INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15-18, 2005

PERFORMANCE OF SURROGATES WITHIN A FRAMEWORK FOR ROBUST OPTIMAL DESIGN

Yang Liu and Tapabrata Ray

Abstract

Robust design optimization problems are known to be computationally expensive as it involves identification of designs that are both good in performance and at the same time have minimal variance in performance among its neighbors. Assessment of performance variance for every design is computationally expensive as numerous designs in the neighborhood needs to be assessed and evaluation of a design could itself involve expensive analysis based on computational fluid dynamics (CFD) or finite element methods (FEM). Approximations can be used in lieu of actual assessments to compute the variance of performance among neighboring designs and their accuracy can be maintained via periodic training. In this paper, we investigate the performance of a radial basis function (RBF) surrogate model and compare its performance with models relying on actual evaluations and a first order Taylor series model. Our optimization model relies on a population based, zero order, stochastic algorithm and it has been implemented using C++ and MPI to make use of multiple processors. Two engineering design optimization examples have been used as case studies to observe the behavior of the surrogate models. Preliminary observations suggest that the Taylor series model is capable of containing the computational cost within affordable limits while maintaining an acceptable level of accuracy. The RBF model did not perform well partly due to its global approximation scheme.

Keywords: Radial Basis Function, Taylor Series, Neighborhood Evaluation.

1. Introduction

In order for a design to be transformed to a real life product, it is necessary to ensure that its performance does not degrade largely under variations in the operating conditions or due to parametric variations that might creep in during the course of manufacturing. Taguchi (1986, 1987) originally proposed the concept of robustness that aims to locate designs that are insensitive to the variations using a loss function. In presence of constraints, the notion of feasibility robustness refers to identifying solutions that are feasible and remain feasible under expected variations. In order to compute the performance of neighboring solutions, various algorithms have been proposed over the years. They can be broadly classified into two categories: (a) gradient based approaches and (b) stochastic methods.

Fundamentally, gradient based approaches are used to compute the sensitivity information of the objective and the constraint functions to arrive at the expected performance values and the performance variances. Models in this category include:

- First order second moment (FOSM) method introduced by Lee and Park (2001).
- Second order second moment (SOSM) method introduced by Luc (2001).

- Reliability based design optimization (RDBO) method proposed by Tu and Choi (1999).
- Advanced first order second moment (AFOSM) method proposed by Jung and Lee (2002).

The gradient based methods have the following limitations in the context of some robust design optimization problems:

- Design problems often have discrete variables or have objective and constraint functions that have functional and slope discontinuity. Gradient based neighborhood assessments cannot be effectively applied to such problems.
- Since gradient based approaches are point improvement methods, they lead to a single solution that limits the flexibility of a designer to choose among multiple designs.

These limitations have led to the development of stochastic methods. Coit and Smith (1996), Tustsui and Ghosh (1997) observed that such methods are quite effective for robust design optimization problems. Jin and Sendhoff (2003) approached to solve the robust design problem as a multiobjective optimization problem while considering performance maximization and performance variance minimization. Ray (2002) introduced a robust design optimization algorithm that did not rely on surrogates for neighborhood assessments and thus was too expensive for real life problems. All these attempts did not pay adequate attention to the cost of computation and did not make use of surrogates and thus is difficult to use for practical problems.

This paper introduces a model for robust optimal design. The underlying optimization algorithm relies on a population based, zero order, stochastic model. The robust design optimization problem is modeled as bi-objective problem i.e. maximize performance and minimize the variance of the performance among its neighbors. The neighborhood performance assessments of a design are based on surrogates while the performance of the design itself is computed using actual analysis. The algorithm employs a novel constraint handling mechanism that takes the feasibility of an individual and its neighborhood feasibility into consideration. The algorithm is implemented using a master-slave topology to make use of multiple processors. The details of the model are presented in Section 2 while the numerical examples are presented in Section 3.

2. Mathematical Formulation

A generic constrained robust design optimization problem can be expressed as follows:

Minimize

$$f = \left[f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \right] \tag{6}$$

where $\mathbf{x} = [x_1 \quad x_2 \quad \ldots \quad x_n]$ is the vector of n design variables, $f_1(\mathbf{x})$ is the objective that needs to be minimized and $f_2(\mathbf{x})$ is the variance of $f_1(\mathbf{x})$ computed using K neighbors randomly created in the hypervolume of the neighborhood of $\mathbf{x}$ defined using $[x_1 + \Delta x_1 \quad x_2 + \Delta x_2 \quad \ldots \quad x_n + \Delta x_n]$ and $[x_1 - \Delta x_1 \quad x_2 - \Delta x_2 \quad \ldots \quad x_n - \Delta x_n]$.

Subject to:

$$g_i(\mathbf{x}) \geq a_i \quad , i = \mathbf{1}, \mathbf{2}, \ldots, q \tag{7}$$

where q is the number of inequality constraints.

For a set of M candidate solutions, the objectives can be represented using a matrix form as follows:

$$OBJECTIVE = \begin{bmatrix} f_{11} & f_{12} \\ \vdots & \vdots \\ f_{M1} & f_{M2} \end{bmatrix} \tag{8}$$

Where f_{11} denotes the value of objective 1 for candidate solution 1. For each candidate solution, the constraint satisfaction vector $\mathbf{c} = [c_1 \quad c_2 \quad \ldots \quad c_q]$ is given by

$$c_i = \begin{Bmatrix} 0 & : \text{if constraint is satisfied} \\ a_i - g_i(\mathbf{x}) & : \text{if constraint is violated} \end{Bmatrix} \tag{9}$$

For the above c_i's, $c_i = 0$ indicates the i^{th} constraint is satisfied, whereas $c_i > 0$ indicates the violation of the constraint. The constraint matrix for a population of M candidate solutions assumes the form

$$CONSTRAINT = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1q} \\ c_{21} & c_{22} & \cdots & c_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M1} & c_{M2} & & c_{Mq} \end{bmatrix} \tag{10}$$

In order to asses the neighborhood feasibility of the i^{th} individual, $c_{i\,q+1} \quad c_{i\,q+2} \quad \cdots \quad c_{i\,2q}$ is computed where $c_{i\,q+1}$ denotes the number of violations of the first constraint among k neighbors where k is an user defined neighborhood size. Thus the modified constraint matrix for the constrained robust optimal design problem assumes the form

$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1q} & c_{1q+1} & c_{1q+2} & \cdots & c_{12q} \\ c_{21} & c_{22} & \cdots & c_{2q} & c_{2q+1} & c_{2q+2} & \cdots & c_{22q} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{M1} & c_{M2} & \cdots & c_{Mq} & c_{Mq+1} & c_{Mq+2} & \cdots & c_{M2q} \end{bmatrix} \tag{11}$$

The pseudocode of the algorithm is presented below:

```
START
Initialize a Population, $gen = 0$ and Specify $\lambda, \gamma$
Evaluate Individuals to compute $f_1(\mathbf{x})$ and constraints $c(x) = [c_1 \quad c_2 \quad \ldots \quad c_q]$
Create a Surrogate Model to Approximate $f_1(x)$ and $c(x) = [c_1 \quad c_2 \quad \ldots \quad c_q]$
While $gen < \lambda$ Do
    $gen = gen + 1$
    Rank Solutions
    Elite Identification and Preservation
    To Fill the remaining members of the Population Do
        Select Partners for Mating via Roulette Wheel based on fitness
        Generate a Child via Recombination
        Call Actual Function Evaluations to Compute $f_1(\mathbf{x})$ and constraints
        $\mathbf{c}(\mathbf{x}) = [c_1 \quad c_2 \quad \ldots \quad c_q]$ for the Child.
        Call Surrogate Model to compute Performance of Neighbors of the Child.
    End
If ( $gen \ mod \ \gamma = 0$ ) Retrain the Surrogate Model
End While
END.
```

Where λ denotes the maximum number of generations allowed for the evolution process and γ denotes the periodic retraining frequency of the surrogate model. The neighborhood sample size and the expected range of variation are user defined inputs.

The pseudo code of the **Elite Identification and Preservation** process is as follows:

```
Compute the non-dominated rank of every solution: $RC_i$ based on the Constraint matrix
and $RO_i$ based on the Objective Matrix.
Set of Elites $E(t) \in Pop(t)$ is formed via the following steps;
    $E(t) \leftarrow \varphi$
    $E(t) \leftarrow I_i$: if $RC_i = 1$
If size of $E(t) > M/2$ and size of $E(t) < M$
    (i) $E(t) \leftarrow \varphi$
    (ii) $E(t) \leftarrow I_i$: if $RO_i \le \frac{1}{M}\sum_{j=1}^{M} RO_j$
Else If size of $E(t) \le M/2$
    (i) $E(t) \leftarrow \varphi$
    (ii) $E(t) \leftarrow I_i$: if $RC_i \le \frac{1}{M}\sum_{j=1}^{M} RC_j$
Else If size of $E(t) = M$ and $\frac{1}{M}\sum_{j=1}^{M} RO_j = 1$
    (i) $E(t) \leftarrow \varphi$
    (ii) $E(t) \leftarrow I_i$: if $CO_i \le \frac{1}{M}\sum_{j=1}^{M} CO_j$ and $CV_i > \frac{1}{M}\sum_{j=1}^{M} CV_j$ or vice versa.
```

where CV_i is the distance of the closest neighbor of the i^{th} individual in the variable space and CO_i is the distance of the closest neighbor of the i^{th} individual in the objective space. These CV_i and CO_i's are tranformed to ranks. When all the solutions of the population turn out to be nondominated, ones that have close neighbors in both the variable and the objective space are dropped from the list of elites. This process is necessary to create the room for solutions that are diverse in both parametric and objective space.

The pseudo code of the **Partner Selection** process is as follows:

- If Number of Feasible Solutions = 0: An indivdual is selected from the Elite List using a Roulette wheel selection based on constraint ranks of the Elites.
- If Number of objectives > 1 and Number of Feasible Solutions > 0: An indivdual is selected from Elite List using a Roulette wheel selection based its the crowding rank

in the objective space (CO_i).

- If Number of objectives = 1 and Number of Feasible Solutions > 0: An indivdual is selected from Elite List using a Roulette wheel selection based on objective ranks of the Elites.

The pseudo code of the **Recombination** operator is as follows:

- Scale every variable between 0-1 using the maximum and minimum value of variables in the population.
- $D = \sum_{j=1}^{N} (I_{P1}^{j} - I_{P2}^{j})^2$; $j = 1,2,\ldots,N$ variables; I_{P1}^{j} denotes the j^{th} variable of the Parent (*P1*) and I_{P2}^{j} denotes the j^{th} variable of the Parent (P2).
- *P* is randomly chosen between *P1* and *P2*. $C(i) = P(i) + N(\mu = 0, \sigma).D$; where σ is the variance of the normal distribution, $i = 1,\ldots,N$ variables. $\sigma=1$ has been used in this study.
- Transform *C(i)'s* back to original scale to get the new child.

The algorithm presented above, attempts to improve the performance of all individuals of a population unlike some evolutionary models where only the good parents participate in mating. This behavior on one hand leads to expensive evaluations, while on the other provides scope for a wider exploration that is useful for problems that are highly nonlinear. The greedy element of the algorithm arises from the stringent elite selection procedure and the crossover operator that explores the neighborhood of the elites. The algorithm is particularly attractive as a design tool, since it does not use scaling and aggregation of constraints and objectives, and does not require additional inputs unlike most of its counterparts. Although the use of nondominated sorting to handle constraints is an expensive operation, it is meaningful for problems where the objective and the constraint functions are equally or even more expensive. Although this is the basic structure of the algorithm, an experienced user can easily modify the selection criteria, fitness assignment mechanism or use different recombination operator.

2.1 Surrogate Models

Actual Evaluation Model: In this scheme, all neighborhood assessments are obtained using actual function evaluations. This is used to compare the accuracy of the RBF and the Taylor series based models.

Taylor Series Model: In this scheme, forward differencing using actual function evaluations is used to compute $\frac{df}{dx_i}$ and subsequently performance of neighbors are assessed using the following:

$$f(x_{Neighbor}) = f(x) + \sum_{i=1}^{N} \frac{df}{dx_i}(x_{Neighbor,i} - x_i) \qquad (12)$$

The number of additional function evaluations is based on the number of variables of the problem.

Radial Basis Function (RBF) Model: In this scheme, neighborhood assessments for objective and constraint functions are based on a prediction using a RBF model. The RBF model is created using M/2 data sets identified by k-means clustering. (M is the population size). The RBF model is retrained every 5 generations and a Gaussian model is adopted where the spread is computed using 2 closest neighbors in the variable space.

3 Design Examples

3.1 Welded Beam Design

The first example deals with a well-studied welded beam design problem. The aim is to minimize the cost of the beam subject to constraints on shear stress, bending stress, buckling load, and the end deflection. The four continuous design variables are thickness of the beam x_1, width of the beam x_2, length of the weld x_3, and the weld thickness x_4.. We have used a population size of 40 to solve this problem and a neighborhood size of 40. The σ of the Gaussian functions were obtained based on average distance of 2 closest neighbors in the variable space and a uniform random sampling scheme has been used to generate 40 neighbors within a range of range of +/- +/-1.0% of the variable range around the design point.

Minimize

$$f_1(\mathbf{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2) \quad (13)$$

$$f_2(\mathbf{x}) = Variance\ of\ f_1 \quad (14)$$

Subject to

$$\tau(\mathbf{x}) - \tau_{max} \leq 0 \quad (15)$$

$$\sigma(\mathbf{x}) - \sigma_{max} \leq 0 \quad (16)$$

$$x_1 - x_4 \leq 0 \quad (17)$$

$$\delta(\mathbf{x}) - \delta_{max} \leq 0 \quad (18)$$

$$P - P_C(x) \leq 0 \quad (19)$$

The other parameters are defined as follows:

$$\tau(\mathbf{x}) = \sqrt{(\tau')^2 + \frac{2\tau'\tau'' x_2}{2R} + (\tau'')^2} \qquad \tau' = \frac{P}{\sqrt{2} x_1 x_2}$$

$$\tau'' = \frac{MR}{J} \qquad M = P(L + \frac{x_2}{2})$$

$$R = \sqrt{\frac{x_2{}^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \qquad J = 2\left\{\frac{x_1 x_2}{\sqrt{2}}\left[\frac{x_2{}^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(\mathbf{x}) = \frac{6PL}{x_4 x_3{}^2} \qquad \delta(\mathbf{x}) = \frac{4PL^3}{Ex_4 x_3{}^3}$$

$$P_C(\mathbf{x}) = \frac{4.013\sqrt{\frac{EGx_3{}^2 x_4{}^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

where, P = 6000 lb., L =14, $\delta_{\max}$ =0.25 in, E =30 × 10^6 psi, G = 12 × 10^6 psi, $\tau =_{max}$ 13,600 psi, σ_{max} = 30,000 psi, $0.125 \le x_1 \le 10.0$, $0.1 \le x_2 \le 10.0$, $0.1 \le x_3 \le 10$ and $0.1 \le x_4 \le 10.0$.

Table 1 Performance Comparison of the Surrogates

	Scheme 1	Scheme 2	Scheme 3
Total Number of Actual Function Evaluations	490,769	59,885	12,021
Number of RBF Approximations	---	---	479,240
Total Number of Solutions Evaluated	12,009	12,009	12,021

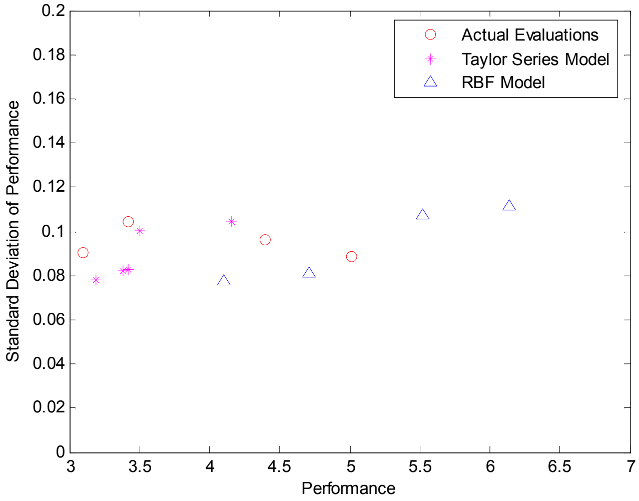

Figure 1. Comparison of performance among the surrogate models (Standard deviation is based on the surrogate model)

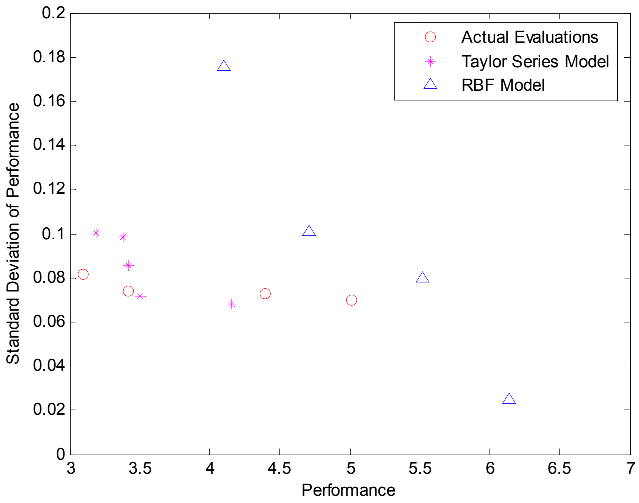

Figure 2. Comparison of performance among the surrogate models (Standard deviation is based on actual evaluations for the designs in Figure 1)

Table 2. Comparison of results obtained using various surrogate models

	X$_1$	*X*$_2$	*X*$_3$	*X*$_4$
Actual	0.3324	5.7456	7.3185	0.6199
	0.3126	5.9998	7.3665	0.3449
	0.3121	5.9908	7.2088	0.4001
	0.347	5.6459	7.2228	0.534
Taylors Series Model	0.2741	6.316	7.9547	0.3674
	0.308	6.4329	7.4903	0.3419
	0.3562	5.4484	6.6113	0.4429
	0.3517	5.1347	7.4686	0.5019
	0.3043	5.7871	7.7526	0.3827
RBF Model	0.3214	5.7324	6.5233	0.5562
	0.5829	6.0147	5.5259	0.7294
	0.6401	3.0474	6.0933	0.6673
	0.5142	6.1407	5.6715	0.679

The results indicate that the Taylor series based model is able to reduce the computational cost to about 1/8th while maintaining reasonable accuracy. The RBF model attempting to solve the same with 1/5th of the computational cost of the Taylor series model did not fair well.

3.2 Three Bar Truss Design

A three-bar truss design problem is considered next. In this problem, the volume is minimized subject to stress constraints. We have used a population size of 20 and a neighborhood size of 20 to solve this problem.

Minimize

$$f(\mathbf{x}) = (2\sqrt{2}x_1 + x_2) \times l \tag{20}$$

$$f_2(\mathbf{x}) = Variance\ of\ f_1 \tag{21}$$

Subject to

$$\frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^{\ 2} + 2x_1x_2}P - \sigma \leq 0 \tag{22}$$

$$\frac{x_2}{\sqrt{2}x_1^{\ 2} + 2x_1x_2}P - \sigma \leq 0 \tag{23}$$

$$\frac{1}{x_1 + \sqrt{2}x_2}P - \sigma \leq 0 \tag{24}$$

where, $0 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 1$; l=100 cm, P=2KN/cm^2, and σ = 2KN/cm^2

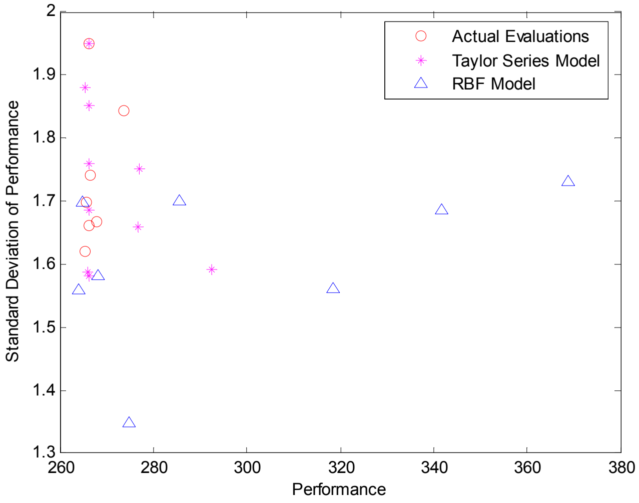

Figure 3. Comparison of performance among the surrogate models (Standard deviation is based on the surrogate model)

Table 3 Performance Comparison of the Surrogates

	Scheme 1	Scheme 2	Scheme 3
Total Number of Actual Function Evaluations	377,684	53,969	18,003
Number of RBF Approximations	---	---	359,660
Total Number of Solutions Evaluated	18,004	18,003	18,003

Table 4. Comparison of results obtained using various surrogate models

	X_1	X_2
Actual	0.785391	0.514556
	0.808259	0.37474
	0.789561	0.428947
	0.805721	0.385574
	0.8112	0.384765
	0.782263	0.444663
	0.798286	0.395592
Taylors Series Model	0.757555	0.781128
	0.802398	0.497857
	0.784544	0.444198
	0.791064	0.53345
	0.800208	0.396092
	0.78419	0.444365
	0.78361	0.4447
	0.783711	0.445021
	0.784083	0.445029
	0.784341	0.436
RBF Model	0.966621	0.953237
	0.839989	0.480761
	0.812598	0.350402
	0.964367	0.688285
	0.827841	0.339764
	0.873021	0.279326
	0.779782	0.434368
	0.809088	0.896365

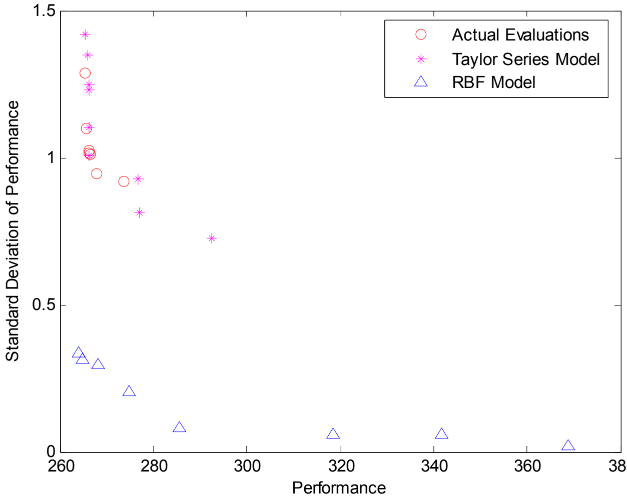

Figure 4. Comparison of performance among the surrogate models (Standard deviation is based on actual evaluations for the designs in Figure 1)

It is clearly visible that the Taylor series model did perform fairly well and reduced the computational cost to about 1/7th as compared to the model relying on actual function evaluations. The performance of the RBF based model using 1/3rd the number of actual evaluations used by the Taylor series model is not good.

4. Summary and Conclusions

In this paper, we observed the behavior of two surrogate models namely the Taylor series model and the RBF based model within the framework of robust optimal design. The robust optimal design problem was formulated and solved using a multiobjective approach where the first objective is to maximize performance and the other being to minimize performance variance. The neighborhood of a solution was defined using a range and a random sampling was used to assess performance variance of the solutions.

Our observations indicate that a simple Taylor series based first order model is able to reduce the computational cost substantially while maintaining suitable accuracy for the problems studied. The RBF model although quite successful in representing nonlinear functions and in our surrogate assisted design optimization framework (SADO), did not perform well for the robust design optimization model. The Taylor series model was quite successful as it uses neighborhood assessments unlike the RBF model relying on a single global model for neighborhood assessments of all solutions. We are currently investigating the performance of the Taylor series based model for robust design of airfoils. Other aspects that we are looking into include various neighborhood sampling schemes and use of surrogates that rely on both a local and a global approximation model.

References

1. Coit, D.W and Smith, A.E. (1996). Reliability Optimization of Series-Parallel Systems Using a Genetic Algorithm, *IEEE Transactions on Reliability,* Vol. 45 No. 2.
2. Jung, D.H and Lee, B.C (2002). Development of a simple and efficient method for robust optimization, *International Journal for Numerical Methods in Engineering*, 53:2201-2215.
3. Luc, H. (2001). Solving Problems of Optimization under Uncertainty as Statistical Decision Problems, *AIAA-2001-1519*.
4. Lee, K.H. and Park, G.J., (2001). Robust Optimization Considering Tolerances of Design Variables, *Computers & Structures*, Vol. 79, pp. 77-86.
5. Ray, T. (2002). Constrained Robust Optimal Design using a Multiobjective Evolutionary Algorithm, Proceedings of *IEEE World Congress on Computational Intelligence,* Congress on Evolutionary Computing (CEC 2002), Hawaii, May 12-17, 2002, pp. 419-424.
6. Tustsui, S. and Ghosh, A. (1997). Genetic Algorithms with a Robust Solution Searching Scheme, *IEEE Transactions on Evolutionary Computation*, 1(3):201-208.
7. Taguchi, G. (1986). Introduction to Quality Engineering. White Plains, NY: Krauss International Publications.
8. Taguchi, G. (1987). System of Experimental Design, Vols. 1 and 2, White Plains, NY: Krauss International Publications.

9. Tu, J. and Choi, K.K. (1999). A New Study on Reliability-Based Design Optimization, *ASME J. of Mech. Des.*, Vol. 121, No. 4, pp. 557-564.

10. Yaochu Jin and Bernhard Sendhoff, (2003). Trade-off between Performance and Robustness: An Evolutionary Multi-objective Approach, *Proceedings of the Second International Conference on Evolutionary Multi-criterion Optimization*, LNCS 2632, Springer, pp. 237-251.