2011

A Computational Study of the Buffered Failure Probability in Reliability-Based Design Optimization

Basova, H.G.; Rockafellar, R.T.; Royset, J.O.

43. H.G. Basova, R.T. Rockafellar, and J.O. Royset, 2011, A Computational Study of the Buffered Failure Probability in Reliability-Based Design Optimization; Proceedings of the 11th International Conference on Application of Statistics and Probability in Civil Engineering, Zurich, Switzerland, 2011.

http://hdl.handle.net/10945/38219

A Computational Study of the Buffered Failure Probability in Reliability-Based Design Optimization

H.G. Basova
Turkish Army
Ankara, Turkey

R.T. Rockafellar
Department of Mathematics, University of Washington, Seattle, WA, USA
Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, USA.

J.O. Royset
Operations Research Department
Naval Postgraduate School, Monterey, CA, USA.

ABSTRACT: The buffered failure probability is an alternative measure of reliability that offers several theoretical, practical, and computational advantages over the traditional failure probability. It 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. This paper examines the difference between the buffered failure probability and the failure probability in several examples and find that the buffered failure probability typically overestimates the failure probability of a structure with a factor of three. We examine the use of the buffered failure probability in reliability-based optimal design and present three algorithms for the solution of the resulting optimization problems. Computational results on six engineering design examples indicate that the problems are solvable in few seconds using standard optimization solvers.

1 INTRODUCTION

Engineering structures are subject to uncertain loads, environmental conditions, material properties, and geometry that must be accounted for in the design, maintenance, and retrofit of such structures. The theory of structural reliability, see, e.g., Ditlevsen and Madsen (1996), 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 used by researchers, designers, and building code developers as a tool for assessing and comparing designs (Ditlevsen and Madsen 1996). While the failure probability is of significant importance, it also possesses troublesome properties that raise several theoretical, practical, and computational issues. First, it only considers two possible states of the structure: failed, i.e., a performance threshold is violated, and safe, i.e., the threshold is not violated. The degree of violation is of no importance within this framework. Second, the exact computation of the failure probability is rarely possible and commonly used geometric approximations such as the first-order and second-order reliability methods have unknown accuracy and may leave serious design risk undetected. Third, the sensitivity of the failure probability or its approximations with respect to parameters may be poorly behaving and difficult to compute even if the underlying model of the structure is differentiable with respect to parameters. Fourth, it is unknown whether the failure probability and its approximations are convex as functions of parameters. For this reason, it may be difficult to obtain a globally optimal design of optimization problems involving the failure probability or its approximations due to their many local minima that are not globally optimal. We refer to Rockafellar and Royset (2010) for a detailed discussion of these issues.

The *buffered failure probability* is an alternative measure of reliability introduced in Rockafellar and Royset (2010) that offers several 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.

This paper summarizes key properties of the buffered failure probability and compares it with the failure probability. Sections 2 and 3 present the failure and buffered failure probabilities, respectively, and are based on Rockafellar and Royset (2010). Section 4 compares the probabilities analytically and numerically. Section 5 presents computational methods for solving design optimization problems.

2 FAILURE PROBABILITY

The failure and buffered failure probabilities are defined in terms a *limit-state function* $g(\mathbf{x}, \mathbf{v})$ that is a function of a vector $\mathbf{x} = (x_1, x_2, ..., x_n)'$ 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 $\mathbf{v} = (v_1, v_2, ..., v_m)'$ 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 $\mathbf{V} = (V_1, V_2, ..., V_m)'$ 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.

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." The traditional approach to structural reliability defines the failure probability of a structure as the probability of such a failure event. As the failure probability depends on the design $\mathbf{x}$, we denote it by $p(\mathbf{x})$. That is,

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

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. We refer to Rockafellar and Royset (2010) for generalizations of the failure and buffered failure probabilities to the case of structural systems with multiple limit-state functions.

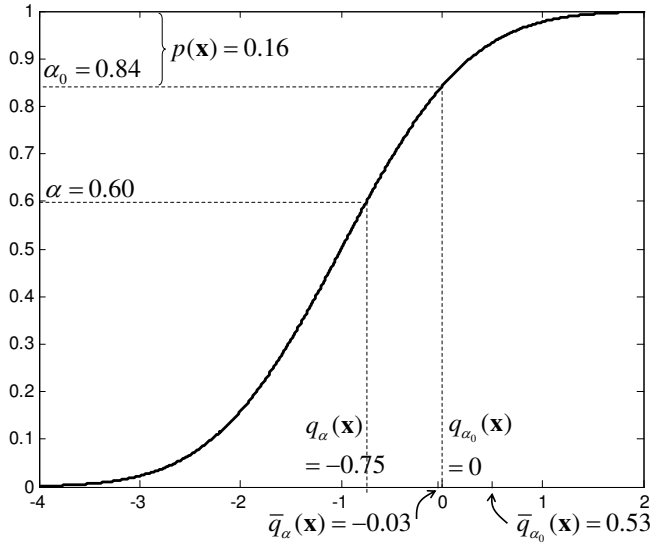

Figure 1: Cumulative distribution function (cdf) of $g(\mathbf{x}, \mathbf{V})$ with examples of α-quantile $q_\alpha(\mathbf{x})$ and α-superquantile $\bar{q}_\alpha(\mathbf{x})$ when normally distributed with mean -1 and standard deviation 1.

3 BUFFERED FAILURE PROBABILITY

In this section, we discuss the buffered failure probability, an alternative to the failure probability, which offers several advantages. The buffered failure probability relates to the conditional value-at-risk (Rockafellar and Uryasev 2000, Rockafellar and Uryasev 2002), which is now widely used under the acronym CVaR in the area of financial engineering to assess investment portfolios. In that area, a quantile is usually called value-at-risk (VaR). Rockafellar (2007) provides a tutorial on CVaR including relation to safety margins and potential replacements for failure probability constraints.

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 Rockafellar and Uryasev (2002). 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 $\mathbf{x}$. Figure 1 illustrates $q_\alpha(\mathbf{x})$ (and $\bar{q}_\alpha(\mathbf{x})$ to be defined below) for the case when $g(\mathbf{x}, \mathbf{V})$ is normally distributed with mean -1 and standard deviation 1. Figure 1 shows the cumulative distribution function of $g(\mathbf{x}, \mathbf{V})$ in this case and quantiles corresponding to probability levels $\alpha = 0.60$ and $\alpha_0 = 0.84$. In view of Figure 1 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 Figure 1 we find that $\alpha_0 = 0.84$ gives $q_{\alpha_0}(\mathbf{x}) = 0$. Hence, $p(\mathbf{x}) = 1 - \alpha_0 = 1 - 0.84 = 0.16$.

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

$$\bar{q}_\alpha(\mathbf{x}) = E[g(\mathbf{x}, \mathbf{V}) | g(\mathbf{x}, \mathbf{V}) \geq q_\alpha(\mathbf{x})]. \tag{2}$$

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 is called CVaR in financial engineering, but we here use the application-independent name superquantile. Figure 1 illustrates 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,

$$\bar{q}_\alpha(\mathbf{x}) = \mu + \frac{\sigma \phi(q_\alpha)}{1 - \alpha}, \tag{3}$$

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. It is clear from the definitions that $q_\alpha(\mathbf{x}) \leq \bar{q}_\alpha(\mathbf{x})$ for any probability level α and design $\mathbf{x}$.

Figure 1 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 the average value of $g(\mathbf{x}, \mathbf{V})$ conditioned on $g(\mathbf{x}, \mathbf{V})$ being larger than -0.75. In this case, that value is -0.03 as computed by (3). Similarly, for probability level 0.84, the corresponding quantile is 0 and the corresponding superquantile is 0.53.

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

$$\bar{q}_\alpha(\mathbf{x}) = 0. \tag{4}$$

That is,

$$\bar{p}(\mathbf{x}) = P[g(\mathbf{x}, \mathbf{V}) \geq q_\alpha(\mathbf{x})], \tag{5}$$

where α is selected such that (4) holds. Hence, $\bar{q}_{1-\bar{p}(\mathbf{x})}(\mathbf{x}) = 0$. We see from Figure 1 that the probability levels $\alpha = 0.60$, which leads to $\bar{q}_\alpha(\mathbf{x}) = -0.03$, and $\alpha_0 = 0.84$, which leads to $\bar{q}_{\alpha_0}(\mathbf{x}) = 0.53$, are slightly too small and much too large, respectively, to result in a corresponding superquantile of zero. However, it is easy to find by trial-and-error and (3) that a probability level $\alpha = 0.62$ results in a quantile of -0.70 and a superquantile of approximately zero. (We present a much easier way than trial-and-error below for computing the superquantile.) By definition, see (5), the buffered failure probability is then $1 - \alpha = 1 - 0.62 = 0.38$, which is somewhat larger than the failure probability of 0.16.

In general, we find that

$$p(\mathbf{x}) \leq \bar{p}(\mathbf{x}) \tag{6}$$

for any $\mathbf{x}$ (Rockafellar and Uryasev 2000, Rockafellar and Uryasev 2002, Rockafellar 2007). 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.

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

$$p(\mathbf{x}) \leq 1 - \alpha_0. \tag{7}$$

In view of Section 1 and Rockafellar and Royset (2010), we note that standard optimization algorithms may have substantial difficulties on problems with constraints of the form (7). We now show that the alternative constraint

$$\bar{p}(\mathbf{x}) \leq 1 - \alpha_0 \tag{8}$$

in terms of the buffered failure probability is much easier to handle. We start by noting that a design $\mathbf{x}$ that satisfies (8) also satisfies (7). Hence, (8) is a conservative requirement.

The ease with which (8) can be handled in optimization algorithms clearly hinges on our ability to evaluate $\bar{p}(\mathbf{x})$ or equivalent expressions. While $\bar{p}(\mathbf{x})$ cannot be expressed explicitly, there is a convenient, equivalent expression for (8) that we derive next.

In view of the definition of the buffered failure probability, we see that (8) holds if and only if

$$\bar{q}_{\alpha_0}(\mathbf{x}) \leq 0. \tag{9}$$

It is shown in Rockafellar and Uryasev (2000) that

$$\bar{q}_{\alpha}(\mathbf{x}) = \min_{z_0} \eta_{\alpha}(z_0, \mathbf{x}), \tag{10}$$

where z_0 is an auxiliary design variable and

$$\eta_{\alpha}(z_0, \mathbf{x}) = z_0 + \frac{1}{1-\alpha} E[\max\{0, g(\mathbf{x}, \mathbf{V}) - z_0\}]. \tag{11}$$

We do not include a derivation of this expression as it is somewhat involved and refer the interested reader to Rockafellar and Uryasev (2000). Hence, the task of finding a design $\mathbf{x}$ that satisfies $\bar{p}(\mathbf{x}) \leq 1 - \alpha_0$ is equivalent to finding a design $\mathbf{x}$ and an auxiliary variable z_0 such that

$$\eta_{\alpha_0}(z_0, \mathbf{x}) \leq 0. \tag{12}$$

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 $p(\mathbf{x}) \leq 1 - \alpha_0$ 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

$$\mathbf{P}: \min_{\mathbf{x} \in X} f(\mathbf{x}) \quad \text{s.t.} \quad p(\mathbf{x}) \leq 1 - \alpha_0. \tag{13}$$

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

$$\mathbf{BP}: \min_{\mathbf{x} \in X, z_0} f(\mathbf{x}) \quad \text{s.t.}$$

$$z_0 + \frac{1}{1-\alpha_0} E[\max\{0, g(\mathbf{x}, \mathbf{V}) - z_0\}] \leq 0$$

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

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

$$\mathbf{BP'}_N: \min_{\mathbf{x} \in X, z_0} f(\mathbf{x}) \quad \text{s.t.}$$

$$z_0 + \frac{1}{N(1-\alpha_0)} \sum_{j=1}^{N} \max\{0, g(\mathbf{x}, \mathbf{v}^j) - z_0\} \leq 0 \tag{14}$$

is an approximation of the problem **BP**. Even if the limit-state function $g(\mathbf{x}, \mathbf{v})$ is continuously differentiable with respect to $\mathbf{x}$ 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 $\mathbf{BP'}_N$. However, an equivalent transcription of $\mathbf{BP'}_N$ facilitates the use of standard nonlinear optimization algorithms.

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

$$\min_{\mathbf{x} \in X, \bar{\mathbf{z}}} f(\mathbf{x}) \quad \text{s.t.}$$

$$z_0 + \frac{1}{N(1-\alpha_0)} \sum_{j=1}^{N} z_j \leq 0$$

$$\max\{0, g(\mathbf{x}, \mathbf{v}^j) - z_0\} = z_j, \quad j = 1, 2, ..., N,$$

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 $\max\{0, g(\mathbf{x}, \mathbf{v}^j) - z_0\} < z_j$ for any $j = 1, 2, ..., N$. Moreover, a constraint of the form $\max\{0, g(\mathbf{x}, \mathbf{v}^j) - z_0\} \leq z_j$ is equivalent to the two constraints $g(\mathbf{x}, \mathbf{v}^j) - z_0 \leq z_j$ and $0 \leq z_j$. This leads to the following equivalent problem of $\mathbf{BP'}_N$:

$$\mathbf{BP}_N: \min_{\mathbf{x} \in X, \bar{\mathbf{z}}} f(\mathbf{x}) \quad \text{s.t.}$$

$$z_0 + \frac{1}{N(1-\alpha_0)} \sum_{j=1}^{N} z_j \leq 0$$

$$g(\mathbf{x}, \mathbf{v}^j) - z_0 \leq z_j, \quad j = 1, 2, ..., N \tag{15}$$

$$z_j \geq 0, \quad j = 1, 2, ..., N.$$

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

4 COMPARISON OF PROBABILITIES

We next compare the buffered failure probability and the failure probability in detail.

4.1 *Advantages of the Buffered Failure Probability*

As mentioned above and discussed in Rockafellar and Royset (2010), the buffered failure probability compares favorable with the failure probability in several aspects. First, we find it 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 **P** to be easily solvable. Moreover, as we see below, there are also efficient algorithms that tackle $\mathbf{BP}'_N$ directly. The optimal values of $\mathbf{BP}_N$ and $\mathbf{BP}'_N$ are close to the optimal value of **BP** when N is large; see Chapter 5 of Shapiro et al. (2009). In addition, **BP** is a restriction of **P** because the buffered failure probability overestimates the failure probability; see (6). Hence, a feasible design in **BP** is also feasible in **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 $\mathbf{BP}_N$ and $\mathbf{BP}'_N$ may be more desirable than those from **P**.

Third, even if $g(\mathbf{x}, \mathbf{v})$ is convex in $\mathbf{x}$, $p(\mathbf{x})$ may not be and, hence, it may be difficult to obtain a globally optimal design of **P**. In contrast, the region defined by the constraints (15) is convex when $g(\mathbf{x}, \mathbf{v}^j)$, $j = 1, 2, ..., N$, are convex functions in $\mathbf{x}$. Hence, every Karush-Kuhn-Tucker point of $\mathbf{BP}_N$ is a globally optimal design when $f(\mathbf{x})$ and $g(\mathbf{x}, \mathbf{v}^j)$, $j = 1, 2, ..., N$, are convex functions and the region X is a convex set. Consequently, $\mathbf{BP}_N$ "preserves" convexity. Even if not all of these conditions are satisfied, we expect it to often be easier to determine a design with a lower objective function value in $\mathbf{BP}_N$ than in **P** because $\mathbf{BP}_N$ deals with $g(\mathbf{x}, \mathbf{v})$ directly instead of the more complex expression $p(\mathbf{x})$.

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 **P** is often replaced by response surface and surrogate models (Gasser and Schueller 1998, Torczon and Trosset 1998, Viana et al. 2009). 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 (Rockafellar and Royset 2010).

4.2 *Analytic Comparison of Probabilities*

While we stress that the buffered failure probability carries more information about a structure's reliability than the failure probability, we may also view the buffered failure probability as an approximation of the failure probability. From (6), we see that the buffered failure probability is a conservative approximation. However, in general, it is difficult to say with how much the buffered failure probability overestimates the failure probability. Still, in some special cases, we can relatively easily examine the differences. We consider one such case next.

Suppose that the limit-state function is affine in $\mathbf{x}$ with an intercept that is normally distributed with mean μ and standard deviation σ, i.e., $g(\mathbf{x}, \mathbf{V}) = \mathbf{a}'\mathbf{x} + V$, where $\mathbf{V}$ is a single normally distributed random variable V with mean μ and standard deviation σ. Then, the failure probability takes the form

$$p(\mathbf{x}) = \text{Prob}[\mathbf{a}'\mathbf{x} + V > 0] = \Phi((\mathbf{a}'\mathbf{x} + \mu)/\sigma), \qquad (16)$$

where $\Phi(\cdot)$ is the cumulative standard normal distribution function. Hence, the constraint $p(\mathbf{x}) \leq 1 - \alpha_0$ simplifies to

$$\mathbf{a}'\mathbf{x} + \mu + \sigma\Phi^{-1}(\alpha_0) \leq 0. \qquad (17)$$

In comparison, the buffered failure probability $\bar{p}(\mathbf{x}) \leq 1 - \alpha_0$ simplifies as follows. Since $g(\mathbf{x}, \mathbf{V})$ is normally distributed with mean $\mathbf{a}'\mathbf{x} + \mu$ and standard deviation σ, in view of (3), the superquantile

$$\bar{q}_{\alpha_0}(\mathbf{x}) = \mathbf{a}'\mathbf{x} + \mu + \frac{\sigma\phi(\Phi^{-1}(\alpha_0))}{1 - \alpha_0}. \qquad (18)$$

Since $\bar{p}(\mathbf{x}) \leq 1 - \alpha_0$ if and only if $\bar{q}_{\alpha_0}(\mathbf{x}) \leq 0$, this buffered failure probability constraint simplifies to

$$\mathbf{a}'\mathbf{x} + \mu + \frac{\sigma\phi(\Phi^{-1}(\alpha_0))}{1 - \alpha_0} \leq 0. \qquad (19)$$

Comparing (17) and (19), we see the exact difference between the failure probability and buffered failure probability constraints in this case and that the difference varies with σ and α_0. Wang and Ahmed (2008) present a similar comparison for the case with $g(\mathbf{x}, \mathbf{V}) = \mathbf{V}'\mathbf{x}$ and $\mathbf{V}$ normally distributed. We next examine numerically the difference between the probabilities in the case of engineering applications.

4.3 *Computational Examples*

We consider six engineering design examples from the literature and compare the difference between the failure and buffered failure probabilities. Example 1 is a simple problem instance with two design variables and two random variables constructed by Hock and Schittkowski (1981). Example 2 involves the design of the thickness and width of a cantilever beam subject to random yield stress, Young's module, and horizontal and vertical loads, as described by Eldred and Bichon (2006). Example 3 deals with the design of the cross-section width and depth of a short column subject to random axial force, bending moment, and yield stress, and is given in Bichon et al. (2009). Example 4 focuses on determining the diameter and thickness of a tubular column under a random load; see (Rao 2009), pp. 10-14. Example 5 involves the design of

Table 1: Design examples with number of design variables (DV) and random variables (RV) listed.

Ex.	Description	DV	RV
1	Analytical (Hock and Schittkowski 1981)	2	2
2	Cantilever (Eldred and Bichon 2006)	2	4
3	Short column (Bichon et al. 2009)	2	3
4	Tubular column (Rao 2009), pp. 10-14	2	1
5	Speed reducer (Rao 2009), pp. 472-473	7	7
6	Vehicle design (Samson et al. 2009)	11	7

Table 2: Estimated failure and buffered failure probabilities for specific optimized designs.

Ex.	Failure probability	Buffered failure prob.
1	[0.00047, 0.00057]	0.00133
2	[0.0000004, 0.0000016]	0.0010150
3	[0.00047, 0.00057]	0.00140
4	[0.00033, 0.00041]	0.00097
5	[0.00042, 0.00052]	0.00449
6	[0.00028, 0.00034]	0.00138

a speed reducer under random material properties as given in (Rao 2009), pp. 472-473. The design variables include face width, module of teeth, number of teeth on pinion, and length and diameter of shafts. Example 6 is taken from Samson et al. (2009) and deals with the design of a motor vehicle under a side-impact crash. The design variables include thickness of pillar, floor side, cross member, door beam and door belt line. All design variables are subject to production uncertainties. Table 1 gives an overview of the examples and lists the number of design variables (DV) and random variables (RV); see also Basova (2010). These examples involve multiple limit-state functions and, hence, we consider the generalization of the failure and buffered failure probabilities to the *system* failure probability and the *system* buffered failure probability as described in Rockafellar and Royset (2010). The examples include a mix of explicitly given linear and nonlinear objective and limit-state functions. Table 2 presents 95% confidence intervals for the failure probability (column 2) for particular designs $\mathbf{x}$ in the six examples. The particular designs are optimized designs as given in Basova (2010). Here, we only aim to illustrate typical differences between the failure and buffered failure probabilities and omit the numerical values of the designs. The confidence intervals are computed in the standard manner by sequentially generating sample points of $\mathbf{V}$ until the width of the confidence interval is less than 10%. Column 3 of Table 2 presents point estimates of the buffered failure probability in each example for the same design as the one used to estimate the failure probability. In each example, we use the same sample as the one that generated the estimate of the failure probability. Specifically, if $\mathbf{v}^1, \mathbf{v}^2, ..., \mathbf{v}^N$ is that sample, then the estimate of the buffered failure probability is simply $1-\alpha$, where α is the probability level that approximately ensures that the empirical estimate of the superquantile $\bar{q}_\alpha(\mathbf{x})$ is zero; see (4). Hence, $1-\alpha$ is computed as follows. First, we order the the values $\{g(\mathbf{x}, \mathbf{v}^j\}_{j=1}^N$ in decreasing order. Without loss of generality, suppose that $g(\mathbf{x}, \mathbf{v}^1) \geq g(\mathbf{x}, \mathbf{v}^2) \geq ... \geq g(\mathbf{x}, \mathbf{v}^N)$. Second, we computing the smallest integer k such that $\sum_{j=k}^N g(\mathbf{x}, \mathbf{v}^j)/(N-k+1) \leq 0$. Third, we obtain the estimate of the buffered failure probability by $k/(N-1)$. Table 2 shows that the buffered failure probability typically overestimates the failure probability of the structures by a factor of three.

5 COMPUTATIONAL METHODS

In this section, we discuss some relatively simple algorithms for solving design optimization problems of the form **BP**. Since the expectation in this problem is rarely computable exactly, we focus on that problem's sample-average approximation $\mathbf{BP}'_N$, where the expectation is approximated by its sample average. We present three algorithms for solving $\mathbf{BP}'_N$ for a fixed N. These algorithm are attractive due to their simplicity and the fact that they can easily be implemented using standard optimization solvers such as SNOPT (Gill et al. 1998), LANCELOT (Conn et al. 1992), and NLPQL (Schittkowski 1985) when the limit-state function $g(\mathbf{x}, \mathbf{v})$ is continuously differentiable with respect to $\mathbf{x}$. They are also guaranteed to generate globally optimal, locally optimal, and stationary points of **BP** under relatively mild assumptions as $N \to \infty$. We refer to Xu and Zhang (2009), Royset (2010), and Chung et al. (2010) for the theoretical basis for these algorithms. Related algorithms based on nonsmooth optimization are discussed by Beliakov and Bagirov (2006), Lim et al. (2010), Iyengar and Ma (2010), and Ogryczak and Sliwinski (2010). Algorithms based on problem decomposition are discussed by Fabian (2008) and Kunzi-Bay and Mayer (2006), and algorithms based on smoothing by Tong et al. (2010). For algorithms based on adaptive selection of the sample size N, we refer to Basova (2010) and Royset (2010).

We next describe the three algorithms in turn.

5.1 *Algorithm 1: Expansion method*

Algorithm 1 for solving $\mathbf{BP}'_N$, first presented in (Rockafellar & Uryasev 2000), considers the variable- and constrained-expanded problem $\mathbf{BP}_N$ and apply a standard nonlinear optimization algorithm to that expanded problem. While this method is simple, it suffers from the drawback that the number of variables and constraints grows linearly in the sample size N. Hence, for large N, it may be impossible to solve $\mathbf{BP}_N$ or the solution may require specialized hardware and software.

5.2 *Algorithm 2: Active-set method*

Algorithm 2 is a specialized version of an algorithm by Chung et al. (2010) and considers only a subset of sample points, a so-called active set, in most iterations. Hence, it may overcome the difficulty faced by

the expansion method. Specifically, Algorithm 2 consists of applying a standard optimization solver to the reduced problem

$$\mathbf{BP}_N(W): \min_{\mathbf{x}\in X, z_j, j\in W} f(\mathbf{x}) \quad \text{s.t.}$$

$$z_0 + \frac{1}{N(1-\alpha_0)} \sum_{j\in W} z_j \leq 0$$

$$g(\mathbf{x}, \mathbf{v}^j) - z_0 \leq z_j, \quad j \in W \quad (20)$$

$$z_j \geq 0, \quad j \in W,$$

where W is a subset of $\{1, 2, ..., N\}$. Given an initial design $\mathbf{x}^0$ and initial auxiliary variables $\bar{\mathbf{z}}^0 = (z_0^0, z_1^0, ..., z_N^0)'$, the active set W first consists of those sample point indices j such that $g(\mathbf{x}^0, \mathbf{v}^j) - z_0^0 - z_j^0 \geq \max_{i=1,2,...,N}\{g(\mathbf{x}^0, \mathbf{v}^i) - z_0^0 - z_i^0\} - \epsilon$, where $\epsilon > 0$ is a parameter. Hence, W consists of sample point indices that yield nearly active constraints in $\mathbf{BP}_N$. Next, Algorithm 2 carries out a given number of iterations, say n, of a standard optimization solver on $\mathbf{BP}_N(W)$. This yields a new design $\mathbf{x}^1$ and auxiliary variables $\bar{\mathbf{z}}^1$. We refer to the process of determining W and carrying out n solver iterations as one *major iteration*. Algorithm 2 then starts another major iteration by augmenting the active set with any sample point index j that satisfies $g(\mathbf{x}^1, \mathbf{v}^j) - z_0^1 - z_j^1 \geq \max_{i=1,2,...,N}\{g(\mathbf{x}^1, \mathbf{v}^i) - z_0^1 - z_i^1\} - \epsilon$. Algorithm 2 proceeds similarly with alternating the calculations of n iterations of the standard optimization solver applied to $\mathbf{BP}_N(W)$ and the updating of W using the latest design and auxiliary variables. Since the cardinality of W may be smaller than N, Algorithm 2 may have a smaller memory requirement than Algorithm 1 and, as seen below, its run time may also be shorter.

5.3 *Algorithm 3: Smoothing method*

Algorithm 3 overcomes the difficulty associated with the nonsmooth max-function in $\mathbf{BP}'_N$ by smoothing instead of by expansion of the number of variables and constraints as in Algorithms 1 and 2. Specifically, it can be shown that the problem

$$\mathbf{BP}_{Nq}: \min_{\mathbf{x}\in X, z_0} f(\mathbf{x}) \quad \text{s.t.}$$

$$z_0 + \frac{1}{N(1-\alpha_0)} \sum_{j=1}^{N} \psi_q^j(\mathbf{x}, z_0) \leq 0, \quad (21)$$

where

$$\psi_q^j(\mathbf{x}, z_0) = \frac{1}{q} \ln(1 + \exp(q(g(\mathbf{x}, \mathbf{v}^j) - z_0))), \quad (22)$$

approximates in some sense $\mathbf{BP}'_N$ for large value of the smoothing parameter $q > 0$; see Alexander et al. (2006), Xu and Zhang (2009), and Royset (2010) for theoretical results and application of this approach. Specifically, (21) is a smooth approximation of (14). It is easy to show that the gradient of $\psi_q^j(\mathbf{x}, z_0)$ is given by

$$\nabla\psi_q^j(\mathbf{x}, z_0) = \mu_q^j(\mathbf{x}, z_0)(\nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{v}^j), -1)', \quad (23)$$

where $\mu_q^j(\mathbf{x}, z_0) = \exp(q(g(\mathbf{x}, \mathbf{v}^j) - z_0))/(1 + \exp(q(g(\mathbf{x}, \mathbf{v}^j) - z_0)))$. $\mathbf{BP}_{Nq}$ involves only one auxiliary design variable z_0 and one constraint (in addition to the constraints abstractly represented by X) regardless of the sample size N. Hence, its memory requirement is moderate. Algorithm 3 simply involves applying a standard optimization solver to $\mathbf{BP}_{Nq}$ for a given value of q. We next compare the run times of Algorithms 1-3 on Examples 1-6.

Table 3: Run times in seconds for Algorithms 1-3 on Examples 1-6.

Ex.	Algorithm 1	Algorithm 2	Algorithm 3
1	1406.1	0.7	15.9
2	1966.6	199.4	42.3
3	1020.8	1.1	30.3
4	435.8	9.8	56.9
5	1500.0	4.3	156.1
6	118.0	1.9	68.8

5.4 *Numerical results*

We implement Algorithms 1-3 in MATLAB with TOMLAB/SNOPT (Holmstrom, 1999) as optimization solver. The computations are run on a desktop computer with 3.25 GB RAM and 3.16 GHz processor speed under Windows XP. We consider **BP** for Examples 1-6 with $1 - \alpha_0 = 0.001349898$, which corresponds to the -3 quantile of the standard normal distribution. In addition to a constraint on the buffered failure probability, Examples 1-6 include bounds on the decision variables as described by Basova (2010) and the original references given above. We use sample size $N = 10000$, parameters $\epsilon = 0.001$ and $n = 5$ in Algorithm 2, and parameter $q = 1000$ in Algorithm 3. We refer to Basova (2010) for a discussion of these parameter choices.

Table 3 gives run times to termination for Algorithms 1-3 in seconds on Examples 1-6. In the case of Algorithms 1 and 3, the calculations are terminated when the default stopping criterion in SNOPT is satisfied. For Algorithm 2, the calculations terminate at major iteration k if $g(\mathbf{x}^k, \mathbf{v}^j) - z_0^k \leq z_j^k$ for all $j = 1, 2, ..., N$ and $|\max_{j=1,2,...,N}\{g(\mathbf{x}^k, \mathbf{v}^j) - z_0^k - z_j^k\} - \max_{j=1,2,...,N}\{g(\mathbf{x}^{k-1}, \mathbf{v}^j) - z_0^{k-1} - z_j^{k-1}\}| \leq 10^{-6}$. That is, the calculations terminate if the current design is feasible with respect to all samples and the difference between two consecutive major iterations is small as measured by the constraints.

As reported by Basova (2010), Algorithms 1-3 obtain essentially identical optimized designs. We find from Table 3, however, that the run times of Algorithm 1 are one to three orders of magnitude slower

than those of Algorithms 2 and 3 due to the large number of variables and constraints that need to be handled. Except in Example 2, Algorithm 2 is faster than Algorithm 3 with run times of just a few seconds.

6 CONCLUSIONS

We compared the buffered failure probability, which is an alternative measure of structural reliability, with the traditional failure probability and find that it offers significant advantages. The buffered failure probability accounts for the degree of violation of a performance threshold, is handled with relative ease in design optimization problems, and is more conservative than the failure probability. The degree of conservativeness is moderate: the buffered failure probability overestimates the failure probability of a structure by a factor of three in several engineering design examples. The paper presents three algorithms for solving optimal design problems involving the buffered failure probability. Two of the algorithms solve six engineering design examples in few seconds.

ACKNOWLEDGEMENT

The third author acknowledges financial supported from Air Force Office of Scientific Research Young Investigator grant F1ATA08337G003.

REFERENCES

Alexander, S., T. Coleman, & Y. Li (2006). Minimizing CVaR and VaR for a portfolio of derivatives. *J. Banking & Finance 30*, 583–605.

Basova, H. G. (2010). *Reliability-based design optimization using buffered failure probability*. Master's thesis, Naval Postgraduate School, Monterey, California.

Beliakov, C. & A. Bagirov (2006). Non-smooth optimization methods for computation of the conditional value-at-risk and portfolio optimization. *Optimization 55*(5-6), 459–479.

Bichon, B. J., S. Mahadevan, & M. S. Eldred (2009). Reliability-based design optimization using efficient global reliability analysis. In *Proceedings of the 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Material Conference*, pp. AIAA 2009–2261.

Chung, H., E. Polak, & S. Sastry (2010). On the use of outer approximations as an external active set strategy. *J. Optimization Theory and Applications 146*(1), 51–75.

Conn, A. R., N. I. M. Gould, & P. Toint (1992). *LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A)*. Heidelberg, Germany: Springer. Springer Series in Computational Mathematics, Vol. 17.

Ditlevsen, O. & H. O. Madsen (1996). *Structural reliability methods*. New York, New York: Wiley.

Eldred, M. S. & B. J. Bichon (2006). Second order reliability formulations in dakota/uq. In *Proceedings of the 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Material Conference*, pp. AIAA 2006–1828.

Fabian, C. I. (2008). Handling CVaR objectives and constraints in two-stage stochastic models. *European J. Operational Research 191*, 888–911.

Gasser, M. & G. I. Schueller (1998). Some basic principles in reliability-based optimization (RBO) of structures and mechanical components. In *Stochastic programming methods and technical applications*, K. Marti and P. Kall (Eds.), Lecture Notes in Economics and Mathematical Systems 458, Springer, Berlin, Germany.

Gill, P., W. Murray, & M. Saunders (1998). 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, California.

Hock, W. & K. Schittkowski (1981). *Test examples for nonlinear programing codes*. Springer.

Iyengar, G. & A. K. C. Ma (2010). Fast gradient descent method for mean-CVaR optimization. *Operations Research Letters*.

Kunzi-Bay, A. & J. Mayer (2006). Computational aspects of minimizing conditional value-at-risk. *Computational Management Science 3*, 3–27.

Lim, C., H. D. Sherali, & S. Uryasev (2010). Portfolio optimization by minimizing conditional value-at-risk via nondifferentiable optimization. *Computational Optimization and Applications 46*(3), 391–415.

Ogryczak, W. & T. Sliwinski (2010). On solving the dual for portfolio selection by optimizing conditional value at risk. *Computational Optimization and Applications*.

Rao, S. S. (2009). *Engineering optimization theory and practice* (4th ed.). John Wiley & Sons.

Rockafellar, R. T. (2007). Coherent approaches to risk in optimization under uncertainty. In *Tutorials in Operations Research*, pp. 39–61. INFORMS.

Rockafellar, R. T. & J. O. Royset (2010). On buffered failure probability in design and optimization of structures. *Reliability Engineering & System Safety 95*, 499–510.

Rockafellar, R. T. & S. Uryasev (2000). Optimization of conditional value-at-risk. *Journal of Risk 2*, 21–42.

Rockafellar, R. T. & S. Uryasev (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance 26*, 1443–1471.

Royset, J. O. (2010). On optimality functions in stochastic programming and applications. *Available at: http://faculty.nps.edu/joroyset/docs/Royset_optimfcn.pdf*.

Samson, S., S. Thoomu, G. Fadel, & J. Reneke (2009). Reliable design optimization under aleatory and epistemic uncertainty. In *Proceedings of ASME 2009 International Design Engineering Technical Conferences*, pp. DETC2009–86473.

Schittkowski, K. (1985). *User's guide to nonlinear programming code, handbook to optimization program package NLPQL*. Stuttgart, Germany: University of Stuttgart.

Shapiro, A., D. Dentcheva, & A. Ruszczynski (2009). *Lectures on Stochastic Programming: Modeling and Theory*. Society of Industrial and Applied Mathematics.

Tong, X., L. Qi, F. Wu, & H. Zhou (2010). A smoothing method for solving portfolio optimization with CVaR and applications in allocation of generation asset. *Applied Mathematics and Computation 216*, 1723–1740.

Torczon, V. & M. W. Trosset (1998). Using approximations to accelerate engineering design optimization. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symp. on Multidisciplinary Analysis and Optimization*, AIAA Paper 98-4800, St. Louis, Missouri.

Viana, F., V. Picheny, & R. Haftka (2009). 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 & Computers and Information in Engineering Conferences*.

Wang, W. & S. Ahmed (2008). Sample average approximation of expected value constrained stochastic programs. *Operations Research Letters 36*(5), 515–519.

Xu, H. & D. Zhang (2009). Smooth sample average approximation of stationary points in nonsmooth stochastic optimization and applications. *Mathematical Programming 119*, 371–401.