Struct Multidisc Optim (2017) 55:1535–1539
DOI 10.1007/s00158-016-1587-y

BRIEF NOTE

A derivative-free trust-region algorithm for reliability-based optimization

Tian Gao[1] · **Jinglai Li**[2]

Received: 3 June 2016 / Revised: 4 September 2016 / Accepted: 6 September 2016 / Published online: 24 September 2016

Abstract In this note, we present a derivative-free trust-region (TR) algorithm for reliability based optimization (RBO) problems. The proposed algorithm consists of solving a set of subproblems, in which simple surrogate models of the reliability constraints are constructed and used in solving the subproblems. Taking advantage of the special structure of the RBO problems, we employ a sample reweighting method to evaluate the failure probabilities, which constructs the surrogate for the reliability constraints by performing only a single full reliability evaluation in each iteration. With numerical experiments, we illustrate that the proposed algorithm is competitive against existing methods.

Keywords Derivative free · Trust region · Monte Carlo · Reliability based optimization

This work was supported by the NSFC under grant number 11301337 and by the National Basic Research Program (973 Program) of China under grant number 2014CB744900.

✉ Jinglai Li
jinglaili@sjtu.edu.cn

[1] Department of Mathematics, Shanghai Jiao Tong University, 800 Dongchuan Rd, Shanghai 200240, China

[2] Institute of Natural Sciences, Department of Mathematics, and the MOE Key Laboratory of Scientific and Engineering Computing, Shanghai Jiao Tong University, 800 Dongchuan Rd, Shanghai 200240, China

1 Introduction

Reliability based optimization (RBO) problems, which optimize the system performance subject to the constraint that the system reliability satisfies a prescribed requirement, are an essential task in many engineering design problems (Valdebenito and Schuëller 2010; Aoues and Chateauneuf 2010). In a standard RBO problem, the reliability constraint is typically formulated as that the failure probability of the system is lower than a threshold value, and a very common class of RBO problems is to minimize a cost function subject to the failure probability constraint:

$$\min_{\mathbf{x}\in D} f(\mathbf{x}), \quad s.t.\ c(\mathbf{x}) := \ln P(\mathbf{x}) - \ln\theta \leq 0, \tag{1.1}$$

where $\mathbf{x}$ is the design parameter, D is the design space, $f(\cdot)$ is the cost function, $P(\mathbf{x})$ is the failure probability associated with design $\mathbf{x}$ and θ is the failure probability threshold. In practice, the cost function is often deterministic and easy to evaluate, while computing the probabilistic constraint is much more costly as it requires expensive Monte Carlo (MC) simulations.

In this work we consider the so-called double loop (DL) RBO methods, where an inner loop estimating the failure probability is nested in the outer loop solving the optimization problem (Aoues and Chateauneuf 2010; Valdebenito and Schuëller 2010; Eldred et al. 2002), and so other methods, such as the single loop and the decoupling algorithms (Valdebenito and Schuëller 2010) are not in our scope. The DL methods only require to evaluate the limit state function of the underlying system, which makes it particularly convenient for problems with black-box models. The computational burden of the DL methods arises from both the inner and the outer loops. Namely, the total computational

cost depends on the number of reliability (failure probability) evaluations required and the cost for performing each single failure probability evaluation. This work aims to address the former: to solve the RBO problems with a small number of reliability evaluations. A difficulty here is that, due to the use of MC simulations, it is very difficult to obtain the derivatives of the reliability constraints. One way to alleviate the difficulty is to perform stochastic sensitivity analysis with the so-called score functions (SF) (Rahman 2009; Rubinstein and Shapiro 1993). Here we consider an alternative type of methods, known as the derivative-free (DF) trust-region (TR) algorithms (Conn et al. 2009), developed to solve problems whose derivatives are difficult to obtain. Loosely speaking, the DF-TR methods consist of solving a set of TR subproblems in which surrogate models of the objective and/or the constraint functions are constructed and used in solving the subproblems. The main contribution o the work is two-fold. First we present a DF-TR algorithm specifically designed for the RBO problems, which does not require the knowledge of the derivative information of the objective and the constraint functions. Note that the computational cost associated with the DF-TR algorithm poses a challenge here, as constructing a surrogate model with regression or interpolation requires to repeatedly evaluate the reliability constraints, which is highly expensive. Thus our second contribution is to employ a sampling reweighting method, which only uses a *single full reliability evaluation* to construct the surrogates in each TR iteration. With a numerical example, we illustrate that the DF-TR algorithm can be a competitive alternative to the score-function based methods.

The paper is organized as follows. We present our DF-TR algorithm for RBO problems in Section 2. We describe the evaluation of reliability constraints in Section 3. Finally we provide a benchmark example to demonstrate the performance of the proposed algorithm in Section 4.

2 The DF-TR algorithm for RBO problems

2.1 The derivative-free trust-region algorithm

A natural idea to solve the RBO problem (1.1) is to construct a computationally efficient surrogate for the constraint $c(\mathbf{x})$, and then solve the optimization problem subject to the surrogate constraint. The TR methods provide a rigorous formulation of this surrogate based approach. The TR methods start from an initial point $\mathbf{x}_0$ and finds a critical point by computing a series of intermediate points $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$. Specifically, suppose the current point is $\mathbf{x}_k$, and to compute the next point, the algorithms solve a TR subproblem in which surrogates of the objective function and the constraints are constructed and used in a neighborhood of $\mathbf{x}_k$. This neighborhood of $\mathbf{x}_k$ is known as the trust-region and the size of it is adjusted in a way that the surrogate models are sufficiently accurate in it. In our problem, the objective function is of simple form, and we only need to construct the surrogate for the constraint function. As a result, in our RBO problems, the TR sub-problem at iteration k becomes,

$$\min_{\mathbf{x}\in D} f(\mathbf{x}),\ \text{s.t.}\ s_k(\mathbf{x}) \le 0 \,\text{and}\, \|\mathbf{x}-\mathbf{x}_k\| \le \rho_k, \tag{2.1}$$

where $s_k(\mathbf{x})$ is the surrogate model of $c(\mathbf{x})$, and ρ_k is the radius of the TR of $\mathbf{x}_k$. In what follows we use the notation: $\mathcal{O}(\mathbf{x}_c, \rho) = \{\mathbf{x} | \|\mathbf{x}-\mathbf{x}_c\| \le \rho\}$. Before discussing the construction of the surrogate models, we first present our main algorithm for solving the RBO problems:

Algorithm 1 The DF-TR RBO algorithm

Input: $f(\mathbf{x})$, $c(\mathbf{x})$, $\mathbf{x}_0$, ρ_0, $\rho_{\min}$, ω^+, ω^-, δ, ϵ^*, M.
Output: Solution $\mathbf{x}_{\text{opt}}$;

```
Outer:= 1; k := 0;
while Outer= 1 do
    Inner:= 1;
    while Inner= 1 do
        [s_k(x), ρ_k] := SurrConstr(x_k, ρ_k, ε*, ω⁻, M);
        x_{k+1} := arg min_{x∈O(x_k,ρ_k)} f(x), s.t. s_k(x) ≤ 0;
        if c(x_{k+1}) < 0 then
            Inner:= 0;
            ρ_k = ω⁺ρ_k;
        else
            ρ_k := ω⁻ρ_k;
        end if
    end while
    if ‖x_{k+1} − x_k‖ < ρ_k or ‖f(x_{k+1}) − f(x_k)‖ ≤ δ or
  ρ_k < ρ_min then
        Outer:= 0;
    else
        ρ_{k+1} := ρ_k;
        k := k + 1;
    end if
end while
x_opt := x_k;
```

A key step in a TR algorithm is to adjust the radius of the TR in each step. In this respect our algorithm follows the procedure given in Augustin and Marzouk (2014), but only adjusts the radius according to the constraint function (Augustin and Marzouk (2014) adjusts it based on both the objective and the constraint functions). Here ρ_0 is the initial TR radius and ω^+ and ω^- are the TR expansion and contraction constants respectively. The TR subproblem (2.1) can be solved with any usual constrained optimization technique, and in this work we choose to use the

sequential quadratic programming (SQP) method. The algorithm terminates when one of the following three conditions is satisfied: $\mathbf{x}_{k+1}$ is an inner point of $\mathcal{O}(\mathbf{x}_k, \rho_k)$, the difference between $f(\mathbf{x}_k)$ and $f(\mathbf{x}_{k+1})$ is below a prescribed threshold δ, or the radius is smaller than a prescribed minimal value $\rho_{\min}$. Moreover, ϵ^* is the error bound for the surrogate models, and M is the number of points used to construct the surrogate models.

We now discuss the construction of the surrogates, which is a critical step in Algorithm 1. In the DF framework, one first writes the surrogate model as a linear combination of a set of basis functions namely,

$$s(\mathbf{x}) = \sum_{l=1}^{L} a_l b_l(\mathbf{x}), \tag{2.2}$$

where $\{b_l(\mathbf{x})\}_{l=1}^{L}$ are a set of basis functions and $\mathbf{a} = (a_1, ..., a_L)^T$ is the vector collecting all the coefficients, and then determines the coefficients $\mathbf{a}$ with either regression or interpolation. We choose to use the popular *quadratic polynomials* surrogates, while noting that the proposed algorithm does not depend on any particular type of surrogates.

In the standard DF-TR algorithms, the surrogate models are required to be fully linear or quadratic (Conn et al. 2009). Imposing such conditions is very difficult in RBO problems as the failure probability is evaluated with sampling methods. Thus here we simply require that the error between the surrogate and the true constraint function is bounded in the TR: for a given fixed $\epsilon > 0$ and a TR $\mathcal{O}(\mathbf{x}_c, \rho)$, $|s(\mathbf{x}) - c(\mathbf{x})| \leq \epsilon$ for any $\mathbf{x} \in \mathcal{O}(\mathbf{x}_c, \rho)$. Now, we propose a scheme to construct TR surrogate with a bounded error, described as:

Algorithm 2 $[s(\cdot), \rho] = \text{SurrConstr}(\mathbf{x}_c, \rho_{\max}, \epsilon^*, \omega, M)$

1: let $\rho := \rho_{\max}$; LOOP:= 1;
2: **while** LOOP= 1 **do**
3: randomly generate $M-1$ points in $\mathcal{O}(\mathbf{x}, \rho)$: $\{\mathbf{x}_m \in \mathcal{O}(\mathbf{x}, \rho)\}_{m=1}^{M-1}$;
4: let $\mathbf{x}_M = \mathbf{x}_c$;
5: evaluate the constraint function $y_m = c(\mathbf{x}_m)$ for $m = 1...M$.
6: compute $s(\mathbf{x})$ using data set $\{(\mathbf{x}_m, y_m)\}_{m=1}^{M}$;
7: estimate the approximation error bound ϵ of $s(\cdot)$ with leave-one-out cross validation;
8: **if** $\epsilon < \epsilon^*$ **then**
9: LOOP:= 0;
10: **else**
11: $\rho := \omega\rho$;
12: **end if**
13: **end while**
14: **return** $s(\cdot)$ and ρ.

Simply put, the algorithm constructs the quadratic regression and examines whether the resulting surrogate satisfies the error bound condition; if not, the algorithm contracts the TR and repeats. In Line 7, we estimate the approximation error with the leave-one-out cross validation method. Namely, let $X = \{\mathbf{x}_1, ..., \mathbf{x}_M\}$ and $Y = \{y_1, ..., y_M\}$ with $y_m = c(\mathbf{x}_m)$ for $m = 1...M$. Let $X_-^m = \{\mathbf{x}_1, ..., \mathbf{x}_{m-1}, \mathbf{x}_{m+1}, ..., \mathbf{x}_M\}$ and $Y_-^m = \{y_1, ..., y_{m-1}, y_{m+1}, ..., y_M\}$. Let $s^m(\mathbf{x})$ be the surrogate model based on data (X_-^m, Y_-^m) and the approximation error ϵ is estimated by $\epsilon = \max\{|c(\mathbf{x}_m) - s^m(\mathbf{x}_m)|\}_{m=1}^{M}$. Apparently, to construct the surrogate, we need to evaluate the reliability constraint at a rather large number of design points, which can be computationally demanding. However, as will be shown in the next section, we apply a sample reweighting strategy, which allows us to obtain the values of the constraint at all the design points by only performing a full sampling based reliability evaluation at $\mathbf{x}_c$. Thus the computational cost is significantly reduced. We also note that, another way to improve the efficiency for evaluating the reliability constraint is to use low-cost surrogate models for the limit state function, but in many practical problems (e.g. the source of uncertainty is modeled by a random process), constructing such surrogates itself can be a very challenging task.

3 The sample reweighting method

In this section, we discuss the evaluation of the reliability constraint $c(\mathbf{x})$, or equivalently, the failure probability $P(\mathbf{x})$. Let $\mathbf{z}$ be a d_z-dimensional random variable with distribution $q(\mathbf{z})$, representing the uncertainty in a system. The system reliability is described by the limit state function $g(\mathbf{z})$, and, namely, the event of failure is defined as $g(\mathbf{z}) < 0$. Following the formulations in Rahman (2009), we assume that the distribution of $\mathbf{z}$ depends on the design parameter $\mathbf{x}$, i.e., $q(\mathbf{z}; \mathbf{x})$, while the limit state function $g(\mathbf{z})$ is independent of $\mathbf{x}$. As a result the failure probability is

$$P(\mathbf{x}) = \mathbb{P}(g(\mathbf{z}) < 0) = \int_{\mathbf{z} \in R^{d_z}} I(\mathbf{z}) q(\mathbf{z}, \mathbf{x}) d\mathbf{z}, \tag{3.1}$$

where $I(\mathbf{z})$ is an indicator function:

$$I(\mathbf{z}) = \begin{cases} 1 & \text{if } g(\mathbf{z}) < 0, \\ 0 & \text{if } g(\mathbf{z}) \geq 0. \end{cases} \tag{3.2}$$

$P(\mathbf{x})$ can be computed with the MC estimation:

$$\hat{P}_{\text{MC}} = \frac{1}{N} \sum_{n=1}^{N} I(\mathbf{z}^{(n)}), \tag{3.3}$$

with samples $\mathbf{z}^{(1)}, ..., \mathbf{z}^{(N)}$ drawn from $q(\mathbf{z}; \mathbf{x})$.

Recall that in Algorithm 1, we need to evaluate the failure probability at a number of design points in the TR to construct the surrogate function. Since each evaluation requires a full MC sampling procedure, the total computational cost can be very high. To improve the efficiency, we present a sample reweighting approach, which allows one to obtain the failure probability values at all design points with one full MC based failure probability evaluation. Suppose we have performed a MC estimation of the failure probability at the center of the TR, $\mathbf{x}_c$, obtaining a set of samples from $q(\mathbf{z}; \mathbf{x}_c)$: $\{(\mathbf{z}^{(n)}, g(\mathbf{z}^{(n)}))\}_{n=1}^N$. For any point $\mathbf{x}$ in the TR, we can write $P(\mathbf{x})$ as,

$$P(\mathbf{x}) = \int I(\mathbf{z}) q(\mathbf{z}; \mathbf{x}) d\mathbf{z} = \int I(\mathbf{z}) r(\mathbf{z}) q(\mathbf{z}; \mathbf{x}_c) d\mathbf{z}, \quad (3.4)$$

where $r(\mathbf{z}) = q(\mathbf{z}; \mathbf{x})/q(\mathbf{z}; \mathbf{x}_c)$. It follows immediately that $P(\mathbf{x})$ can be estimated as

$$\hat{P}(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} I(\mathbf{z}^{(n)}) r(\mathbf{z}^{(n)}), \quad (3.5)$$

i.e., by simply assigning new weights $r(\mathbf{z})$ to the samples generated in the evaluation of $P(\mathbf{x}_c)$. Note that, in this method, only the computation of $P(\mathbf{x}_c)$ involves the evaluations of the limit state function $g(\cdot)$, which is referred to as *a full reliability evaluation*. This method uses the same formulation as importance sampling (IS), but it differs from a standard IS as its purpose is not to reduce the sampling variance, but to reuse the samples. We use this approach to construct the surrogates in Algorithm 2.

4 A benchmark example

As an illustrating example, we consider a cantilever beam problem, with width W, height T, length L, and subject to transverse load Y and horizontal load X. This is a well adopted benchmark problem in optimization under uncertainty (Eldred et al. 2002), where the system failure is defined as the maximum deflection exceeding a threshold value:

$$g = D_o - \frac{4L^3}{EWT} \sqrt{\left(\frac{Y}{T^2}\right)^2 + \left(\frac{X}{W^2}\right)^2}. \quad (4.1)$$

Here D_o is the deflection threshold and E is the Young's modulus. In this example we assume the beam length L is fixed to be 100 and $D_0 = 6$. The random variables are: the elastic modulus $E \sim \mathcal{N}(29 \times 10^6, (1.45 \times 10^6)^2)$, external loads $X \sim \mathcal{N}(500, 25^2)$ and $Y \sim \mathcal{N}(500, 25^2)$, and the actual beam width $W \sim \mathcal{N}(w, \sigma^2)$ and height $T \sim \mathcal{N}(t, \sigma^2)$, respectively. The mean width w and the mean height t are design variables, and our goal is to minimize the construction cost $f(w, t) = wt$, subject to that the associated failure probability is smaller than $\theta = 0.1$. In the numerical tests, we solve the problem with $\sigma = 10^{-1}$ and $\sigma = 10^{-2}$.

Table 1 The parameter values of the DF-TR algorithm

ρ_0	$\rho_{\min}$	ϵ^*	ω^-	ω^+	M	δ
0.1	10^{-6}	0.1θ	0.9	1.1	20	10^{-4}

For comparison, we solve the problem with three methods: the DF-TR with reweighting (denoted by DF-TR-R), the DF-TR method without reweighting (denoted by DF-TR), and a standard active set method, where the gradients are computed with the SF method (denoted by SF). The algorithm parameter values of the DF-TR and DF-TR-R algorithms are given in Table 1. In the MC simulations of all the methods, we use two samples sizes $N = 10^4$ and $N = 10^5$. Since all the methods are subject to random errors, to take that into account, we repeatedly solve the problem with all the three methods 100 times and summarize the results in Table 2. Specifically, we compare the average errors of the obtained solutions (compared to a benchmark solution computed by the SF method with 5×10^6 samples), and the average number of full reliability evaluations. We see from the results that in all the test cases, the DF-TR-R algorithm outperforms the SF based method, in terms of both average errors and the number of full reliability evaluations. This suggests that the proposed DF-TR-R algorithm can be more robust and efficient than the SF method for small sample size. In the comparison of the two DF-TR algorithms, we can see that, both algorithms yield comparable results in terms of accuracy, while the DF-TR-R algorithm uses significantly less full reliability evaluations than the algorithm without reweighting. We note that more numerical tests are needed to have a conclusive performance comparison of the

Table 2 Performance comparison of the three methods

σ	N	method	avg error	full evals
10^{-1}	10^4	SF	0.079	142
		DF-TR	0.025	620
		DF-TR-R	0.0273	49
	10^5	SF	0.063	140
		DF-TR	0.021	380
		DF-TR-R	0.0217	28
10^{-2}	10^4	SF	0.0334	114
		DF-TR	0.015	420
		DF-TR-R	0.0201	29
	10^5	SF	0.031	126
		DF-TR	0.011	320
		DF-TR-R	0.0165	18

methods. Nevertheless, the results suggest that the DF-TR-R algorithm provides an efficient and easy-to-use alternative to the SF based methods.

5 Conclusions

In summary, we present a DF-TR algorithm to solve the RBO problems without using the gradients of the reliability constraints. A sample reweighting method is employed so that the TR surrogate can be obtained by performing a single full reliability evaluation. Due to space limitation we only present a simple benchmark example, and applications of the method to some real-world design problems will be reported in a future work. Moreover, we note that in general the design parameters $\mathbf{x}$ could also affect the limit state function itself, and in this case the sample reweighting method does not apply directly. We hope to address this issue in future studies.

References

Aoues Y, Chateauneuf A (2010) Benchmark study of numerical methods for reliability-based design optimization. Struct Multidiscip Optim 41(2):277–294

Augustin F, Marzouk Y (2014) NOWPAC: a provably convergent derivative-free nonlinear optimizer with path-augmented constraints. arXiv:1403.1931

Conn AR, Scheinberg K, Vicente LN (2009) Introduction to derivative-free optimization. SIAM

Eldred MS, Giunta AA, Wojtkiewicz S, Trucano TG (2002) Formulations for surrogate-based optimization under uncertainty. In: 9th AIAA/ISSMO symposium on multidisciplinary analysis and optimization

Rahman S (2009) Stochastic sensitivity analysis by dimensional decomposition and score functions. Probab Eng Mech 24(3):278–287

Rubinstein RY, Shapiro A (1993) Discrete event systems: sensitivity analysis and stochastic optimization by the score function method. Wiley

Valdebenito MA, Schuëller GI (2010) A survey on approaches for reliability-based optimization. Struct Multidiscip Optim 42(5):645–663