

$$\overline{p}(\mathbf{x}) = P[g(\mathbf{x},\mathbf{V}) \ge q_\alpha(\mathbf{x})], \tag{11}$$

where α is selected such that (10) holds. Hence, $\overline{q}_{1\text{-}\overline{p}(\mathbf{x})}(\mathbf{x}) = 0$. We see from Figs. 6 and 7 that the probability levels $\alpha = 0.60$, which led to $\overline{q}_\alpha(\mathbf{x}) =$ -0.03, and $\alpha_0 = 0.84$, which led to $\overline{q}_{\alpha_0}(\mathbf{x}) = 0.53$, are slightly too small and much to large, respectively, to result in a corresponding superquantile of zero. However, it is easy to find by trial-and-error and (8) that a probability level $\alpha = 0.62$ results in a quantile of -0.70 and a superquantile of approximately zero as illustrated in Figs. 8 and 9. (We present a much easier way than trial-and-error below for computing the superquantile.) By definition, see (11), the buffer probability is then $1\text{-}\alpha = 1\text{-}0.62 = 0.38$, which is somewhat larger than the failure probability of 0.16.

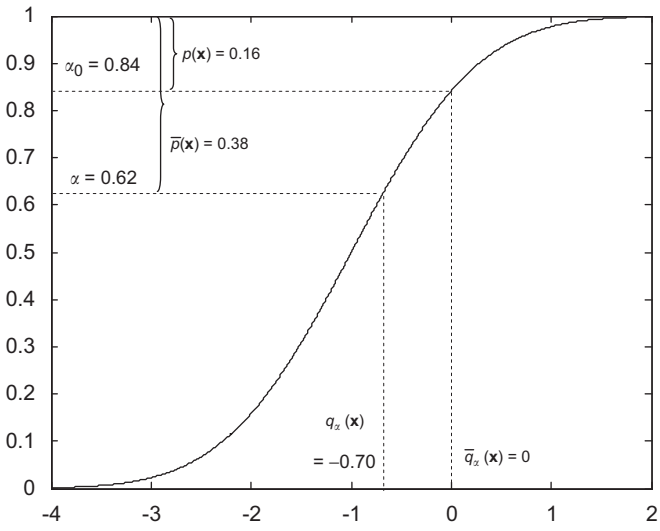

Fig. 8. Cumulative distribution function (cdf) of $g(\mathbf{x},\mathbf{V})$, as in Fig. 6, with α selected such that the α- superquantile $\overline{q}_\alpha(\mathbf{x}) = 0$. Illustration of the buffered failure probability $\overline{p}(\mathbf{x})$ and the failure probability $p(\mathbf{x})$.

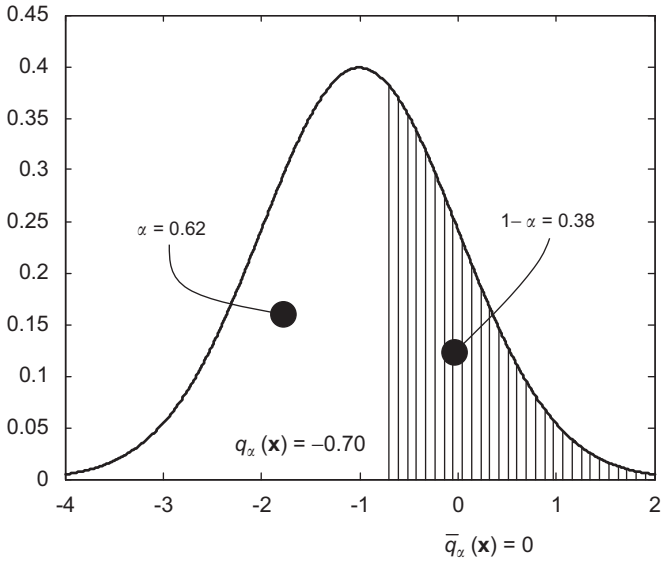

Fig. 9. Probability density function (pdf) of $g(\mathbf{x},\mathbf{V})$, as in Fig. 7, with α selected such that the α- superquantile $\overline{q}_\alpha(\mathbf{x}) = 0$. Illustration of the buffered failure probability $\overline{p}(\mathbf{x})$ and the failure probability $p(\mathbf{x})$.

In general, we find that

$$p(\mathbf{x}) \le \overline{p}(\mathbf{x}) \tag{12}$$

for any $\mathbf{x}$, see [19,20,18]. Hence, the buffered failure probability is a conservative estimate of the failure probability for any design $\mathbf{x}$. As we see below, the degree of overestimation is usually modest. We stress, however, that the buffered failure probability carries more information about the design than the failure probability as it includes information about the upper tail of $g(\mathbf{x},\mathbf{V})$. Hence, for designs where the probability of $g(\mathbf{x},\mathbf{V})$ taking on values substantially above zero is relatively large, the buffered failure probability tends to be somewhat larger than the failure probability. In contrast, if the probability of $g(\mathbf{x},\mathbf{V})$ taking on large values is small, then the buffered failure probability is typically close to the failure probability.

As we discuss below, the buffered failure probability is surprisingly easy to compute, possesses several convenient properties, and avoids many of the difficulties associated with the failure probability. Hence, we believe there are substantial advantages to replacing the failure probability by the buffered failure probability in engineering design.

Example 3. Consider the limit-state function given in Example 1 and recall that $p(0) = p(1) = 4.29 \times 10^{-4}$. We now compute the buffered failure probability for the designs $x_1 = 0$ and 1. Since $g(0,\mathbf{V})$ is given by a triangular probability density function, we determine an α such that $\overline{q}_\alpha(0) = 0$ by integration and find that $\overline{p}(0) = 1\text{-}\alpha = 9.65 \times 10^{-4}$. For design $x_1 = 1$, $\overline{q}_\alpha(\mathbf{x})$ is the expectation of a truncated normal distribution, which is easily calculated by (8). We use trial-and-error to determine an α such that $\overline{q}_\alpha(1) = 0$ and find that $\overline{p}(1) = 1\text{-}\alpha = 1.13 \times 10^{-3}$. We first observe that both designs satisfy (12) as expected. We also see that design $x_1 = 0$ has a smaller buffered failure probability than design $x_1 = 1$ and is therefore "safer" in the sense of the buffered failure probability. This corresponds to our intuition discussed in Example 1, where we concluded that design $x_1 = 0$ was preferable due to the smaller probability of extreme violation of the threshold.

Example 3 illustrates the fact that the buffered failure probability takes into account the tail behavior of the distribution of $g(\mathbf{x},\mathbf{V})$ and hence offers an alternative measure of reliability of a structure that may better reflect designers' concerns.

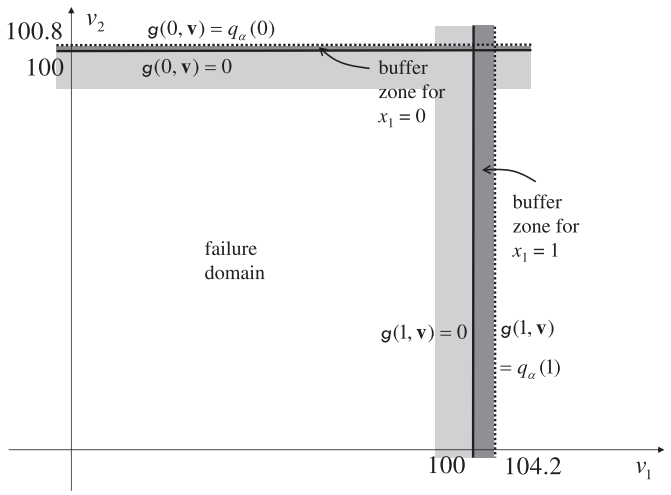

Fig. 10. Contours of limit-state function in Examples 1 and 3 for $x_1 = 0$ and 1.

As seen from (11) and (1), the buffered failure probability shifts the threshold level from zero downwards to $q_\alpha(\mathbf{x})$ (a negative number) and therefore adds a "buffer zone" to the failure domain. We observe that the threshold shift and the buffer zone depend on the probability distribution of $g(\mathbf{x},\mathbf{V})$ and, hence, on $\mathbf{x}$ as illustrated in Fig. 10 for the limit-state function in Examples 1 and 3. That figure shows two solid and two dotted lines. The vertical solid line represents $\{\mathbf{v}|g(1,\mathbf{v}) = 0\}$, i.e., the limit between the failure domain (to the left) and its complement the safe domain (to the right) for design $x_1 = 1$. The dotted vertical line represents $\{\mathbf{v}|g(1,\mathbf{v}) = q_\alpha(1)\}$. This line shifts to the right as compared to the solid line resulting in a buffer zone (shaded dark). Similarly, the horizontal solid line represents $\{\mathbf{v}|g(0,\mathbf{v}) = 0\}$, i.e., the limit between the failure domain (below) and the safe domain (above) for design $x_1 = 0$. The dotted horizontal line represents $\{\mathbf{v}|g(0,\mathbf{v}) = q_\alpha(0)\}$. This line shifts up as compared to the solid line resulting in a narrow buffer zone. We observe that the threshold shift and buffer zone are substantially smaller for $x_1 = 0$ than for $x_1 = 1$. In general, the line, surface, or hypersurface $\{\mathbf{v}|g(\mathbf{x},\mathbf{v}) = q_\alpha(\mathbf{x})\}$ may not be parallel to $\{\mathbf{v}|g(\mathbf{x},\mathbf{v}) = 0\}$ for a given $\mathbf{x}$. In Fig. 10, however, those lines are parallel due to the fact that the limit-state function in Examples 1 and 3 is affine in $\mathbf{v}$.

From the above definition of the superquantile, it may appear difficult to compute the buffered failure probability in general. However, this is not the case as the next subsection describes.

3.2. Buffered failure probability in design optimization

Suppose we would like find a design with failure probability no larger than a threshold $1\text{-}\alpha_0$. That is, we would like to determine a design $\mathbf{x}$ that satisfies the constraint

$$p(\mathbf{x}) \le 1\text{-}\alpha_0. \tag{13}$$

In view of Section 2, we observe that standard optimization algorithms may have substantial difficulties on problems with constraints of the form (13). We now show that the alternative constraint

$$\overline{p}(\mathbf{x}) \le 1\text{-}\alpha_0 \tag{14}$$

in terms of the buffered failure probability is much easier to handle. We start by noting that a design $\mathbf{x}$ that satisfies (14) also satisfies (13). Hence, (14) is a conservative requirement.

The ease with which (14) can be handled in optimization algorithms clearly hinges on our ability to evaluate $\overline{p}(\mathbf{x})$ or equivalent expressions. While $\overline{p}(\mathbf{x})$ cannot be expressed explicitly, there is a convenient, equivalent expression for (14) that we derive next.

In view of Fig. 8, we see that (14) holds if and only if

$$\overline{q}_{\alpha_0}(\mathbf{x}) \le 0. \tag{15}$$

It is shown in [19] that

$$\overline{q}_\alpha(\mathbf{x}) = \min_{z_0} \eta_\alpha(z_0,\mathbf{x}), \tag{16}$$

where z_0 is an auxiliary design variable and

$$\eta_\alpha(z_0,\mathbf{x}) = z_0 + \frac{1}{1\text{-}\alpha} E[\max\{0, g(\mathbf{x},\mathbf{V})\text{-}z_0\}]. \tag{17}$$

We do not include a derivation of this expression as it is somewhat involved and refer the interested reader to [19]. Hence, the task of finding a design $\mathbf{x}$ that satisfies $\overline{p}(\mathbf{x}) \le 1\text{-}\alpha_0$ is equivalent of finding a design $\mathbf{x}$ and an auxiliary variable z_0 such that

$$\eta_{\alpha_0}(z_0,\mathbf{x}) \le 0. \tag{18}$$

Suppose that the goal is to determine a design $\mathbf{x}$ that minimizes some continuously differentiable objective function $f(\mathbf{x})$ (e.g., cost) subject to the reliability constraint $p(\mathbf{x}) \le 1\text{-}\alpha_0$ and a finite number of continuously differentiable equality and inequality constraints abstractly represented by the set X. That is, we would like to solve the design optimization problem

$$\mathbf{P}: \min_{\mathbf{x}} \quad f(\mathbf{x}) \quad \text{s.t.} \quad p(\mathbf{x}) \le 1\text{-}\alpha_0, \quad \mathbf{x} \in X.$$

In view of the discussion above, the alternative formulation in terms of the buffered failure probability takes the form

$$\mathbf{BP}: \min_{\mathbf{x},z_0} \quad f(\mathbf{x}) \quad \text{s.t.} z_0 + \frac{1}{1\text{-}\alpha_0} E[\max\{0, g(\mathbf{x},\mathbf{V})\text{-}z_0\}] \le 0, \quad \mathbf{x} \in X,$$

where we observe that the optimization is over both $\mathbf{x}$ and z_0.

We usually cannot compute $E[\max\{0, g(\mathbf{x},\mathbf{V})\text{-}z_0\}]$ explicitly. However, the expectation can be estimated by its sample average. Let $\mathbf{v}^1,\ldots,\mathbf{v}^N$ be realizations of $\mathbf{V}$. Then, the optimization problem

$$\begin{aligned}\mathbf{BP}'_N: \min_{\mathbf{x},z_0} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & z_0 + \frac{1}{N(1\text{-}\alpha_0)} \sum_{j=1}^{N} \max\{0, g(\mathbf{x},\mathbf{v}^j)\text{-}z_0\} \le 0, \\ & \mathbf{x} \in X,\end{aligned} \tag{19}$$

is an approximation of the problem **BP**. Even if the limit-state function $g(\mathbf{x},\mathbf{v})$ is continuously differentiable for all $\mathbf{v}$, $\mathbf{BP}'_N$ is not directly tractable by standard nonlinear optimization algorithms due to the nonsmoothness of the max- function in (19). $\mathbf{BP}'_N$ is solvable by a specialized algorithm found in [16], but we do not describe that algorithm here. Instead we present an equivalent transcription of $\mathbf{BP}'_N$ that facilitates the use of standard nonlinear optimization algorithms.

We let $z_1,\ldots,z_N$ be auxiliary design variables and denote $\overline{\mathbf{z}} = (z_0, z_1,\ldots,z_N)'$. Then, $\mathbf{BP}'_N$ is equivalent to the following intermediate problem

$$\begin{aligned}\min_{\mathbf{x},\overline{\mathbf{z}}} \quad & f(\mathbf{x}) \quad \text{s.t.} \quad z_0 + \frac{1}{N(1\text{-}\alpha_0)} \sum_{j=1}^{N} z_j \le 0, \quad \max\{0, g(\mathbf{x},\mathbf{v}^j)\text{-}z_0\} \\ & = z_j, \ \ j = 1,2,\ldots,N \quad \mathbf{x} \in X,\end{aligned} \tag{20}$$

where we simply force the auxiliary design variables to take on the "right" values. We can relax the equality constraints to less-than-or-equal constraints as there is no benefit to let the variables take on values such as $\max\{0, g(\mathbf{x},\mathbf{v}^j)\text{-}z_0\} < z_j$ for any $j = 1,2,\ldots,N$. Moreover, a constraint of the form $\max\{0, g(\mathbf{x},\mathbf{v}^j)\text{-}z_0\} \le z_j$ is equivalent to the two constraints $g(\mathbf{x},\mathbf{v}^j)\text{-}z_0 \le z_j$ and $0 \le z_j$. This

leads to the following equivalent problem of $\mathbf{BP}'_N$:

$$\mathbf{BP}_N: \min_{\mathbf{x},\bar{z}} \quad f(\mathbf{x}) \quad \text{s.t.} \quad z_0 + \frac{1}{N(1\text{-}\alpha_0)}\sum_{j=1}^{N} z_j \le 0$$
$$g(\mathbf{x},\mathbf{v}^j)\text{-}z_0 \le z_j,\ \ j=1,2,\ldots,N,$$
$$z_j \ge 0,\ \ j=1,2,\ldots,N,$$
$$\mathbf{x}\in X. \tag{21}$$

We propose that engineers consider $\mathbf{BP}_N$ instead of $\mathbf{P}$ when designing structures for reasons summarized next.

3.3. Comparison of probabilities

There are four main advantages to consider $\mathbf{BP}_N$ instead of $\mathbf{P}$. First, as discussed above, the failure probability $p(\mathbf{x})$ and its gradient cannot generally be computed exactly and must be approximated in ways which, in some cases, might turn a blind eye to serious risks. The first-order approximation $\Phi(\text{-}\beta(\mathbf{x}))$ has unknown accuracy and may not be continuously differentiable. Monte Carlo estimates of $p(\mathbf{x})$ have error bounds, but the estimates have gradients only under assumptions that are difficult to verify in practice. Hence, it is highly problematic to apply standard nonlinear optimization algorithms to optimization problems involving $p(\mathbf{x})$. In contrast, $\mathbf{BP}_N$ is solvable by standard nonlinear optimization algorithms as long as the limit-state function $g(\mathbf{x},\mathbf{v})$ is continuously differentiable with respect to $\mathbf{x}$. This is a substantially less stringent condition than those required for $\mathbf{P}$. The optimal value of $\mathbf{BP}_N$ is close to the optimal value of $\mathbf{BP}$ when N is large (see [29, Chapter 4] for specific results on the "proximity" of $\mathbf{BP}_N$ to $\mathbf{BP}$). Moreover, $\mathbf{BP}$ is a restricted problem compared to $\mathbf{P}$ because the buffered failure probability overestimates the failure probability, see (12). Hence, a feasible design in $\mathbf{BP}$ is also feasible in $\mathbf{P}$.

Second, the buffered failure probability provides an alternative measure of structural reliability which accounts for the tail behavior of the distribution of $g(\mathbf{x},\mathbf{V})$. Hence, designs obtained from $\mathbf{BP}_N$ may be more desirable than those from $\mathbf{P}$.

Third, even if $g(\mathbf{x},\mathbf{v})$ is convex in $\mathbf{x}$, $p(\mathbf{x})$ and $\Phi(\text{-}\beta(\mathbf{x}))$ may not be and, hence, it may be difficult to obtain a globally optimal design of $\mathbf{P}$. In contrast, the region defined by the constraints (21) is convex when $g(\mathbf{x},\mathbf{v}^j)$, $j=1,2,\ldots,N$, are convex functions in $\mathbf{x}$. Hence, every KKT point of $\mathbf{BP}_N$ is a globally optimal design when $f(\mathbf{x})$ and $g(\mathbf{x},\mathbf{v}^j)$, $j=1,2,\ldots,N$, are convex functions and the region X is a convex set. Hence, $\mathbf{BP}_N$ "preserves" convexity. Even if not all of these conditions are satisfied, we expect it to often be easier to determine a design with a low objective function value in $\mathbf{BP}_N$ than in $\mathbf{P}$ because $\mathbf{BP}_N$ deals with $g(\mathbf{x},\mathbf{v})$ directly instead of the more complex expression $p(\mathbf{x})$.

We expect $g(\mathbf{x},\mathbf{v})$ to be convex in $\mathbf{x}$ in several practical situations. For example, suppose that $\mathbf{x}=x_1$ represents the size of a part of the structure and the strength $R(\mathbf{x},\mathbf{v})$ of the structure grows as x_1 grows for all possible realizations $\mathbf{v}$. Moreover, suppose that this growth in strength is constant or tapers off as x_1 grows. Then, $R(\mathbf{x},\mathbf{v})$ is concave for all $\mathbf{v}$ and, hence, the limit-state function $g(\mathbf{x},\mathbf{v})=S(\mathbf{v})\text{-}R(\mathbf{x},\mathbf{v})$ is convex, where $S(\mathbf{v})$ describes the load on the structure. Since the convexity of $g(\mathbf{x},\mathbf{v})$ with respect to $\mathbf{x}$ was of little importance in the context of $p(\mathbf{x})$, few researchers have focused on developing convex limit-state functions or approximations thereof. As the importance of convexity is now clear, we hope that this paper will spur research into the development of convex limit-state functions. While physics dictate to a large extent the form of limit-state functions, engineers may still have opportunities for skillful modeling, including the development of useful approximations. In the same manner as a simple limit-state function $g(\mathbf{x},\mathbf{v})=v_2\text{-}x_1v_1$, which is linear in $\mathbf{x}$, is equivalent to the limit-state function $\hat{g}(\mathbf{x},\mathbf{v})=v_2/(x_1v_1)\text{-}1$, which is nonlinear in $\mathbf{x}$, we expect the development of (approximately) equivalent convex limit-state functions to existing nonconvex limit-state functions.

Fourth, $\mathbf{BP}_N$ facilitates the development of approximation schemes for limit-state functions that are expensive to evaluate. For example, if the evaluation of the limit-state function involves the output of a finite element model, it may not be possible to evaluate the limit-state function more than a few hundred or a few thousand times. In such situations, the failure probability in $\mathbf{P}$ is often replaced by response surface and surrogate models, see, e.g., [7,31,34]. This allows quick optimization, but the quality of the resulting design depends on the fidelity of the response surface or surrogate model used. As $p(\mathbf{x})$ may be a highly nonlinear, nonconvex function, we conjecture that it may be more difficult and computationally expensive to develop a good surrogate model of $p(\mathbf{x})$ than of $g(\mathbf{x},\mathbf{v})$, about which we may have problem-specific insight. With a surrogate model of $g(\mathbf{x},\mathbf{v})$, the optimization of $\mathbf{BP}_N$ using that surrogate model in place of $g(\mathbf{x},\mathbf{v})$ can often be accomplished relatively quickly; see Section 5.2. For example, suppose that $\mathbf{x}^k$, $k=1,2,\ldots,K$, is a selection of designs. Then, for any $\mathbf{v}$ and k,

$$g(\mathbf{x},\mathbf{v}) \approx g(\mathbf{x}^k,\mathbf{v}) + \nabla_{\mathbf{x}} g(\mathbf{x}^k,\mathbf{v})'(\mathbf{x}\text{-}\mathbf{x}^k) \tag{22}$$

when $\mathbf{x}$ is close to $\mathbf{x}^k$ and $g(\mathbf{x},\mathbf{v})$ is continuously differentiable with respect to $\mathbf{x}$. Obviously, the selection of $\mathbf{x}^k$, $k=1,2,\ldots,K$, e.g., by means of an experimental design, influences the accuracy of this approximation and is an important topic in its own right. In this paper, however, we do not discuss this topic further. Interested readers are referred to [34] and references therein. Using this linear approximation of the limit-state function, we obtain the following approximation of $\mathbf{BP}_N$, which is intended for the case when $g(\mathbf{x},\mathbf{v}^j)$ is convex in $\mathbf{x}$ for all $j=1,2,\ldots,N$:

$$\mathbf{LBP}_N: \min_{\mathbf{x},\bar{z}} \quad f(\mathbf{x}) \quad \text{s.t.} \quad z_0 + \frac{1}{N(1\text{-}\alpha_0)}\sum_{j=1}^{N} z_j \le 0,$$
$$g(\mathbf{x}^k,\mathbf{v}^j) + \nabla_{\mathbf{x}} g(\mathbf{x}^k,\mathbf{v}^j)'(\mathbf{x}\text{-}\mathbf{x}^k)\text{-}z_0 \le z_j,\ \ j=1,2,\ldots,N,\ \ k=1,2,\ldots,K,$$
$$z_j \ge 0,\ \ j=1,2,\ldots,N \qquad \mathbf{x}\in X.$$

Under that convexity assumption, $\mathbf{LBP}_N$ can be made to approximate $\mathbf{BP}_N$ arbitrarily well by selecting more designs appropriately, i.e., increasing K. We note, however, that $\mathbf{LBP}_N$ is a nonconservative approximation of $\mathbf{BP}_N$. The construction of conservative approximations of $\mathbf{BP}_N$ would also be possible under suitable assumptions, but that topic is beyond the scope of the current paper.

Solving $\mathbf{LBP}_N$ only requires the evaluation of the limit-state function and its gradient KN times to generate the problem data in $\mathbf{LBP}_N$. During optimization no evaluation of the limit-state function or its gradient is needed and, hence, can be carried out quickly. If the objective function $f(\mathbf{x})$ and the constraints defining X are linear, then $\mathbf{LBP}_N$ is a linear program that can be solved quickly by standard linear programming solvers or decomposition algorithms. In this case, the introduction of integrality restrictions on $\mathbf{x}$ may also be tractable as this makes $\mathbf{LBP}_N$ a mixed-integer linear optimization problem that often can be solved in moderate computing times. In comparison, it is difficulty to solve $\mathbf{P}$ in the case of integrality constraints as it then becomes a mixed-integer, nonlinear, nonconvex, optimization problem.

3.4. Variance reduction

While a large sample size N provides a good approximation of $\mathbf{BP}_N$ to $\mathbf{BP}$, the number of constraints and decision variables in $\mathbf{BP}_N$ grows linearly in N. The accuracy of $\mathbf{BP}_N$ for a moderate N

is often substantially improved through variance reduction techniques such as importance sampling, see, e.g. [26]. If all realizations $\mathbf{v}^1,\ldots,\mathbf{v}^N$ result in satisfactory structural performance for relevant designs, i.e., $g(\mathbf{x},\mathbf{v}^j)\le 0$ for all $j=1,2,\ldots,N$, then globally optimal solutions for $z_0,z_1,\ldots,z_N$ are all zero. This implies that the optimal design in $\mathbf{BP}_N$ is simply the $\mathbf{x}$ that minimizes the objective function $f(\mathbf{x})$ over X. Consequently, the possibility of failure of the structure is not accounted for in $\mathbf{BP}_N$ in the case of such realizations. Hence, it is important that some of the realizations result in $g(\mathbf{x},\mathbf{v}^j)>0$ for relevant designs. We can typically accomplish this by increasing N or, more efficiently, by importance sampling, which we describe next.

Let $\mathbf{W}$ be a random vector with m random variables with joint probability density function $f_{\mathbf{W}}(\mathbf{w})$ with $f_{\mathbf{W}}(\mathbf{w})>0$ for all $\mathbf{w}$ satisfying $f_{\mathbf{V}}(\mathbf{w})>0$. Let $\mathbf{w}^1,\ldots,\mathbf{w}^N$ be realizations of $\mathbf{W}$. Then, we redefine

$$\begin{aligned}\mathbf{BP}_N:\ &\min_{\mathbf{x},\overline{\mathbf{z}}}\quad f(\mathbf{x})\\ &\text{s.t.}\quad z_0+\frac{1}{N(1\text{-}\alpha_0)}\sum_{j=1}^{N} z_j\le 0,\\ &\quad (g(\mathbf{x},\mathbf{w}^j)\text{-}z_0)\frac{f_{\mathbf{V}}(\mathbf{w}^j)}{f_{\mathbf{W}}(\mathbf{w}^j)}\le z_j,\ \ j=1,2,\ldots,N,\\ &\quad z_j\ge 0,\ \ j=1,2,\ldots,N,\qquad \mathbf{x}\in X.\end{aligned}\tag{23}$$

By generating realizations from an appropriately selected probability density $f_{\mathbf{W}}(\mathbf{w})$, we can ensure that a substantial number of realizations $\mathbf{w}^j$ satisfies $g(\mathbf{x},\mathbf{w}^j)>0$ for relevant designs. In practice, $f_{\mathbf{W}}(\mathbf{w})$ can typically be selected by increasing (decreasing) mean values of random variables describing loads (material strength). For more sophisticated approaches to selecting $f_{\mathbf{W}}(\mathbf{w})$ we refer to [26].

4. System reliability

4.1. Problem formulation

The performance of a structure is often given by multiple limit-state functions representing quantities such as stresses and deformations at different locations. Let $g_k(\mathbf{x},\mathbf{v})$, $k=1,2,\ldots,K$, be a collection of limit-state functions describing the relevant limit states for a structure. We define a cut-set to be a (sub)set of these limit-state functions with the characteristics that if all the limit-state functions in the cut-set are unsatisfactory for a given design $\mathbf{x}$ and realization $\mathbf{v}$, i.e., $g_k(\mathbf{x},\mathbf{v})>0$, then the structure experiences system failure. A cut-set is minimal if no limit-state function can be removed from the cut-set without rendering the resulting set not a cut-set. We refer to an individual limit-state function being unsatisfactory as component failure. Suppose there are i_c minimal cut-sets. We denote the set of limit-state functions belonging to minimal cut-set i by C_i, $i\in I=\{1,2,\ldots,i_c\}$. As system failure occurs in the event of component failure with respect to all limit-state functions in any minimal cut-set, the system failure probability is defined as

$$p^s(\mathbf{x})=P\left[\bigcup_{i\in I}\bigcap_{k\in C_i}\{g_k(\mathbf{x},\mathbf{V})>0\}\right].\tag{24}$$

If the cardinality of C_i, denoted $|C_i|$, is one for all $i\in I$, then the structure is a series structural system as the failure of any component results in system failure. On the other hand, if $i_c=1$, then the structure is a parallel system as system failure only occurs if all components fail.

It follows directly from (24) that

$$p^s(\mathbf{x})=P[\{g(\mathbf{x},\mathbf{V})>0\}],\tag{25}$$

where

$$g(\mathbf{x},\mathbf{v})=\max_{i\in I}\min_{k\in C_i} g_k(\mathbf{x},\mathbf{v})\tag{26}$$

is a system limit-state function. Hence, the design optimization problem with system failure constraints generalizes $\mathbf{P}$ and takes the form

$$\mathbf{P}^s:\ \min_{\mathbf{x}}\quad f(\mathbf{x})\quad\text{s.t.}\quad p^s(\mathbf{x})\le 1\text{-}\alpha_0,\quad \mathbf{x}\in X.$$

As $\mathbf{P}^s$ is at least as intractable as $\mathbf{P}$, we consider a formulation involving the buffered *system* failure probability.

4.2. Using the buffered failure probability

Following the approach of Section 3, we define analogously to $\mathbf{BP}_N$ the problem

$$\begin{aligned}\mathbf{BP}_N^s:\ &\min_{\mathbf{x},\overline{\mathbf{z}}}\quad f(\mathbf{x})\\ &\text{s.t.}\quad z_0+\frac{1}{N(1\text{-}\alpha_0)}\sum_{j=1}^{N} z_j\le 0,\\ &\min_{k\in C_i}\quad g_k(\mathbf{x},\mathbf{v}^j)\text{-}z_0\le z_j,\ \ i\in I,\ \ j=1,2,\ldots,N,\\ &\quad z_j\ge 0,\ \ j=1,2,\ldots,N,\qquad \mathbf{x}\in X,\end{aligned}\tag{27}$$

where we use in (27) the fact that

$$\max_{i\in I}\min_{k\in C_i} g_k(\mathbf{x},\mathbf{v}^j)\text{-}z_0\le z_j,\quad j=1,2,\ldots,N\tag{28}$$

is equivalent to (27). The relationship between $\mathbf{P}^s$ and $\mathbf{BP}_N^s$ is identical to those between $\mathbf{P}$ and $\mathbf{BP}_N$. Hence, we recommend designers to consider $\mathbf{BP}_N^s$ instead of $\mathbf{P}^s$.

In the case of series structural systems, i.e., each minimal cut-set includes only one limit-state function, $\mathbf{BP}_N^s$ is identical to $\mathbf{BP}_N$, except it includes more constraints of the same form. Hence, $\mathbf{BP}_N^s$ is tractable by standard nonlinear optimization algorithms when $g_k(\mathbf{x},\mathbf{v})$, $k=1,2,\ldots,K$, are continuously differentiable. Moreover, convexity is preserved as in the case of $\mathbf{BP}_N$.

Cases with general or parallel structural systems are more complicated. The minimum over limit-state functions in (27) causes $\mathbf{BP}_N^s$ to become a nonsmooth optimization problem even if $g_k(\mathbf{x},\mathbf{v})$, $k=1,2,\ldots,K$, are continuously differentiable. Hence, standard nonlinear optimization algorithms are not applicable. We propose three alternative approaches to overcome this difficulty.

The first alternative transcribes the problem into a finite, but potentially large number of optimization subproblems. Specifically, $\mathbf{BP}_N^s$ is equivalent to

$$\begin{aligned}&\min_{k_{ij}\in C_i, i\in I, j=1,2,\ldots,N}\quad \min_{\mathbf{x},\overline{\mathbf{z}}} f(\mathbf{x})\\ &\text{s.t.}\quad z_0+\frac{1}{N(1\text{-}\alpha_0)}\sum_{j=1}^{N} z_j\le 0,\\ &\quad g_{k_{ij}}(\mathbf{x},\mathbf{v}^j)\text{-}z_0\le z_j,\ \ i\in I,\ \ j=1,2,\ldots,N,\\ &\quad z_j\ge 0,\ \ j=1,2,\ldots,N,\quad \mathbf{x}\in X.\end{aligned}\tag{29}$$

This problem amounts to minimizing $\prod_{i\in I}|C_i|^N$ subproblems essentially of the form $\mathbf{BP}_N$ and retaining the design with the best objective function value. A main advantage of this transcription is that it preserves convexity. That is, if $g_k(\mathbf{x},\mathbf{v})$, $k=1,2,\ldots,K$, are convex functions with respect to $\mathbf{x}$, $f(\mathbf{x})$ is a convex function, and X is a convex set, then each of the $\prod_{i\in I}|C_i|^N$ subproblems are convex.

A design found in one of the subproblems can be used to warm start the calculations of the next subproblem. However, the main

challenge with this approach is the large number of subproblems to solve. If it is not practical to (approximately) solve all subproblems, then it is always possible to solve only a subset of the subproblems. This provides a conservative design as (29) is a restriction of (27) and further improvement might be possible after solving other subproblems.

The second alternative avoids the large number of subproblems by using exponential smoothing [1,17]. This alternative replaces the nonsmooth function

$$\psi(\mathbf{x},\mathbf{v}) = \min_{k\in C_i} g_k(\mathbf{x},\mathbf{v}) \tag{30}$$

in (27) by a continuously differentiable approximation. For any approximation parameter $\varepsilon > 0$, let

$$\tilde{g}_i(\mathbf{x},\mathbf{v};\varepsilon) = -\varepsilon \ln\left(\sum_{k\in C_i} e^{-g_k(\mathbf{x},\mathbf{v})/\varepsilon}\right) \tag{31}$$

be this approximation. We know that

$$0 \le \psi(\mathbf{x},\mathbf{v}) - \tilde{g}_i(\mathbf{x},\mathbf{v};\varepsilon) \le \varepsilon \ln|C_i| \tag{32}$$

for all $\mathbf{x}$, $\mathbf{v}$, and $\varepsilon > 0$. Hence, the smooth approximation $\tilde{g}_i(\mathbf{x},\mathbf{v}^j;\varepsilon)$ underestimates $\psi(\mathbf{x},\mathbf{v}^j)$ and the error in the approximation vanishes as $\varepsilon \to 0$.

We now simply replace $\min_{k\in C_i} g_k(\mathbf{x},\mathbf{v}^j)$ in (27) by its smooth approximation for all i and j. This results in the following problem

$$\begin{aligned}
\mathbf{BP}_N^s(\varepsilon):\ \min_{\mathbf{x},\bar{z}}\quad & f(\mathbf{x}) \\
\text{s.t.}\quad & z_0 + \frac{1}{N(1-\alpha_0)}\sum_{j=1}^{N} z_j \le 0, \\
& \tilde{g}_i(\mathbf{x},\mathbf{v}^j;\varepsilon) + \varepsilon\ln|C_i| - z_0 \le z_j,\ \ i\in I,\ \ j=1,2,\ldots,N, \\
& z_j \ge 0,\ \ j=1,2,\ldots,N,\quad \mathbf{x}\in X.
\end{aligned} \tag{33}$$

Since we included the error term $\varepsilon\ln|C_i|$ in (33), a design that is feasible in $\mathbf{BP}_N^s(\varepsilon)$ is also feasible in $\mathbf{BP}_N^s$. If the limit-state functions $g_k(\mathbf{x},\mathbf{v})$, $k=1,2,\ldots,K$, are continuously differentiable, then standard nonlinear optimization algorithms are applicable to $\mathbf{BP}_N^s(\varepsilon)$.

We observe, however, that even if $g_k(\mathbf{x},\mathbf{v})$, $k=1,2,\ldots,K$, are convex, $\mathbf{BP}_N^s(\varepsilon)$ is not a convex optimization problem. In essence, minimal cut-sets with cardinality larger than one introduce nonconvexity in the design optimization problem. We also note that exponential smoothing can be used in $\mathbf{BP}_N$ to replace the N constraints (21) by one single constraints.

The third alternative for solving $\mathbf{BP}_N^s$ adapts the approach in [12]. In that paper it is shown that $\mathbf{BP}_N^s$ is equivalent to the following problem:

$$\begin{aligned}
\mathbf{EBP}_N^s:\ \min_{\mathbf{x},\bar{z},\mu_{ij}^k, i\in I, j=1,\ldots,N, k\in C_i}\quad & f(\mathbf{x}) \\
\text{s.t.}\quad & z_0 + \frac{1}{N(1-\alpha_0)}\sum_{j=1}^{N} z_j \le 0, \\
& \sum_{k\in C_i} \mu_{ij}^k g_k(\mathbf{x},\mathbf{v}^j) - z_0 \le z_j,\ \ i\in I,\ \ j=1,2,\ldots,N, \\
& z_j \ge 0,\ \ j=1,2,\ldots,N,\quad \mathbf{x}\in X, \\
& \sum_{k\in C_i} \mu_{ij}^k = 1,\ \ i\in I,\ \ j=1,\ldots,N, \\
& \mu_{ij}^k \ge 0,\ \ i\in I, j=1,\ldots,N,\ \ k\in C_i,
\end{aligned} \tag{34}$$

where $\mu_{ij}^k, i\in I,\ j=1,\ldots,N,\ k\in C_i$, is a set of auxiliary design variables that effectively "select" which limit-state functions in (27) are active. The equivalence between $\mathbf{EBP}_N^s$ and $\mathbf{BP}_N^s$ is in the sense that a globally (locally) optimal solution from one problem can be used to construct a globally (locally) optimal solution of the other problem. If $g_k(\mathbf{x},\mathbf{v})$, $k=1,2,\ldots,K$, are continuously differentiable, then standard nonlinear optimization algorithms are applicable for solving $\mathbf{EBP}_N^s$. However, even if $g_k(\mathbf{x},\mathbf{v})$, $k=1,2,\ldots,K$, are convex, $\mathbf{EBP}_N^s$ may not be a convex problem because (34) involves a product of design variables.

5. Computational studies

We illustrate the use of the buffered failure probability with the design of a truss structure and a motor vehicle.

5.1. Optimal truss design

Consider the simply supported truss in Fig. 11. Let V_k be the yield stress of member k, $k=1,2,\ldots,7$. Members 1 and 2 have lognormally distributed yield stresses with mean 100 N/mm^2 and standard deviation 20 N/mm^2. The other members have lognormally distributed yield stresses with mean 200 N/mm^2 and standard deviation 40 N/mm^2. The yield stresses of members 1 and 2 are correlated with correlation coefficients 0.8. However, their correlation coefficients with the other yield stresses are 0.5. Similarly, the yield stresses of members 3–7 are correlated with correlation coefficients 0.8. The truss is subject to a random load V_8 in its mid-span. V_8 is lognormally distributed with mean 1000 kN and standard deviation 400 kN. The load V_8 is independent of the yield stresses. We use a joint lognormal distribution (see [5, Section 7.2]) and the above correlation coefficients to approximate the joint distribution of $\mathbf{V} = (V_1, V_2, \ldots, V_8)$.

The design vector $\mathbf{x} = (x_1, x_2, \ldots, x_7)$, where x_k is the cross-section area (in 1000 mm^2) of member k. The truss fails if any of the members exceed their yield stress (We ignore the possibility of buckling.) This gives rise to seven limit state functions:

$$g_k(\mathbf{x},\mathbf{v}) = v_8/\zeta_k - v_k x_k,\quad k=1,2,\ldots,7, \tag{35}$$

where ζ_k is a factor given by the geometry and loading of the truss. From Fig. 11, we determine that $\zeta_k = 1/(2\sqrt{3})$ for $k=1,2$, and $\zeta_k = 1/\sqrt{3}$ for $k=3,4,\ldots,7$.

We impose the constraint that the series system failure probability with the seven limit-state functions should be no larger than 0.00135, i.e.,

$$p^s(\mathbf{x}) = P\left[\bigcup_{k=1}^{7}\{g_k(\mathbf{x},\mathbf{V}) > 0\}\right] \le 0.00135. \tag{36}$$

We also impose the 14 deterministic constraints $0.5 \le x_k \le 2$, $k=1,2,\ldots,7$, that limit the allowable area of each member to be between 500 and 2000 mm^2. We seek a design of the truss that minimizes the cost of the truss. Since all members are equally long, the cost is $f(\mathbf{x}) = \sum_{k=1}^{7} x_k$. This problem is of the form $\mathbf{P}^s$ and, hence, we solve $\mathbf{BP}_N^s$ with (27) replaced by

$$g_k(\mathbf{x},\mathbf{v}^j) - z_0 \le z_j,\quad k=1,2,\ldots,7,\ \ j=1,2,\ldots,N \tag{37}$$

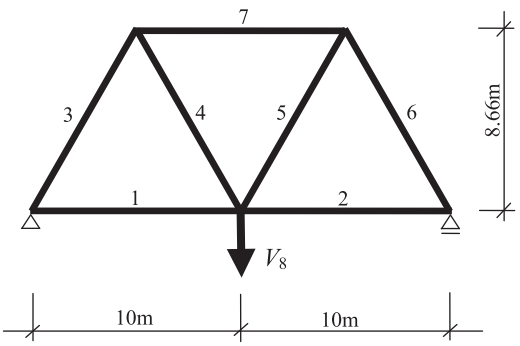

Fig. 11. Design of Truss.

Table 1
Design of truss.

Method	1-α_0	Design of member (in mm^2)							Cost (mm^2)	Failure prob.	Buffered prob.
		1	2	3	4	5	6	7			
$\mathbf{BP}_N^s$	0.00135	1320	1332	1272	1278	1271	1278	1271	9022	0.00047	0.00135
[25]	0.00135	1138	1156	1118	1107	1119	1113	1108	7859	0.00153	0.00410
$\mathbf{BP}_N^s$	0.00410	1153	1179	1100	1105	1106	1109	1101	7852	0.00154	0.00410

Table 2
Design of motor vehicle.

Method	Optimized design							Cost	Failure prob.	Buffered prob.
	x_1	x_2	x_3	x_4	x_5	x_6	x_7			
$\mathbf{BP}_N^s$	0.5000	1.3524	0.5000	1.2989	0.6103	1.5000	0.5000	24.60	0.00067	0.00130
[27]	0.5000	1.3251	0.5000	1.2919	0.5964	1.5000	0.5000	24.37	0.00347	0.01769

as our example is a series structural system. Since the limit-state functions, objective functions, and constraints are linear in $\mathbf{x}$, $\mathbf{BP}_N^s$ is a linear program that can be solved quickly by standard optimization solvers. We use sample size $N = 10\,000$ and the variance reduction technique of Section 3.4 where we select the sampling distribution to be identical to the original distribution except that the mean value of the load is increased with three standard deviations.

We implement the resulting linear instance of $\mathbf{BP}_N^s$ in the General Algebraic Modeling System (GAMS) Distribution 22.9 [6] on a laptop computer with 1 GB of RAM and 2.16 GHz processor running Windows XP. The globally optimal design of $\mathbf{BP}_N^s$ is found by the solver CPLEX 11.2 [11] with default options in 19.4 s. The optimized design is shown in row 3 of Table 1 with the resulting buffered failure probability in the last column of the table. In this case, the buffered failure probability is about three times larger than the failure probability. Here, the failure probability is estimated by importance sampling with a 5% coefficient of variation using an independent sample. For a rigorous solution validation procedure we refer to [21]. For comparison, we also report in row 4 of Table 1 the design found for the same truss in [25] by approximately solving $\mathbf{P}^s$ using sample average approximations. Since $\mathbf{P}^s$ requires the failure probability to be no larger than a threshold and $\mathbf{BP}_N^s$, effectively, imposes the same threshold on the buffered failure probability, the design of row 4 is naturally cheaper than the one in row 3. However, the former design is less safe with an estimated failure probability of 0.00153 (5% coefficient of variation of estimate), which slightly exceeds the threshold of 0.00135. While the algorithm in [25] is guaranteed to converge to a feasible design satisfying the KKT conditions under suitable assumptions, termination of the algorithm after a finite amount of calculation time may result in such infeasibilities. In contrast, the design obtained by using the buffered failure probability has an estimated failure probability below the required threshold of 0.00135. Moreover, the calculation time of the algorithm in [25] is substantially longer than that of solving $\mathbf{BP}_N^s$, with a time exceeding one hour to obtain the design in row 4 of Table 1. While an improved implementation of the algorithm in [25] will reduce this time, the advantage of the buffered failure probability appears substantial.

To better compare the design obtained using the buffered failure probability with that using the failure probability, we also solve $\mathbf{BP}_N^s$ with probability threshold 0.00410. This threshold equals the buffered failure probability of the design in row 4 of Table 1. Row 5 of the table gives the resulting design obtained after 20.5 s using CPLEX. We see that the designs in rows 4 and 5 are essentially identical, which indicate that optimization with the buffered failure probability gives a similar design to that obtained using the failure probability when the threshold is appropriately adjusted. We note again that the computing time is dramatically reduces when using the buffered failure probability.

5.2. Motor vehicle design

We consider an example given in [27] (see also [8]) where the goal is to minimize the weight of a part of a motor vehicle subject to reliability constraints related to side impact. We formulate this problem in the form $\mathbf{P}^s$ with a series system failure probability with respect to ten limit-states functions and a reliability level 1-α_0 of 0.0013. The limit-state functions are surrogate models of the real structural performance; see [27]. The example has seven design variables relating to the thickness of material (The paper [27] includes four additional variables, which we simply fix to the values reported in [27], i.e., 0.345, 0.345, 0, and 0.) All thicknesses must be in the interval [0.5 1.5]. The thicknesses cannot be manufactured exactly and, hence, the limit-state functions include normally distributed manufacturing errors with zero mean and standard deviation 0.03 for each thickness. The errors are statistically independent. We refer to [27] for details of this example.

We implement $\mathbf{BP}_N^s$ for this example with sample size $N = 7500$ using the same hardware as above, but now solve the problem using SNOPT [9] as implemented in TOMLAB [30]. Table 2 gives the optimized design in row 3, which was obtained after 166 s, and the resulting buffered failure probability; see the last column. The corresponding failure probability is estimated by Monte Carlo sampling with a 5% coefficient of variation using an independent sample; see the second to last column of row 3. For comparison, we also report the design given in [27] with estimated failure and buffered failure probabilities (5% coefficient of variation); see row 4 of Table 2. Again, we see that our methodology results in a reasonable design in short computing time.

6. Conclusions

We discuss several theoretical, practical, and computational issues associated with the failure probability with particular

emphasis on difficulties arising in design optimization. We propose an alternative measure, the buffered failure probability, that offers significant advantages. The buffered failure probability accounts for the degree of violation of a performance threshold, is more conservative than the failure probability, and is handled with relative ease in design optimization problems. The paper formulates several design optimization problems in terms of the buffered failure probability and discusses their relation to design optimization problems in terms of the failure probability. We find the buffered failure probability to be superior to the failure probability and recommends its use in design and optimization of structures.

While the buffered failure probability appears promising for use in design optimization with reliability constraints, its applicability in other optimization models such as those with a von Neumann–Morgenstern maximum expected utility criterion is unclear. Moreover, the buffered failure probability requires the estimation of an expectation, which may be computationally costly, and may result in large-scale optimization models. These challenges should be the subject of further study.

Acknowledgments

The second author acknowledges financial supported from Air Force Office of Scientific Research Young Investigator grant F1ATA08337G003. The authors are thankful for comments provided by Prof. A. Der Kiureghian, University of California, Berkeley, on a draft of this paper and also acknowledge assistance from Mr. Habib G. Basova, Naval Postgraduate School, with a numerical example. Opinions and statements expressed in this paper, however, are solely those of the authors.

References

[1] Bertsekas DP. Constrained optimization and Lagrange multiplier methods. New York, New York: Academic Press; 1982.
[2] Bertsekas DP. Nonlinear programming, 2nd ed. Belmont, MA: Athena Scientific; 1999.
[3] Stephen Boyd, Lieven Vandenberghe. Convex optimization. New York: Cambridge University Press; 2004.
[4] Conn AR, Gould NIM, Toint PhL. LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A). Springer series in computational mathematics, vol. 17. Heidelberg, Germany: Springer; 1992.
[5] Ditlevsen O, Madsen HO. Structural reliability methods. New York: Wiley; 1996.
[6] GAMS. GAMS Distribution 22.9. GAMS Development Corporation, Washington, DC, 2008 〈http://www.gams.com〉 (last accessed 22 December 2008).
[7] Gasser M, Schueller GI. Some basic principles in reliability-based optimization RBO of structures and mechanical components. In: Marti K, Kall P, editors. Stochastic programming methods and technical applications. Lecture notes in economics and mathematical systems, vol. 458. Berlin, Germany: Springer; 1998.
[8] Gu L, Yang RJ, Tho CH, Makowski M, Faruque O, Li Y. Optimization and robustness for crashworthiness of side impact. International Journal of Vehicle Design 2001;26(4):348–60.
[9] Gill P, Murray W, Saunders M. User's guide for SNOPT 5.3: a Fortran package for large-scale nonlinear programming. Technical Report SOL-98-1, System Optimization Laboratory, Stanford University, Stanford, CA, 1998.
[10] Horst R, Pardalos PM, Thoai NV. Introduction to global optimization, 2nd ed. Dordrecht, Netherlands: Kluwer Academic; 2000.
[11] ILOG. CPLEX 11.2 Documentation. ILOG, Inc., Mountain View, California, 2008 〈http://www.cplex.com〉 (last accessed 22 December 2008).
[12] Kirjner-Neto C, Polak E. On the conversion of optimization problems with max-min constraints to standard optimization problems. SIAM J Optim 1998;8(4):887–915.
[13] Liu P-L, Der Kiureghian A. Optimization algorithms for structural reliability. Struct Saf 1991;9:161–77.
[14] Marti K. Differentiation formulas for probability functions: the transformation method. Math Programming 1996;75:201–20.
[15] Marti K. Stochastic optimization methods. Berlin: Springer; 2005.
[16] Polak E, Qi L, Sun D. First-order algorithms for generalized semi-infinite min–max problems. Comput Optim Appl 1999;13:137–61.
[17] Polak E, Royset JO, Womersley RS. Algorithms with adaptive smoothing for finite minimax problems. J Optim Theory Appl 2003;119(3):459–84.
[18] Rockafellar RT. Coherent approaches to risk in optimization under uncertainty. In: Tutorials in operations research, INFORMS, 2007. p. 39–61.
[19] Rockafellar RT, Uryasev S. Optimization of conditional value-at-risk. J Risk 2000;2:21–42.
[20] Rockafellar RT, Uryasev S. Conditional value-at-risk for general loss distributions. J Banking Finance 2002;26:1443–71.
[21] Royset JO. Optimality functions in stochastic programming. Available at: 〈http://faculty.nps.edu/joroyset/docs/Royset_optimfcn.pdf〉, 2009.
[22] Royset JO, Polak E. Implementable algorithm for stochastic programs using sample average approximations. J Optim Theory Appl 2004;122(1):157–84.
[23] Royset JO, Polak E. Reliability-based optimal design using sample average approximations. J Probab Eng Mech 2004;19(4):331–43.
[24] Royset JO, Polak E. Extensions of stochastic optimization results from problems with simple to problems with complex failure probability functions. J Optim Theory Appl 2007;133(1):1–18.
[25] Royset JO, Polak E. Sample average approximations in reliability-based structural optimization: theory and applications. In: Papadrakakis M, Tsompanakis Y, Lagaros ND, editors. Structural design optimization considering uncertainties. Taylor & Francis; 2008. p. 307–34.
[26] Rubinstein RY, Kroese DP. Simulation and the Monte Carlo method. New York: Wiley; 2008.
[27] Samson S, Thoomu S, Fadel G, Reneke J. Reliable design optimization under aleatory and epistemic uncertainty. In: Proceedings of ASME 2009 international design engineering technical conferences and 35th design automation conference, DETC2009-86473, San Diego, CA, 2009.
[28] Schittkowski K. User's guide to nonlinear programming code, handbook to optimization program package NLPQL. University of Stuttgart, Stuttgart, Germany, 1985.
[29] Shapiro A, Ruszczynski A. Lectures on stochastic programming. Available at: 〈http://www2.isye.gatech.edu/ashapiro/publications.html〉, 2008.
[30] Holmstrom K. Tomlab optimization 〈http://tomopt.com〉, 2009.
[31] Torczon V, Trosset MW. Using approximations to accelerate engineering design optimization. In: Proceedings of the 7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, AIAA Paper 98-4800, St. Louis, Missouri, 1998.
[32] Tretiakov G. Stochastic quasi-gradient algorithms for maximization of the probability function. A new formula for the gradient of the probability function. In: Stochastic optimization techniques. New York: Springer; 2002. p. 117–42.
[33] Uryasev S. Derivatives of probability functions and some applications. Ann Oper Res 1995;56:287–311.
[34] Viana FAC, Picheny V, Haftka RT. Conservative prediction via safety margin: design through cross-validation and benefits of multiple surrogates. In: Proceedings of the ASME 2009 international design engineering technical conferences and computers and information in engineering conferences, 2009.
[35] Wikipedia. Truncated normal distribution 〈en.wikipedia.org/wiki/Truncated_normal_distribution〉, 2009.
[36] Zhao Y-G, Ono T. Moment methods for structural reliability. Struct Saf 2001;23:47–75.