

## Research Article

# Minimizing the Discrepancy between Simulated and Historical Failures in Turbine Engines: A Simulation-Based Optimization Method

**Ahmed Kibria, Krystel K. Castillo-Villar, and Harry Millwater**

*Mechanical Engineering Department, The University of Texas at San Antonio, San Antonio, TX 78249, USA*

Correspondence should be addressed to Krystel K. Castillo-Villar; [krystel.castillo@utsa.edu](mailto:krystel.castillo@utsa.edu)

Received 4 July 2014; Revised 30 December 2014; Accepted 31 December 2014

Academic Editor: Chia-Cheng Tsai

Copyright © 2015 Ahmed Kibria et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The reliability modeling of a module in a turbine engine requires knowledge of its failure rate, which can be estimated by identifying statistical distributions describing the percentage of failure per component within the turbine module. The correct definition of the failure statistical behavior per component is highly dependent on the engineer skills and may present significant discrepancies with respect to the historical data. There is no formal methodology to approach this problem and a large number of labor hours are spent trying to reduce the discrepancy by manually adjusting the distribution's parameters. This paper addresses this problem and provides a simulation-based optimization method for the minimization of the discrepancy between the simulated and the historical percentage of failures for turbine engine components. The proposed methodology optimizes the parameter values of the component's failure statistical distributions within the component's likelihood confidence bounds. A complete testing of the proposed method is performed on a turbine engine case study. The method can be considered as a decision-making tool for maintenance, repair, and overhaul companies and will potentially reduce the cost of labor associated to finding the appropriate value of the distribution parameters for each component/failure mode in the model and increase the accuracy in the prediction of the mean time to failures (MTTF).

## 1. Introduction

There are several failure modes in gas turbine engines. Typically, the failure rates of different components are recorded throughout the life of the engine. Upon obtaining the failure rates of such components, the first step is to conduct a statistical distribution fitting to the initial data per component or failure mode. The mean percentage of failures computed while considering the fitted statistical distribution may not be similar to the historical data due to the lack of quantity or quality of the data. Therefore, a common practice is to perform some adjustments in the distribution parameters based on human intelligence and the experience of the engineers. Currently, there is no formal methodology to approach this complex problem.

The reliability modeling of turbine engines is a complex stochastic system. The complexities that arise when randomness is embedded within a system make the analysis of these

systems a difficult task. Unfortunately, epistemic uncertainty is a common and unavoidable characteristic among real-world systems. The development of simulation as means to evaluate stochastic systems enhances the ability to obtain performance measure estimates under several given conditions. Furthermore, these performance measures are much more accurate compared to the estimations of analytical techniques, which often make crude assumptions about the system's condition and operation.

Computer simulations are highly effective in answering evaluative questions concerning a stochastic system. However, it is often necessary to determine the values of the model's decision parameters such that a performance measure is maximized or minimized. This type of problems can be tackled by simulation-based optimization methods [1].

This paper aims to develop a simulation-based optimization method to reduce the discrepancy between simulated and historical failures, which will not only significantly

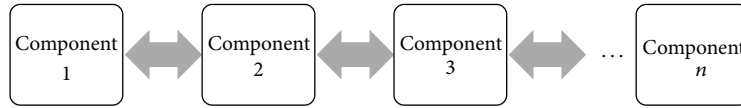


FIGURE 1: Series system for the mock engine model.

reduce the cost of labor hours associated to performing this activity manually but will also find an appropriate value of the distribution parameters for each turbine component/failure mode so that the simulation's outcome better agrees with the real-life data.

This research was done in collaboration with an aerospace industry's maintenance, repair, and overhaul company referred to as ABC Company. A 54-component turbine engine is considered as a case study. For confidentiality reasons, the names of the components, distributions, and distributions' parameters are not mentioned.

The structure of the paper is organized as follows. Section 2 presents the proposed model and its notation. Section 3 discusses the details of the proposed methodology. In Section 4, the case study is presented along with a design of experiments (DOE) to tune the parameters of the optimization algorithm. Section 5 shows the results of the case study. Finally, concluding remarks and future work are presented in Section 6.

## 2. Turbine Engine Model

The engine model is developed using the following assumptions: (1) the component events are independent of each other, (2) the component failure rates interact in a series fashion (see Figure 1), and (3) some of the components may have accumulative time (e.g., an item that has been repaired will recover its previous cumulative time). The model considers the lowest TTF associated to any component as the system's time to failure (i.e., whichever component will have the lowest TTF at each run of the simulation will cause the system to fail)

$$g(T^k) = \min(t_1, t_2, t_3, t_4, \dots, t_n). \quad (1)$$

Equation (1) computes the system's TTF where  $g(T^k)$  is the system's TTF and  $t_i$  is the TTF of  $i$ th component.

## 3. Methodology

The proposed simulation-based optimization method searches for a high quality solution (i.e., the value of the distribution's parameters of the components within their feasible regions) with the use of the simulated annealing (SA) metaheuristic. The proposed method is broken down into two phases.

Phase I consists of a Monte Carlo simulation to obtain the simulated percentage of failure per component, given an initial set of distribution parameters. The simulated percentages of failures from Phase I are used to compute the initial discrepancy between the simulated and the historical

percentages of failures, which is expressed as the sum of squares of errors (SSE). Nevertheless, other combinations of feasible distribution's parameters values may lead to lower SSE values. (i.e., other high quality solutions could exist in nearby neighborhoods). These neighborhoods are explored in the next phase of the method.

In Phase II, a SA-based search procedure is used to explore the neighborhood within the feasible region of the distributions' parameters in order to locate solutions of potentially higher quality. The objective is to minimize the SSE obtained from Phase I. The SA-based procedure works well for Phase II because it has the ability to move away from local minima in contrast with gradient methods that cannot overcome local minima. The SA neighborhood function is defined based on the bounds of the distribution parameters of each component. The following section discusses the implementation in further detail.

*3.1. Monte Carlo Simulation to Obtain the Failure Time of Each Component.* A Monte Carlo simulation is often used to obtain not only the simulated failure times per component, but also the distribution of the statistical estimates of the system's TTFs: MTTF and STTF.

At this stage of the proposed methodology, the distribution and the distributions' parameters values per component (obtained from initial data) are considered as primary inputs. The data set for each component contains suspensions (censored data) which represent the units that have not failed by the failure mode in question. They may have failed by different failure modes or not failed at all [2].

In the case of life data, these data sets are composed of units that did not fail. For example, if five units were tested and only three had failed by the end of the test, it would have *right censored* data (or suspension data) for the two units that did not fail. The term *right censored* implies that the event of interest (i.e., the time to failure) is to the right of our data point. In other words, if the units were to keep on operating, the failure would occur to the right on the time scale [3].

A random number, drawn from a uniform distribution over the interval  $[0, 1]$  is used to generate a failure time from the distribution's inverse cumulative distribution function (CDF). For each component of the model, the formula for the CDF corresponding to the distribution of each component is used and then solved for  $t$  (the failure time). Table 1 displays the equations to generate failure times ( $t$ ).

For the normal or lognormal distributions, there are no closed-form expressions but there are good mathematical approximations for that purpose. The approximation utilized in this paper is the one developed by Weisstein [4], which considers the error function (erfc). The definition for the erfc can be found in the Appendix. The inverse CDF equations,

TABLE 1: Inverse CDF equations for different distributions in the case study.

Distribution	Inverse CDF
Normal	$\left[ -\sqrt{2\sigma} \operatorname{erfcinv} \left( 2 \left( 1 - \left( (1 - F(t)) \left( 1 - \left( 0.5 \operatorname{erfc} \left( -\left( \frac{T - \mu}{\sqrt{2}\sigma} \right) \right) \right) \right) \right) \right) \right) \right] + \mu - T$
Log normal	$e^{[-(\sqrt{2\xi} \operatorname{erfcinv}(2((1-F(t))(1-(0.5 \operatorname{erfc}(-(\ln T - \lambda)/\sqrt{2\xi}))))+1)) + \lambda]} - T$
Weibull (2 parameters)	$\alpha \left( \left( -\ln \left( (1 - F(t)) e^{-T/(\alpha^\beta)} \right) \right)^{1/\beta} \right) - T$
Exponential	$\mu \ln \left[ \frac{1}{1 - F(t)} \right]$

shown in Table 1, include cumulated time ( $T$ ); that is, as different individual components are replaced, the different operating times can be adjusted for the components whose reliability has been restored (through either refurbishment or replacement). Components that are not replaced are returned to service (i.e., their operating times are not different than when they were removed from service). This reliability model will be used to compute the system's conditional probability of failure once it has been repaired. The inverse CDF equations are derived from the conditional reliability equations. If there is no cumulated time (e.g.,  $T = 0$ ), then the inverse CDF equations would be transformed into the unconditional form.

The system's TTF is computed considering the component with the lowest TTF as the system failure rate (1). Thus, a single iteration of the loop computes a single system TTF for the model. Statistical estimates (mean and standard deviation of the system TTFs MTTF and STTF, resp.) as well as the simulated percentages of failures for each component are obtained after performing all the iterations of the Monte Carlo simulation.

**3.2. Simulated Annealing-Based Optimization Procedure.** The simulated annealing (SA) algorithm is a well-known local search metaheuristic used to address discrete, continuous, and multiobjective optimization problems. Its ease of implementation, convergence properties, and ability to escape local optima have made it a popular technique during the past two decades. The application of the SA algorithm to optimization problems emerges from the work of Kirkpatrick et al. [5] and Černý [6]. In these pioneering works, the SA was applied to graph partitioning and very-large-scale integration design. In the 1980s, the SA algorithm had a major impact on the field of heuristic search for its simplicity and efficiency in solving combinatorial optimization problems. Moreover, the SA algorithm has been extended to deal with continuous optimization problems [7–9].

The SA algorithm is based on the principles of statistical mechanics whereby the annealing process requires heating and then a slow cooling of a substance to obtain a strong crystalline structure. The strength of the structure depends on the rate of cooling of the substance. If the initial temperature is not sufficiently high or a fast cooling is applied, imperfections (metastable states) are obtained. In this case, the cooling solid will not attain thermal equilibrium at each temperature. Strong crystals are grown from careful and slow

cooling. The SA algorithm simulates the energy changes in a system subjected to a cooling process until it converges to an equilibrium state (steady frozen state). This scheme was developed in 1953 by Metropolis et al. [10].

The SA-based procedure has been successfully implemented to address problems in manufacturing enterprises such as [11, 12]. The interested reader is referred to the works of Eglese [13], Aarts and Lenstra [14] and Suman and Kumar [15], for a comprehensive description of the simulated annealing metaheuristic.

**3.2.1. Optimization Problem Definition, Objective, and Constraints.** After all the iterations of the Monte Carlo simulation, the percentage of failures is obtained for each component and it is compared to the historical data to obtain the squared error. The discrepancy or the squared error between the simulated and historical percentage of failure is obtained for each component to conform the sum of squared errors (SSE).

The objective function for the optimization process is shown in (2) and aims to minimize the sum of squared errors (SSE) of the system's percentage of failures subject to constraints which are the confidence bounds (contour plots) on the distribution's parameters:

$$\min \left[ \text{SSE} = \sum_{i=1}^n (\hat{x}_i - x_i)^2 \right], \quad (2)$$

where  $\hat{x}_i$  is the simulated percentage of failures,  $x_i$  is the historical percentage of failures, and  $n$  is the number of components.

The simulation-based optimization method will propose new values of the distribution parameters ( $P_{ij}$ ), where  $P_{ij}$  is the  $i$ th distribution parameter of  $j$ th component within their contour plots (see Figure 2). According to Wasserman [16], the likelihood contour plot can be drawn satisfying the following relationship:

$$\ln L(P_{ij}) = \ln L^* - \frac{1}{2} \chi_{1,a}^2, \quad (3)$$

where  $\ln L^*$  is the log-likelihood of the  $j$ th component at the estimated distribution parameters using MLE,  $\ln L(P_{ij})$  is the log-likelihood of the  $j$ th component for the perturbed  $i$ th distribution parameter, and  $a = 1 - \text{confidence level}$ .

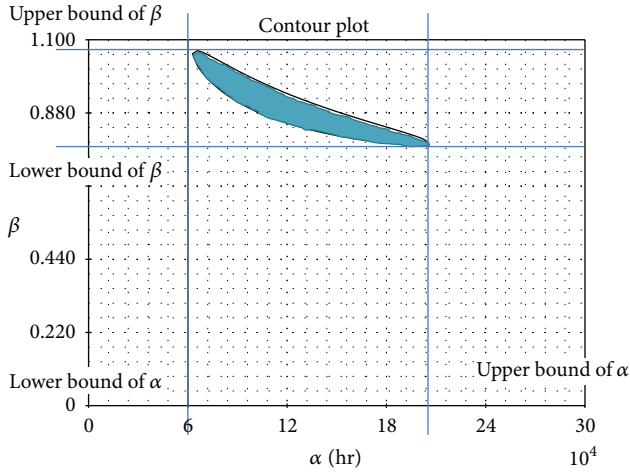


FIGURE 2: Feasible region (contour plot) for the distribution parameters of one component.

In case of unfeasibility, that is, the distribution's parameters at the new state not satisfying (3), one of the distribution parameters is set to its nearest bound (upper or lower bound, whichever is closer) and solved for the other distribution parameter so that they satisfy (3). Equation (4) shows the equation of likelihood contour plot for the Weibull two-parameter distribution as an example:

$$\begin{aligned} \ln L(\alpha (\text{lower bound or upper bound}), \beta) \\ = \ln L^* - \frac{1}{2} \chi_{1,\alpha}^2. \end{aligned} \quad (4)$$

The shaded region in Figure 2 shows an example of the parameters' feasible region for the simulation-based optimization method.

A schematic diagram of the SA-based procedure, used to determine the best value of the distributions' parameters, is presented in Figure 3.

Obtaining the confidence bounds of the distribution's parameters is an integral part of the proposed methodology. There are multiple methods for obtaining the bounds of these parameters. The following section provides the necessary background behind the selection of the likelihood ratio based on the confidence bounds as the preferred method as well as the steps for obtaining the bounds of the distributions' parameters.

**3.2.2. Confidence Bounds on the Distribution Parameters.** Our data set contains suspensions (censored data); therefore, the procedure for computing confidence bounds includes censored data in the parameter estimation. In this section, we present the methodology to obtain the likelihood ratio-based confidence bounds of the distribution's parameters.

In the general case, the distribution parameters are estimated using the maximum likelihood estimation (MLE) method with a modified likelihood function for censored data. Using these estimated parameters, confidence bounds can be calculated. The parameters are estimated using

“mlecustom” function of MATLAB. The general form of the likelihood function considering censored samples is [2]

$$L = \prod_{i=1}^r f(x_i) \prod_{j=1}^k (1 - F(T_j)), \quad (5)$$

where  $r$  is the number of units run to failure,  $k$  is the number of unfailed units,  $x_1, x_2, x_3, \dots, x_r$  are the known failure times, and  $T_1, T_2, T_3, \dots, T_k$  are the operating times on each unfailed unit.

For example, if the time to failure distribution is Weibull-distributed, the modified likelihood function for censored samples would be

$$L = \prod_{i=1}^r \left( \frac{\beta}{\alpha} \right) \left( \frac{x_i}{\alpha} \right)^{\beta-1} e^{-(x_i/\alpha)^\beta} \prod_{j=1}^k e^{-(T_j/\alpha)^\beta}. \quad (6)$$

The “mlecustom” function of MATLAB was used to compute the bounds of the distribution's parameters using the normal approximation method. The normal approximation method for confidence interval estimation is used in most commercial statistical packages because of the relative easiness for the bound's computation. However, the performance of such procedures could be poor when the sample size is not large, or when heavy censoring is considered [17].

On one hand, since the input data set contains heavy censoring, the computation of confidence bounds on parameters using the normal approximation is not recommended. On the other hand, Fisher matrix bounds tend to be more optimistic than the nonparametric rank based bounds. This may be a concern, particularly when dealing with small sample sizes. Fisher matrix bounds are too optimistic when dealing with small sample sizes and, usually, it is preferred to use other techniques for calculating confidence bounds, such as the likelihood ratio bounds [2]. Hence, the likelihood ratio-based confidence bounds estimation method is preferred.

Without loss of generality, the use of the likelihood ratio statistic for developing confidence intervals about a parameter of interest ( $\gamma$ ) can be described. The likelihood ratio (LR) test statistic provides a formal framework for testing the hypothesis:  $H_0: \gamma = \gamma_0$  and  $H_1: \gamma \neq \gamma_0$ .

As its name implies, the LR statistic test is a ratio of likelihood functions. However, it is more convenient to work with the log form, which is computed as the difference between two log-likelihood expressions. Specifically,  $H_0$  is rejected at some level of confidence  $(1 - \alpha)$ , if

$$2(\ln L^* - \ln L^*(\gamma_0)) > \chi_{1,\alpha}^2, \quad (7)$$

where  $\ln L^*$  is the log-likelihood function evaluated at the maximum likelihood (ML) point  $\ln L^*(\hat{\beta}, \hat{\gamma})$  and  $\ln L^*(\gamma_0) = \max_{\beta} \ln L(\gamma_0, \beta)$ . If  $\hat{\beta}(\gamma_0)$  is denoted by the solution to  $\max_{\beta} \ln L(\gamma_0, \beta)$ , then

$$\ln L^*(\gamma_0) = \ln L(\gamma_0, \hat{\beta}(\gamma_0)). \quad (8)$$

An asymptotically correct confidence interval or region on  $\gamma$  consists of all the values of  $\gamma$  for which the null

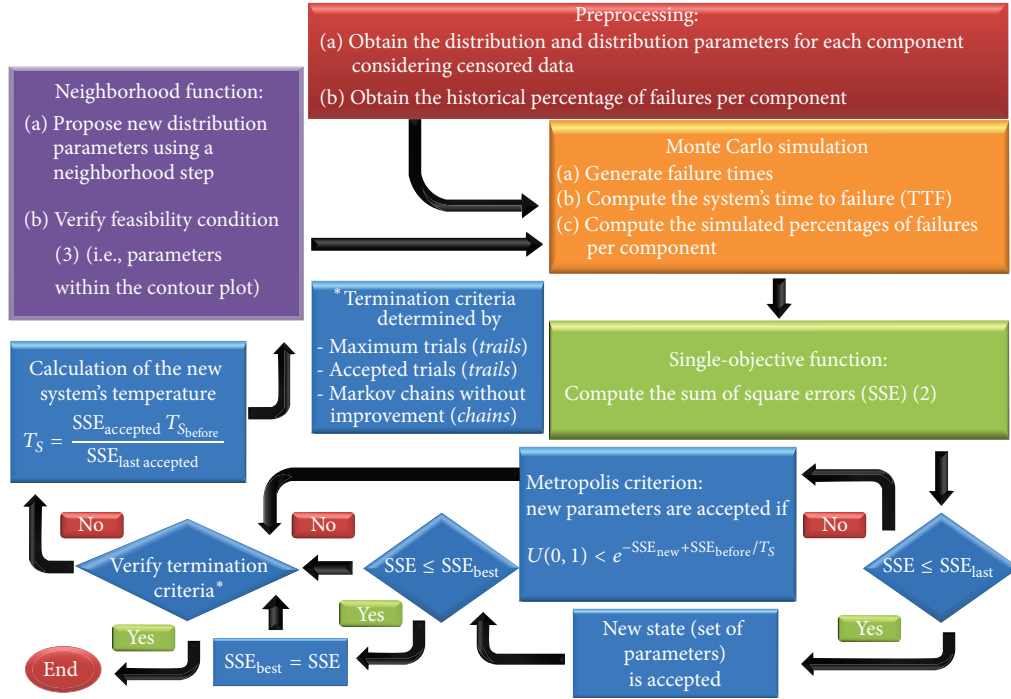


FIGURE 3: Schematic diagram of the SA-based optimization procedure.

hypothesis,  $\gamma = \gamma_0$ , is not rejected at some stated level of significance  $(1 - \alpha)$ . In other words, it consists of all values of  $\gamma_0$  satisfying

$$\ln L^*(\gamma_0) \leq \ln L^* - \frac{1}{2} \chi^2_{1,\alpha}. \quad (9)$$

The LR confidence intervals can be graphically identified with the use of the likelihood contours, which consist of all the values of  $\gamma$  and  $\beta$  for which  $\ln L(\gamma, \beta)$  is a constant. Specifically, the contours constructed are solutions to

$$\ln L(\gamma, \beta) = \ln L^* - \frac{1}{2} \chi^2_{1,\alpha}. \quad (10)$$

Solutions to  $\ln L^*(\gamma_0) = \ln L^* - (1/2) \chi^2_{1,\alpha}$  will lie on these contours. For the Weibull, lognormal, and normal distributions, the confidence intervals can be graphically estimated by drawing lines that are both perpendicular to the coordinate axis of interest and tangent to the likelihood contour [18]. For one-parameter distributions (e.g., exponential), no contour plot is generated and the confidence bound on the parameter will be considered. The summary of steps to obtain the LR based bounds on parameters follows.

*Step 1.* Load the experimental data and define separate sets of information containing the failure and the censored data points from the input data set.

*Step 2.* Define the custom probability density function (PDF) and cumulative density function (CDF) equations based on the distribution of the component.

*Step 3.* Use the “mlecustom()” function in MATLAB to estimate the distribution’s parameters values.

*Step 4.* Compute the output value of the likelihood function (i.e., the modified likelihood function for a component with censored data points) for the parameter estimates obtained in Step 3 using (5).

*Step 5.* Obtain the graphical estimate of confidence intervals of the parameters satisfying (10) from the likelihood contour plot.

*Step 6.* Obtain the chi square statistic value and substitute that value in (11) [19] for the specified confidence level:

$$-2 \ln \left( \frac{L(\theta)}{L(\hat{\theta})} \right) \geq \chi^2_{1-\alpha;k}, \quad (11)$$

where  $L(\theta)$  is the likelihood function for the unknown parameter vector  $\theta$ ,  $L(\hat{\theta})$  is the likelihood function calculated at the estimated vector  $\hat{\theta}$ , and  $\chi^2_{\alpha;k}$  is the chi-squared statistic with probability  $1 - \alpha$  and  $k$  degrees of freedom, where  $k$  is the number of quantities jointly estimated.

*Step 7.* Using the graphical estimates of confidence intervals on the parameters as initial guess, two nonlinear optimization problems are solved to obtain accurate LR limits. Since there are two unknowns (e.g., Weibull’s shape and scale parameters) and only one equation, use an iterative method to obtain the values of the parameters (i.e., for given values of scale, obtain the maximum and minimum value of shape parameter that satisfy (11)) and vice versa. The “fsolve()” function in

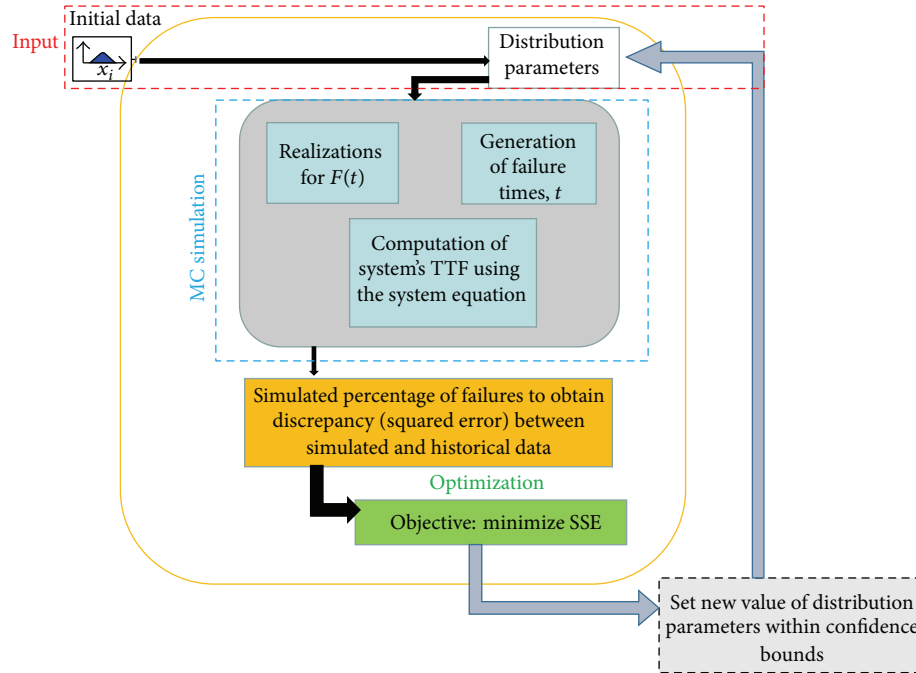


FIGURE 4: Schematic flowchart of the proposed method.

MATLAB was used to compute solution iteratively. The default algorithm for “fsolve()” is the “Trust-Region Dogleg,” which was developed by Powell [20, 21].

**3.3. Summary of Proposed Solution Procedure.** The following steps summarize the proposed methodology.

*Step 1.* The initial data, consisting of failure times and operating times of unfailed units, is provided. The input data is used to determine the distribution behavior of the components’ failure rate and to determine the initial distribution parameters values for each component using Weibull++.

*Step 2.* The components’ distributions and their parameters are used to simulate the expected failure time ( $t$ ). For each component, a random number, that is,  $F(t)$  in the range  $0 \leq F(t) \leq 1$ , is generated using uniform distribution  $U(0, 1)$ . A failure time is generated corresponding to  $F(t)$  from the inverse cumulative distribution function (CDF) equation for each component. Finally, the system’s time to failure (TTF) is computed using the system’s equation (1).

*Step 3.* The simulated percentages of failure are obtained from the Monte Carlo simulation for all the components in the system. The discrepancies (e.g., SSE) between the simulated and historical percentages of failures are computed.

*Step 4.* The simulated annealing algorithm minimizes the sum of square errors (SSE) subject to constraints (e.g., the confidence bounds on distribution parameters).

A schematic diagram of the proposed method is shown in Figure 4.

TABLE 2: Distribution fittings.

Distribution	Parameters	Component number
Weibull (2P)	Shape ( $\beta$ ), scale ( $\alpha$ )	1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 19, 20, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54
		2, 3, 17, 18, 22, 44
Log normal	$\log-\mu$ ( $\lambda$ ), $\log-\sigma$ ( $\zeta$ )	4
Normal	Mean ( $\mu$ ), standard deviation ( $\sigma$ )	21
Exponential	Mean ( $\mu$ )	21

## 4. Case Study

**4.1. Distribution and Parameter Fittings.** The turbine engine model consists of 54 components/failure modes. Table 2 summarizes the parameters corresponding to the distribution of each component used in the analysis. To select the appropriate distribution and parameter values, a weighted goodness of fit measure using Weibull++ software along with engineering experience was utilized. A fixed cumulative time of 1850 hours is used for the 18th component of the model.

**4.2. The SA’s Parameters Tuning through Statistical Design of Experiments (DOE).** A statistical design of experiments can be defined as a series of tests in which purposeful changes are made to the input variables of a process or system so that changes in the output response can be observed and quantified [22].

TABLE 3: Factors and levels during the DOE development.

Factor name	Levels		
	1	2	3
<i>Trials</i>	50	100	200
<i>Chains</i>	1	2	4
<i>Alpha</i>	0.9	0.95	0.98

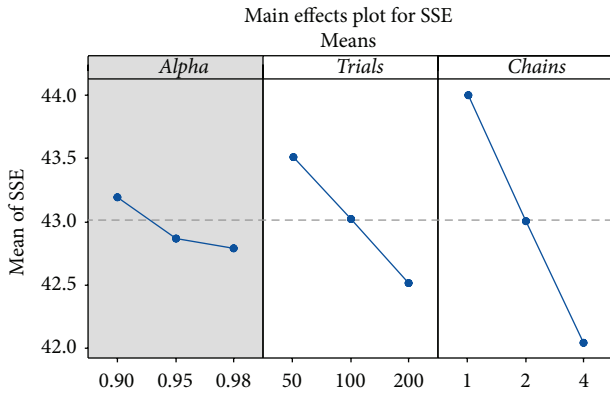


FIGURE 5: Main effects plot for SSE. The gray background represents terms that were not significant in the ANOVA.

In order to tune the parameters of the optimization algorithm, a design of experiments is performed to understand the cause and effect relationship among the SA's parameters and the output response.

The factors that were selected during the DOE development are (1) the number of accepted trials before the system's temperature is decreased (*Trials*), (2) the number of Markov chains (*Chains*), and (3) the rate of cooling (*Alpha*). Table 3 shows the selected factors and their levels.

The responses are (1) the sum of squared errors (SSE), (2) the mean of squared errors (MSE), and (3) the average CPU time.

## 5. Results and Discussion

**5.1. DOE Results.** The DOE analysis was performed using Minitab. A total number of  $3^3 = 27$  computational experiments were performed while considering a 95% significance level. The ANOVA model adequacy verification did not exhibit issues with regard to (1) the residual's normality behavior, (2) the residual's independency, and (3) the residual's constant variance assumptions.

From the ANOVA results, it was observed that the cooling rate (*Alpha*) was not a significant factor and it was fixed to 0.98. The significant factors were number of accepted trials (*Trials*) and the number of Markov chains (*Chains*).

For the sake of brevity, only selected plots are presented. Figure 5 shows the main effects plot considering SSE as a response.

From Figure 5, it can be observed that both factors (e.g., *Trials* and *Chains*) have great impact on the response (i.e.,

TABLE 4: Comparison of the responses (before and after the optimization).

	Sum of squared errors (SSE)	Mean of squared errors (MSE)
Before optimization	50.6541	0.9380
After optimization	35.1834	0.6515

SSE). After a careful inspection of the DOE results, the best combination of SA's parameter values was found to be

- (i) cooling rate (*Alpha*): 0.98,
- (ii) number of accepted trials (*Trials*): 200,
- (iii) number of Markov chains (*Chains*): 4.

The initial systems' temperature was fixed to 10. In order to get this value, the initial temperature is initialized at a very low value and an experimental run is performed where the temperature is raised by a ratio until a minimum acceptance ratio defined by the practitioner (0.8) is met; the acceptance ratio is computed as the number of accepted trials (*Trials*) divided by the trials.

A sensitivity analysis was conducted to determine the best value for the Monte Carlo sample (MC is not an SA algorithmic parameter but it is a parameter in the overall solution procedure). It was observed that the SSE becomes stable for Monte Carlo sample size of 10,000 and larger values; that is, the SSE is not sensible to changes in the Monte Carlo sample size after a value of 10,000 samples. On the other hand, if the sample size for the Monte Carlo loop is below 10,000, the SSE considerably increases for this particular problem. Hence, the selected number of Monte Carlo samples was fixed at 10,000.

**5.2. Test Case Results.** The evaluation of the test case was performed based on the outcomes of the previously presented DOE analysis. After conducting a computational evaluation of the test case, the value of the SSE was found to be 35.1834 having an average CPU time of 55,239 seconds. Table 4 compares the values of the SSE before and after the optimization.

The SA-based procedure was executed five times considering the best combination of algorithmic parameters. Figure 6 shows the average performance of the solution procedure.

Table 5 shows the discrepancies between the simulated and historical percentages of failures for the components before and after the optimization.

Figure 7 shows the histogram of percentage of failures for the historical data, before and after the optimization.

In order to observe the shifting of the distribution's parameters at the different evaluations of the DOE, contour plots were constructed for each component. Figure 8 shows the shifting of the distribution's parameters at the different evaluations of the DOE for the 45th component and Figure 9 shows both the original (fitted) and best found combination of distribution parameters values for 45th component.

TABLE 5: Comparison of the responses (before and after the optimization).

Component number	Historical percentage of failures	Simulated percentage of failure	Before		After		
			Discrepancy [%]	Squared error	Discrepancy [%]	Squared error	
1	0.86	0.5211	-0.3389	0.1149	0.46	-0.4	0.1600
2	1.82	1.5673	-0.2527	0.0639	1.66	-0.16	0.0256
3	3.53	3.6	0.07	0.0049	3.76	0.23	0.0529
4	0.88	0.5947	-0.2853	0.0814	0.63	-0.25	0.0625
5	2.4	2.5271	0.1271	0.0162	2.37	-0.03	0.0009
6	1.31	1.2096	-0.1004	0.0101	1.22	-0.09	0.0081
7	0.76	0.6553	-0.1047	0.0110	0.78	0.02	0.0004
8	0.4	3.5392	3.1392	9.8546	3.04	2.64	6.9696
9	1.04	0.1024	-0.9376	0.8791	0.09	-0.95	0.9025
10	5.35	5.0636	-0.2864	0.0820	5.09	-0.26	0.0676
11	4.56	4.21	-0.35	0.1225	4.26	-0.3	0.0900
12	8.32	7.9168	-0.4032	0.1626	7.87	-0.45	0.2025
13	2.5	2.0153	-0.4847	0.2349	2.38	-0.12	0.0144
14	1.39	1.1423	-0.2477	0.0614	1.18	-0.21	0.0441
15	2.8	2.5067	-0.2933	0.0860	2.45	-0.35	0.1225
16	0.3	0.259	-0.041	0.0017	0.27	-0.03	0.0009
17	0.51	0.499	-0.011	0.0001	0.47	-0.04	0.0016
18	0.91	1.8601	0.9501	0.9027	1.81	0.9	0.8100
19	1.34	0.9799	-0.3601	0.1297	1.24	-0.1	0.0100
20	0.68	0.4956	-0.1844	0.0340	0.68	0	0.0000
21	0.45	0.4621	0.0121	0.0001	0.53	0.08	0.0064
22	0.51	0.4554	-0.0546	0.0030	0.4	-0.11	0.0121
23	0.25	0.1628	-0.0872	0.0076	0.17	-0.08	0.0064
24	11.07	6.3014	-4.7686	22.7395	7.03	-4.04	16.3216
25	4.03	4.1075	0.0775	0.0060	4.26	0.23	0.0529
26	0.55	0.0571	-0.4929	0.2430	0.05	-0.5	0.2500
27	0.7	0.6549	-0.0451	0.0020	0.72	0.02	0.0004
28	5.8	4.4421	-1.3579	1.8439	4.31	-1.49	2.2201
29	0.38	0.2823	-0.0977	0.0095	0.22	-0.16	0.0256
30	3.32	2.1621	-1.1579	1.3407	2.35	-0.97	0.9409
31	0.5	0.4702	-0.0298	0.0009	0.43	-0.07	0.0049
32	0.99	0.7867	-0.2033	0.0413	0.91	-0.08	0.0064
33	0.45	0.3763	-0.0737	0.0054	0.33	-0.12	0.0144
34	0.4	0.3421	-0.0579	0.0034	0.38	-0.02	0.0004
35	0.63	0.822	0.192	0.0369	0.81	0.18	0.0324
36	0.85	1.0452	0.1952	0.0381	0.98	0.13	0.0169
37	0.4	0.561	0.161	0.0259	0.47	0.07	0.0049
38	0.38	0.5465	0.1665	0.0277	0.49	0.11	0.0121
39	0.23	0.3274	0.0974	0.0095	0.36	0.13	0.0169
40	6.02	6.3466	0.3266	0.1067	6.39	0.37	0.1369
41	2.98	4.0667	1.0867	1.1809	3.88	0.9	0.8100
42	0.61	0.9346	0.3246	0.1054	0.93	0.32	0.1024
43	0.6	0.8256	0.2256	0.0509	0.91	0.31	0.0961
44	0.76	0.2627	-0.4973	0.2473	0.27	-0.49	0.2401



TABLE 5: Continued.

Component number	Historical percentage of failures	Simulated percentage of failure	Before		Simulated percentage of failure	After	
			Discrepancy [%]	Squared error		Discrepancy [%]	Squared error
45	6.28	9.0659	2.7859	7.7612	7.84	1.56	2.4336
46	1.23	1.8439	0.6139	0.3769	1.55	0.32	0.1024
47	0.85	1.1438	0.2938	0.0863	1.1	0.25	0.0625
48	1.66	2.1355	0.4755	0.2261	2.37	0.71	0.5041
49	1.29	1.8673	0.5773	0.3333	1.76	0.47	0.2209
50	0.71	0.9715	0.2615	0.0684	1.01	0.3	0.0900
51	2.11	2.9768	0.8668	0.7513	2.92	0.81	0.6561
52	0.38	0.558	0.178	0.0317	0.68	0.3	0.0900
53	0.5	0.7342	0.2342	0.0548	0.82	0.32	0.1024
54	0.45	0.6368	0.1868	0.0349	0.66	0.21	0.0441
			Total	<b>50.6541</b>		Total	<b>35.1834</b>

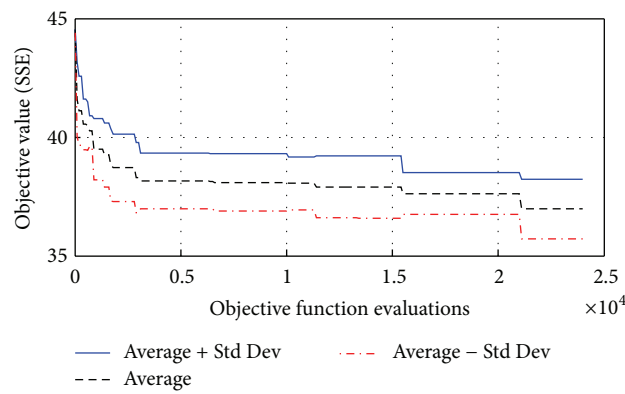


FIGURE 6: The average performance of the SA-based procedure.

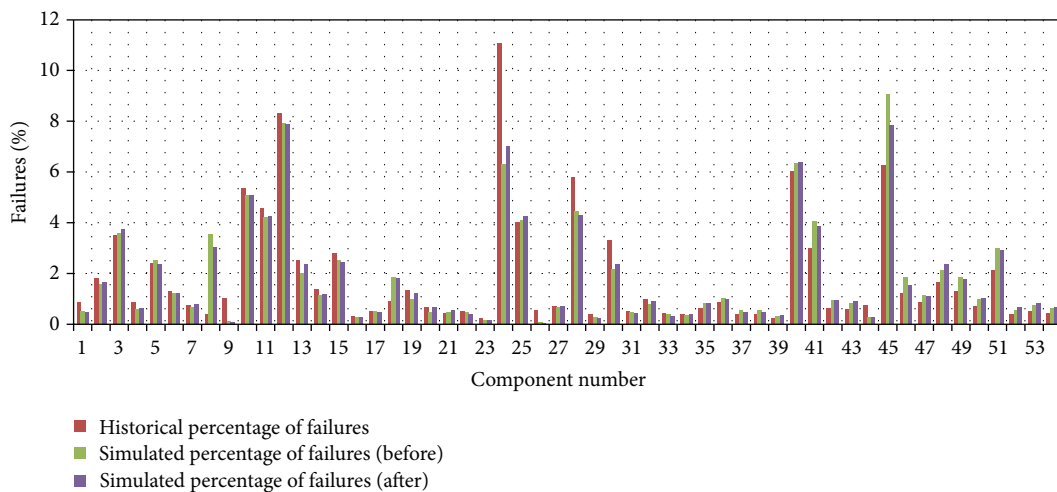


FIGURE 7: Histogram of percentages of failures before and after optimization.

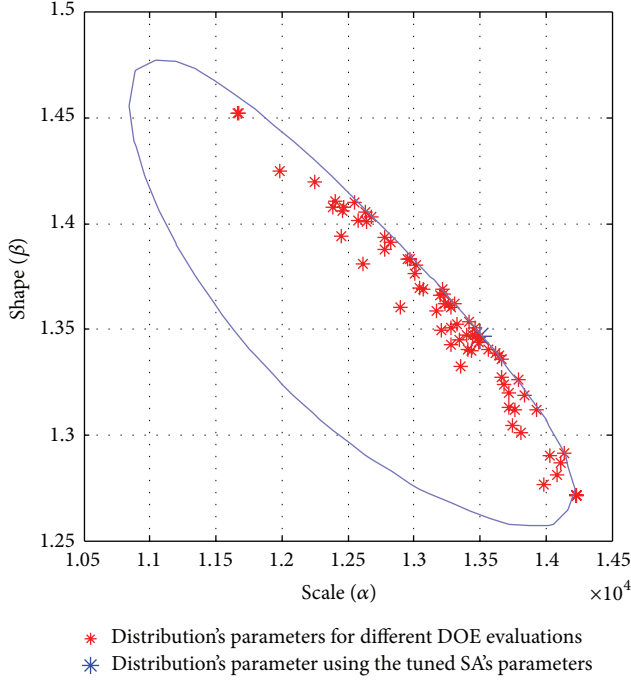


FIGURE 8: Distribution parameters for the 45th component for different evaluation cases.

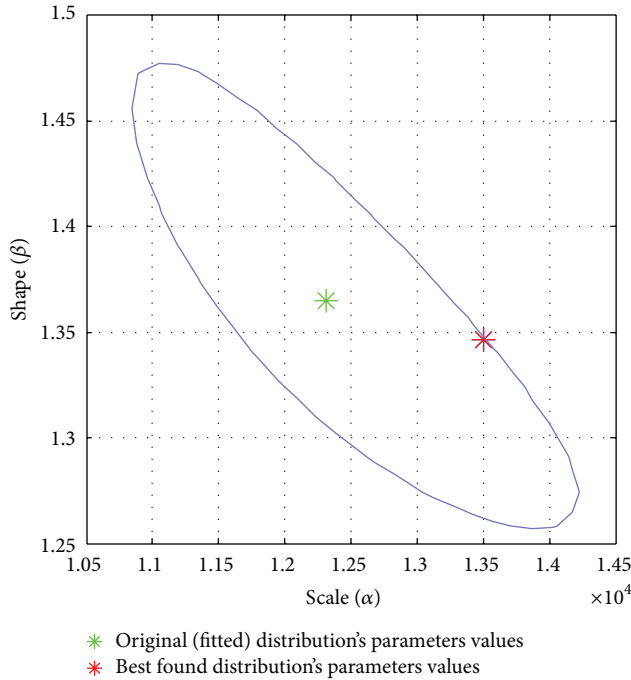


FIGURE 9: Original distribution's parameters values and best found combination of distribution's parameters values for the 45th component.

## 6. Conclusions and Future Work

This paper presented a simulated annealing-based optimization method to minimize the discrepancy of historical and simulated percentages of failures in a turbine engine model. A DOE was performed for the tuning of the algorithmic

parameters. The results showed a 30% reduction of the SSE (e.g., from 50.6541 to 35.1834) for the engine model. The average CPU time was approximately 15 hours mainly due to the calculations involved in the likelihood function. Alternative neighborhood and feasibility functions can be investigated by studying the trends in the shifting parameters' values per component. For instance, it was observed that the distribution's parameters shifted around the edge of the contour plot for some components.

The proposed simulation-based optimization method can serve as a decision-making tool for maintenance, repair, and overhaul companies and will potentially reduce the cost of labor associated to finding the appropriate value of the distribution's parameters for each component/failure mode in the model and increase the accuracy in the prediction of the mean time to failures (MTTF).

Future research lines involve parallelization of the algorithm to solve larger models (e.g., thousands of components) and comparing the performance of the simulated annealing with other metaheuristics such as Evolutionary Algorithms, Tabu Search, and Particle Swarm Optimization, among others.

## Appendix

*Error Function.* The error function  $\text{erf}(X)$  is twice the integral of the Gaussian distribution with 0 mean and variance of 1/2. The equation of error function is

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \quad (\text{A.1})$$

The complementary error function or  $\text{erfc}(X)$  is the complement of the error function; that is,  $\text{erfc}(X) = 1 - \text{erf}(X)$

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt. \quad (\text{A.2})$$

## Nomenclature

- MTTF: Mean time to failure of the system
- STTF: Standard deviation of time to failure of the system
- TTF: Time to failure of the system
- $g(T^K)$ : System TTF
- $T^K$ : Vector of  $K$  variables
- $t$ : Failure time
- $F(t)$ : Cumulative distribution function of failure time ( $t$ )
- $n$ : Number of components in the engine model
- $\alpha$ : Scale parameter of a Weibull distribution
- $\beta$ : Shape parameter of a Weibull distribution
- $\lambda$ : Mean of  $\log(x)$  of a lognormal distribution
- $\zeta$ : Standard deviation of  $\log(x)$  of a lognormal distribution

$\mu$ : Mean of a normal distribution or an exponential distribution  
 $\sigma$ : Standard deviation of a normal distribution.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This research was supported by the Air Force Research Laboratory through General Dynamics Information Technology, TaskOrder F5702-11-04-SC63-01 (PrimeContract no. FA8650-11-D-5702/0004). This support is gratefully acknowledged. The authors wish to acknowledge Carlos Torres, Alan Lesmerises, and Eric Vazquez of StandardAero for their technical assistance (Distribution A. approved for public release, distribution unlimited; Case no. 88ABW-2014-4202).

## References

- [1] S. L. Rosen and C. M. Harmonosky, "An improved simulated annealing simulation optimization method for discrete parameter stochastic systems," *Computers and Operations Research*, vol. 32, no. 2, pp. 343–358, 2005.
- [2] R. B. Abernethy, *The New Weibull Handbook*, Dr. Robert B. Abernethy, 2006.
- [3] Life Data Classification, "ReliaWiki Web Resource," 2012, [http://reliawiki.org/index.php/Life\\_Data\\_Classification](http://reliawiki.org/index.php/Life_Data_Classification).
- [4] E. W. Weisstein, Log Normal Distribution, MathWorld—A Wolfram Web Resource, 2006, <http://mathworld.wolfram.com/LogNormalDistribution.html>.
- [5] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [6] V. Černý, "Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.
- [7] A. Dekkers and E. Aarts, "Global optimization and simulated annealing," *Mathematical Programming, Series B*, vol. 50, no. 1–3, pp. 367–393, 1991.
- [8] M. Locatelli, "Simulated annealing algorithms for continuous global optimization: convergence conditions," *Journal of Optimization Theory and Applications*, vol. 104, no. 1, pp. 121–133, 2000.
- [9] L. Özdamar and M. Demirhan, "Experiments with new stochastic global optimization search techniques," *Computers and Operations Research*, vol. 27, no. 9, pp. 841–865, 2000.
- [10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [11] K. K. Castillo-Villar, N. R. Smith, and J. L. Simontony, "The impact of the cost of quality on serial supply-chain network design," *International Journal of Production Research*, vol. 50, no. 19, pp. 5544–5566, 2012.
- [12] K. K. Castillo-Villar, N. R. Smith, and J. F. Herbert, "Design and optimization of capacitated supply chain networks including quality measures," *Mathematical Problems in Engineering*, vol. 2014, Article ID 218913, 17 pages, 2014.
- [13] R. W. Eglese, "Simulated annealing: a tool for operational research," *European Journal of Operational Research*, vol. 46, no. 3, pp. 271–281, 1990.
- [14] E. E. H. Aarts and J. K. Lenstra, *Local Search in Combinatorial Optimization*, Princeton University Press, Princeton, NJ, USA, 1997.
- [15] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization," *Journal of the Operational Research Society*, vol. 57, no. 10, pp. 1143–1160, 2006.
- [16] G. Wasserman, *Reliability Verification, Testing, and Analysis in Engineering Design*, vol. 153, CRC Press, 2002.
- [17] Y. Hong, L. A. Escobar, and W. Q. Meeker, "Coverage probabilities of simultaneous confidence bands and regions for log-location-scale distributions," *Statistics and Probability Letters*, vol. 80, no. 7–8, pp. 733–738, 2010.
- [18] J. F. Lawless, *Statistical Models and Methods for Lifetime Data*, vol. 362, John Wiley & Sons, 2011.
- [19] M. Kováčová, "Reliability likelihood ratio confidence bounds," *Aplimat—Journal of Applied Mathematics*, vol. 2, no. 2, pp. 217–225, 2009.
- [20] T. J. Dekker, "A floating-point technique for extending the available precision," *Numerische Mathematik*, vol. 18, no. 3, pp. 224–242, 1971.
- [21] M. J. Powell, "A FORTRAN subroutine for solving systems of nonlinear algebraic equations," Tech. Rep. AERE-R-5947, Atomic Energy Research Establishment, Harwell, UK, 1968.
- [22] D. C. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, New York, NY, USA, 2008.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

