JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 5, August 2008

Development of Multi-Objective Six-Sigma Approach for Robust Design Optimization

Koji Shimoyama*
University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8654, Japan

Akira Oyama† and Kozo Fujii‡
Institute of Space and Astronautical Science, Sagamihara, Kanagawa, 229-8510, Japan

DOI: 10.2514/1.30310

In this study, a new optimization approach for robust design, design for multi-objective six sigma, has been developed and applied to three robust optimization problems. The design for multi-objective six sigma builds on the ideas of design for six sigma, coupled with multi-objective evolutionary algorithm, for an enhanced capability to reveal tradeoff information considering both optimality and robustness of design. While design for six sigma requires careful input parameter setting, design for multi-objective six sigma needs no such prior tuning, plus it can reveal the tradeoff information in a single optimization run. Three robust optimization problems were taken as to demonstrate the capabilities of design for multi-objective six sigma. Results indicate that design for multi-objective six sigma has a more practical and more efficient capability than the design for six sigma to reveal tradeoff design information considering both optimality and robustness of design.

Nomenclature

C	airfoil chord length
d	airfoil drag
$F(\boldsymbol{x})$	fitness value
$\boldsymbol{f}(\boldsymbol{x})$	objective function vector
G	number of iterations (equivalent to number of generations in evolutionary algorithm)
LSL	lower specification limit
l	airfoil lift
M	number of objective functions
M_∞	freestream Mach number
m	number of dispersive design variables
N	number of solutions evaluated in each iteration (equivalent to population size in evolutionary algorithm)
n	sigma level
S	number of sample points to measure μ_f and σ_f of each solution

Received 7 February 2007; revision received 8 June 2008; accepted for publication 9 June 2008.

* Research Fellow, Institute of Fluid Science, 2-1-1 Katahira, Aoba-ku, currently, Tohoku University, Sendai, 980-8577, shimoyama@edge.ifs.tohoku.ac.jp. AIAA Member.
† Research Associate, Department of Space Transportation Engineering, 3-1-1 Yoshinodai, oyama@flab.eng.isas.jaxa.jp. AIAA Member.
‡ Professor, Department of Space Transportation Engineering, 3-1-1 Yoshinodai, fujii@flab.eng.isas.jaxa.jp. AIAA Fellow.

USL	upper specification limit
w_μ	weighting factor of μ_f
w_σ	weighting factor of ${\sigma_f}^2$
X	airfoil chordwise coordinate
$\boldsymbol{x}$	design variable vector
Y	airfoil vertical coordinate

Subscripts

i	identification number for each solution
j	identification number for each design variable
k	identification number for each objective function

Superscripts

p	identification number for each sample point

Symbols

μ	mean value (of a dispersive variable or function)
μ_f	mean value of $f(\boldsymbol{x})$
$\boldsymbol{\mu_x}$	mean value vector of $\boldsymbol{x}$
σ	standard deviation (of a dispersive variable or function)
σ_f	standard deviation of $f(\boldsymbol{x})$
$\boldsymbol{\sigma_x}$	standard deviation vector of $\boldsymbol{x}$

I. Introduction

DESIGN optimization can be described as the problem of determining the inputs of an objective function that will maximize or minimize its value at a certain design point. For instance, an airplane wing may be optimized as to determine the ideal wing shape for a more efficient aerodynamic performance, e.g., fly faster, more stably, or more economically, at an assumed design point. Although this conventional approach that considers only the optimality of design (i.e., performance at design point) should work fine in a controlled environment, real-world applications inevitably involve errors and uncertainties (be it in the design process, manufacturing process, and/or operating conditions), so that the resulting performance may be lower than expected. Actually, the airplane wing design is such a case where the aerodynamic performance is very sensitive to the wing shape and flight conditions, and thus it may deteriorate drastically when subject to wing manufacturing errors and wind variations.

More recently, an approach that considers not only optimality but also robustness (i.e., performance sensitivity against design errors and uncertainties) has attracted considerable attention in the search for more practical designs. A brief comparison between conventional optimization and robust optimization is illustrated in Fig. 1. Solution A obtained by a conventional optimization is the best in terms of optimality, but disperses widely in terms of objective function against the dispersion of design variable or environmental variable, and this dispersion may extend to an unacceptable (unsafe) range. On the other hand, solution B obtained by a robust optimization is moderately good in terms of optimality and also good in terms of robustness, i.e., dispersion of objective function is narrow against dispersion of design variable.

Indeed, improvements in optimality and robustness are usually competing in real-world design problems. Therefore, not a single but multiple robust optimal solutions actually exist. Objectives of robust design optimization are a) to find compromised solutions between optimality and robustness, and b) to reveal the tradeoff information between them. With an additional design consideration such knowledge can aid the upper-level decision maker to pick one solution from the compromised solutions.

To date, several robust optimization approaches have been proposed by many researchers. In the approaches proposed by Gunawan and Azarm [1] and Li et al. [2], acceptable ranges are prespecified for each objective and/or constraint function dispersions. Both approaches consist of inner and outer optimization problems. In the inner subproblem, the maximum size (radius) of the dispersive region in the design variable space, so that the objective

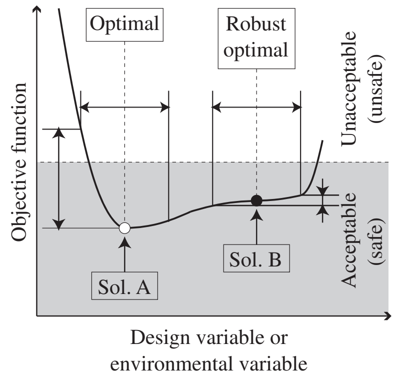


Fig. 1 Comparison between conventional optimization and robust optimization (for objective function to be minimized).

and/or constraint function dispersions are safely included within the prespecified ranges, is evaluated as a robustness measure. In the outer main problem, the approach of Gunawan and Azarm considers the radius as an additional constraint, while that of Li et al. considers the radius as an additional objective function. Clearly, both approaches result in searching only robust optimal solutions the objective and/or constraint function dispersions of which fit their acceptable ranges. This means that these approaches require additional optimization runs with prespecification of different acceptable ranges to search other robust optimal solutions with more or less robust characteristics, i.e., to extract information on the tradeoff between optimality and robustness. In addition, the resulting robust optimal solutions are strongly dependent on the prespecified ranges. In cases in which the prespecified ranges are not appropriate (e.g., they are too severe) or some information about the ranges is unavailable, these approaches may fail in searching for robust optimal solutions.

On the other hand, the approach proposed by Deb and Gupta [3] pre-specifies ranges for dispersion of all design variables, and sets an upper limit for the objective function dispersion as a robustness measure in the form of a constraint. This approach can search not only robust optimal solutions the objective function dispersion of which fits the upper limit, but also those for which the objective function dispersion becomes smaller (i.e., more robust) than the upper limit. Thus, this approach can search for a wider range of robust optimal solutions as compared to the above approaches [1,2]. However, this approach still has difficulty in pre-specifying the upper limit appropriately.

The approach proposed by Ong et al. [4] considers the worst value of a dispersive objective function value (for prespecified design variable dispersions) as a single objective function. By optimizing the worst objective function value, this approach works well in terms of finding an extremely robust optimal solution, but it becomes difficult to search for a global tradeoff between optimality and robustness because this approach is based on a single-objective formulation.

Aiming at a practical method to obtain the global tradeoff relation between optimality and robustness of design, the approaches of Ray [5] and Jin and Sendhoff [6] use multi-objective evolutionary algorithms (MOEAs) [7]. Both approaches deal with two statistical values (mean value and standard deviation) of a dispersive objective function, and treat them as multiple separate objective functions corresponding to optimality and robustness measures, respectively. Therefore, these approaches can find multiple robust optimal solutions which correspond to global tradeoff between optimality and robustness. Then, as mentioned above, designers can pick one from the multiple solutions according to their preferences. However, it is difficult to perform this selection based upon mean and standard deviation values themselves, as they do not represent the robustness quality (e.g. safety probability) explicitly.

Here, we focus on the approach called design for six sigma (DFSS) [8], which is a popular robust optimization approach in various fields of engineering. The DFSS treats the weighted summation of mean value and variance of a dispersive objective function as a single objective function, and considers the constraint on “sigma level.” Generally speaking, the sigma level corresponds to the safety probability that the dispersive objective function does not exceed its lower and upper limits. The sigma level represents a robust design quality explicitly and can be used to pick one solution from multiple robust optimal solutions obtained after the optimization. However, the DFSS requires the specification of sigma level in advance, and as such, it also leads to the same difficulty of careful input parameter setting. In addition, the DFSS has difficulties in setting weighting factors and revealing tradeoff information between optimality and robustness because it is based on a single-objective optimization formulation as mentioned above.

The present paper has two main objectives: (1) to propose a new robust design optimization approach "design for multi-objective six sigma (DFMOSS)," that combines the ideas of DFSS and MOEA, to overcome some difficulties in the DFSS as mentioned above, and (2) to demonstrate the capabilities of the new DFMOSS approach by considering three robust optimization problems (test function problem, welded beam design problem, and airfoil design problem for a Mars airplane) as present applications for the DFMOSS.

The paper is organized as follows: Sec. II gives a brief description of the DFSS approach and formulation, while Sec. III explains the proposed approach DFMOSS. Section IV presents the results for the three robust optimization problems, and discusses the capabilities of DFMOSS and DFSS comparatively. Finally, Sec. V concludes the present paper.

II. Design for Six Sigma

DFSS [8] is based on the "six-sigma concept," which was originally established as a measure of excellence for business processes. The aim is to achieve a process with such a small dispersion that the range of size 12σ $(-6\sigma \sim 6\sigma)$ around the mean value μ is included in an acceptable (safe) range for the performance parameter. The level of dispersion can be defined as "sigma level n," as shown in Fig. 2 where larger sigma level indicates smaller dispersion. In the context of robust design optimization, smaller dispersion translates to a more robust characteristic.

Consider a single-objective nonconstrained optimization problem where the value of objective function $f(\boldsymbol{x})$ of design variable $\boldsymbol{x}$ must be minimized as follows

$$\text{Minimize: } f(\boldsymbol{x}) \tag{1}$$

In robust design optimization using DFSS, Eq. (1) is rewritten to the problem where the weighted summation of the mean value μ_f and the variance ${\sigma_f}^2$ of the objective function $f(\boldsymbol{x})$ must be minimized as follows:

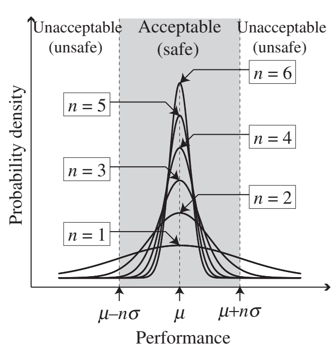
$$\text{Minimize: } w_\mu \mu_f + w_\sigma {\sigma_f}^2 \tag{2}$$

μ_f and σ_f are estimated by sampling $\boldsymbol{x}$ following its probability distribution and evaluating $f(\boldsymbol{x})$ at each sample point. In DFSS, the following inequality constraints are specified in advance to achieve the expected sigma level quality of the obtained solution, as shown in Fig. 2.

$$\text{Subject to: } \mu_f - n\sigma_f \geqslant \text{LSL} \tag{3a}$$

$$\mu_f + n\sigma_f \leqslant \text{USL} \tag{3b}$$

Figure 3 illustrates the flowchart of robust optimization using DFSS. First, parameters such as weighting factors w_μ and w_σ, sigma level n, and LSL/USL (USL and LSL are called upper and lower specification limits, which correspond to the upper and lower bounds of an acceptable range, respectively) are pre-specified by the user, and



Sigma level n	Acceptable (safety) probability, %
1	68.25
2	95.46
3	99.73
4	99.9937
5	99.999943
6	99.9999998

Fig. 2 Characteristics of sigma level *n*.

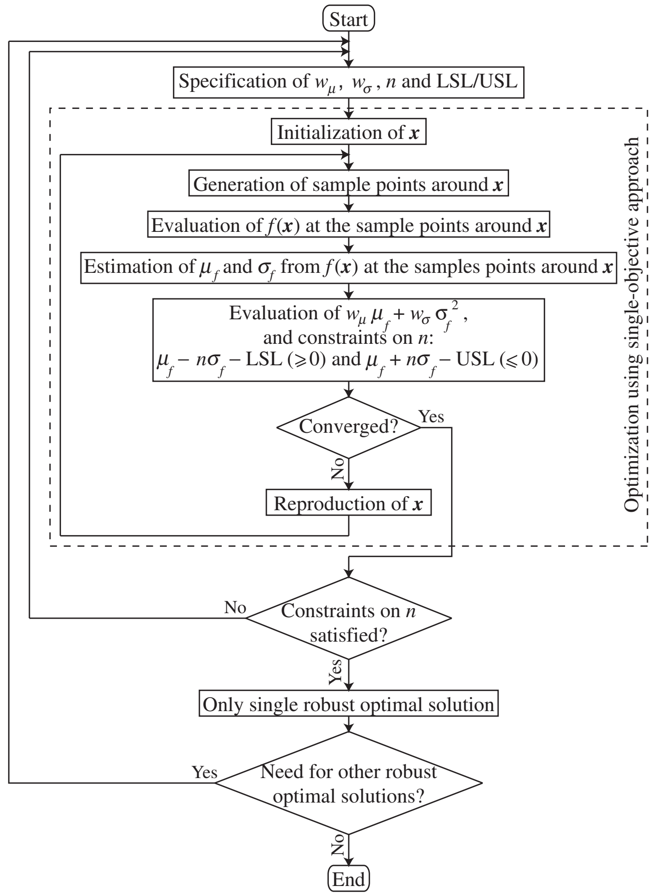


Fig. 3 Flowchart of robust optimization using DFSS.

then it proceeds to the optimization block. In this block, sample points $\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \ldots, \boldsymbol{x}_i^S$ are generated around each solution $\boldsymbol{x}_i$ $(i = 1, 2, \ldots, N)$, and corresponding $f(\boldsymbol{x}_i^1), f(\boldsymbol{x}_i^2), \ldots, f(\boldsymbol{x}_i^S)$ are evaluated. From the sampled $f(\boldsymbol{x}_i^p)$ $(p = 1, 2, \ldots, S)$, μ_f and σ_f are estimated for each solution x_i, and then $w_\mu \mu_f + w_\sigma {\sigma_f}^2$ is dealt with as one objective function. Then, $\mu_f - n\sigma_f - \mathrm{LSL}(\geqslant 0)$ and/or $\mu_f + n\sigma_f - \mathrm{USL}(\leqslant 0)$ are evaluated as two constraint functions. For the next step, the solutions are reproduced based on the evaluated objective and constraint functions, and this optimization process is iterated until maximum number of iterations is reached or an optimal solution does not change any more. The total number of function evaluations is $S \times N \times G$ for a single DFSS optimization run. This single-objective optimization can be carried out by using any single-objective optimization approach (optimization approach used in here is given in Sec. IV).

The DFSS has some limitations as follows. First, it is necessary to prespecify the weighting factors w_μ and w_σ carefully, even though it is difficult for the user to prespecify values for weighting factors appropriately because the tradeoff information is still unknown. Also, it is necessary to prespecify an appropriate sigma level n, although essentially, the sigma level satisfying Eqs. (3a) and (3b) is known only after an optimization run. Therefore, users must prespecify the sigma level without any information, while a robust optimal solution may not even be obtainable at the prespecified value of sigma level. Second, only one robust optimal solution can be obtained in one optimization run, for the DFSS operates according to the single-objective function given by Eq. (2). As a result, many optimization runs with different values of weighting factors and sigma level must be performed by the user to obtain multiple robust optimal solutions, which shall then reveal the tradeoff relation between optimality and robustness of performance. Furthermore, the tradeoff relation between optimality and robustness may not be derived even after many optimization runs (e.g. if the obtained multiple optimal solutions distribute only locally).

III. Design for Multi-objective Six Sigma

Before proposing our robust optimization approach, a brief description of the evolutionary algorithms (EAs) [9] is described here. It is well known that a multi-objective optimization problem usually involves tradeoff among competing objective functions, and thus has a set of multiple "Pareto-optimal solutions," which are located on a tradeoff curve (in two-objective case) or surface (in three-objective case) in the objective function space. An optimizer for multi-objective problems needs to work well to find the Pareto-optimal solutions. The EAs are the optimization approaches which simulate the mechanism of natural evolution (selection, crossover, and mutation). Because the EAs deal with a population of multiple solutions (individuals) to be searched for in parallel, it is possible to find a global optimal solution without trapping in local optima. In addition, in multi-objective optimization problems, the EAs can find a set of the "nondominated solutions" in a single optimization run, based on the following "nondominance concept:"

When M objective functions $f_1(\boldsymbol{x}),\ f_2(\boldsymbol{x}),\ldots,\ f_M(\boldsymbol{x})$ must be minimized:

1) a solution $\boldsymbol{x}_{i1}$ is said to "dominate" the other solution $\boldsymbol{x}_{i2}$ if $f_k(\boldsymbol{x}_{i1}) \leqslant f_k(\boldsymbol{x}_{i2}),\ ^{\forall}k = 1, 2, \ldots, M$ and $f_k(\boldsymbol{x}_{i1}) < f_k(\boldsymbol{x}_{i2}),\ ^{\exists}k = 1, 2, \ldots, M$;
2) a solution $\boldsymbol{x}_{i1}$ is said to be "nondominated" if there is no other solution $\boldsymbol{x}_{i2}$ which dominates $\boldsymbol{x}_{i1}$.

The nondominated solutions correspond to compromised solutions among competing objective functions, which are substitutes of the Pareto-optimal solutions. It means that the EAs are powerful tool to reveal tradeoffs existing in the multi-objective optimization problems effectively (these are called the MOEAs [7]).

The idea of our robust optimization approach called DFMOSS is to incorporate a MOEA into DFSS to overcome its limitations as mentioned in Sec. II. In DFMOSS, the mean value (μ_f) and the standard deviation (σ_f) of the objective function $f(\boldsymbol{x})$ are dealt with as multiple objective functions and thus minimized separately (for $f(\boldsymbol{x})$ minimization problem) as follows

$$\text{Minimize: } \mu_f \tag{4a}$$

$$\sigma_f \tag{4b}$$

Figure 4 illustrates the flowchart of robust optimization using DFMOSS. There is no need to prespecify weighting factors w_μ and w_σ before optimization as in DFSS (Fig. 3), because DFMOSS deals with the multi-objective optimization problem given by Eqs. (4a) and (4b). There is no need to prespecify sigma level n either, because DFMOSS does not consider the constraint on sigma level n given by Eqs. (3a) and (3b) during the optimization process. The sigma level n satisfying Eqs. (3a) and (3b) can be evaluated from the robust optimal solutions in the post-processing, as shown in Fig. 5 (this will be described in the next paragraph). During the optimization process itself, multiple solutions (individuals) $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ are dealt with simultaneously using MOEA. For each individual $\boldsymbol{x}_i\ (i = 1, 2, \ldots, N)$, μ_f and σ_f are evaluated as two separate objective functions from $f(\boldsymbol{x}_i^p)\ (p = 1, 2, \ldots, S)$ which is sampled around $\boldsymbol{x}_i$. Then, better solutions are selected based on the nondominance concept between μ_f and σ_f, which are evaluated for all solutions. Solutions $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ for the next step are reproduced by crossover and mutation from the selected solutions. This optimization process is iterated until maximum number of iterations is reached or the front of nondominated solutions in μ_f–σ_f space does not change any more, and multiple robust optimal solutions are obtained. We simply preset the maximum number of iterations to be sufficiently large as a convergence criterion (specific operators used in the MOEA are given in the Sec. IV). The total number of function evaluations is $S \times N \times G$ for a single DFMOSS optimization run, so it indicates that the computational efficiency is almost same between the DFSS and DFMOSS optimizations when S, N, and G are the same. Note that the DFSS optimization has to perform multiple runs with different weighting factors to reveal tradeoff between optimality and robustness, while the DFMOSS needs a single run. Therefore, the DFMOSS has superior efficiency to the DFSS.

The postevaluation of sigma level n is illustrated in Fig. 5, where four robust optimal solutions (A, B, C, and D) obtained by a DFMOSS optimization were taken as examples. The shaded region indicates the area satisfying the constraint of 6σ robustness quality, i.e. points included in this region (solution C) have robustness quality equal to or above 6σ. Points outside this region (solutions A, B, and D) do not satisfy the constraint of 6σ robustness quality. Solution B for instance, is included in the area satisfying the constraint of 3σ robustness quality, thus inferior to solution C in terms of robustness. Therefore, the satisfied sigma level of each obtained robust optimal solution can be evaluated in a flexible sense, considering the tradeoff between optimality and robustness of design.

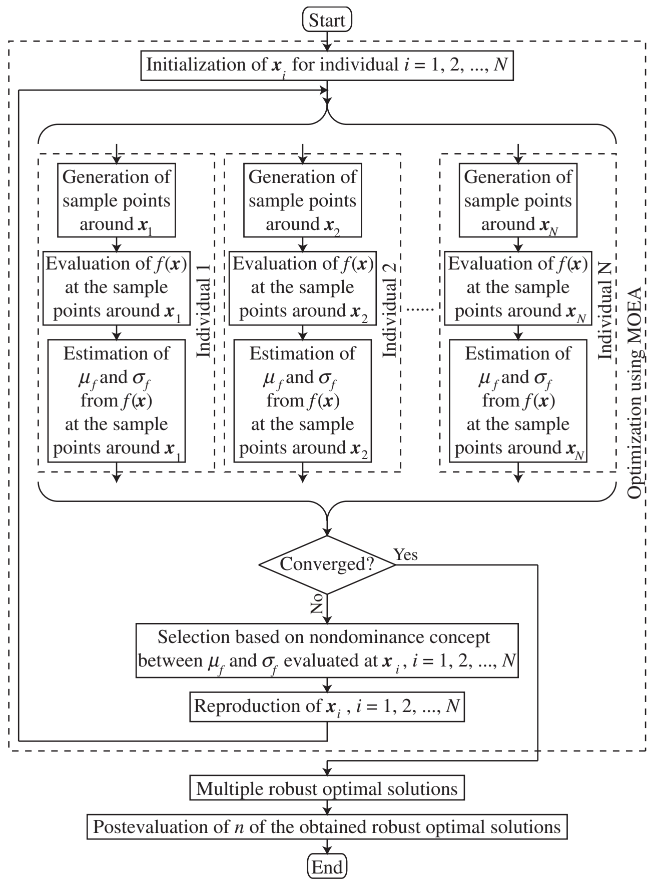


Fig. 4 Flowchart of robust optimization using DFMOSS.

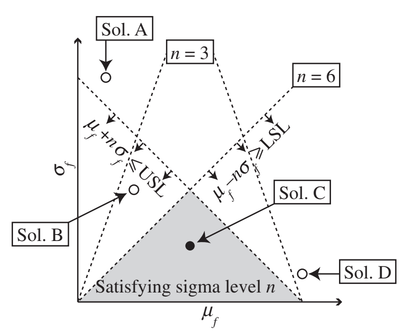


Fig. 5 Post-evaluation of sigma level *n*.

As an alternative to the above postevaluation, the satisfying sigma level $n\,(=\left(\mu_f - \mathrm{LSL}\right)/\sigma_f$ or $\left(\mathrm{USL} - \mu_f\right)/\sigma_f)$ can also be treated as an objective function for robustness improvement in the DFMOSS optimization as

$$\text{Minimize: } \mu_f \tag{5a}$$

$$\text{Maximize: } n \tag{5b}$$

This formulation can avoid the difficulty in careful prespecification of n and make possible to search for desirable robust optimal solutions with large n. In addition, this formulation can consider not only the robustness of objective function but also that of constraint function, while the previous formulation (Eqs. (4a) and (4b)) limits its consideration to the robustness of objective function. Therefore, the choice of DFMOSS formulations should be made according to whether a design problem has constraint functions to be robust or not.

As an additional comment, the DFMOSS optimization for a many-objective problem is described here. The DFMOSS formulation considering a single objective function (Eqs. (4a) and (4b)) can be simply extended to that considering M objective functions; $\mu_{f1}, \mu_{f2}, \ldots, \mu_{fM}$ and $\sigma_{f1}, \sigma_{f2}, \ldots, \sigma_{fM}$ must be considered as the objectives in the DFMOSS formulation. In a practical use, however, such formulation with too many objectives cannot be solved easily, and thus may require some modifications of the optimizer to improve its capability of global search, e.g., by adopting another concept except for nondominance. In addition, even if the DFMOSS optimization based on the above formulation can be solved, it may lead to difficulties in discussing output data in a high-dimensional objective function space. In future, we will have to work on how to handle the many-objective problems using the DFMOSS approach.

IV. Simulation Examples

The DFMOSS is applied to three robust optimization problems, and the simulation results obtained through DFMOSS are compared to those obtained through DFSS. The first is a test function problem (Sec. IV.B), the second is a welded beam design problem (Sec. IV.C), and the third is an airfoil design problem for a Mars airplane (Sec. IV.D). Before showing those results, details of numerical methods for DFSS and DFMOSS optimizations, which are used in those problems, are described in Sec. IV.A.

A. Numerical Methods

As for optimization algorithms, DFMOSS uses the MOEA while single-objective evolutionary algorithm (SOEA) was taken for DFSS in this study. The pseudocodes for each optimization are shown in Figs. 6a and 6b, respectively.

First, in both optimizations, N individuals $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ are determined as initial population at $G = 0$ by giving random fractions within a whole design space.

Then, S sample points $\boldsymbol{x}_i^1, \boldsymbol{x}_i^2 \ldots, \boldsymbol{x}_i^S$, which follows the normal distribution (its mean value and standard deviation differ in each problem, and those specific values are described in Sec. VI.B–IV.D) are generated around each individual $\boldsymbol{x}_i$ $(i = 1, 2, \ldots, N)$, and corresponding $f(\boldsymbol{x}_i^1), f(\boldsymbol{x}_i^2), \ldots, f(\boldsymbol{x}_i^S)$ are evaluated. From the sampled $f(\boldsymbol{x}_i^1), f(\boldsymbol{x}_i^2), \ldots, f(\boldsymbol{x}_i^S)$, statistical values μ_f and σ_f are estimated for $i = 1, 2, \ldots, N$. In the first and the second problems, the Monte Carlo simulation (MCS) with descriptive sampling (DS) [10] is used for the statistical evaluation. This technique can evaluate the statistics accurately, although it needs many sample points (usually $S \geqslant 1000$). The first and the second have only objective functions which can be evaluated promptly, thus the MCS can be applied to these problems. On the other hand, the third problem involves expensive computation for function evaluations, so the second-order Taylor's series expansion approach is used in this problem. This approach estimates the statistics μ_f and σ_f against dispersive design variables $\boldsymbol{x} = [x_1, x_2, \ldots, x_m]^T$ as follows

$$\mu_f = f(\boldsymbol{\mu_x}) + \frac{1}{2} \sum_{j=1}^{m} \frac{\partial^2 f}{\partial {x_j}^2} {\sigma_{xj}}^2 \tag{6a}$$

$$\sigma_f = \sqrt{\sum_{j=1}^{m} \left(\frac{\partial f}{\partial x_j}\right)^2 {\sigma_{xj}}^2 + \frac{1}{2} \sum_{j1,j2=1}^{m} \left(\frac{\partial^2 f}{\partial x_{j1} \partial x_{j2}}\right)^2 {\sigma_{xj1}}^2 {\sigma_{xj2}}^2} \tag{6b}$$

where $\boldsymbol{\mu_x}$ is a vector of user-specified mean values of $\boldsymbol{x}$, and σ_{xj} is user-specified standard deviation of the jth dispersive design variable x_j $(j = 1, 2, \ldots, m)$. The first and the second derivatives of $f(\boldsymbol{x})$ with respect to x_j in Eqs. (6a) and (6b) are evaluated by central differencing. This approach needs $S = 2m + 1$ sample points (if cross terms in Eq. (6b) are neglected), thus it can be applied to the third problem which involves expensive computation.

In DFSS optimization using SOEA, $w_\mu \mu_f + w_\sigma {\sigma_f}^2$, $\mu_f - n\sigma_f - \mathrm{LSL}$ $(\geqslant 0)$ and/or $\mu_f + n\sigma_f - \mathrm{USL}$ $(\leqslant 0)$ for each individual are evaluated as objective and constraint functions, respectively. From these function values, ranks

```
g = 0
for i = 1, 2, ..., N
    generate random individual x_i
end
   generate random individual

while g ⩽ G
    for i = 1, 2, ..., N
        for p = 1, 2, ..., S
            generate sample points x_i^p around x_i
                (following the normal distribution)
            evaluate f(x_i^p)
        end

        estimate statistics μ_fi and σ_fi from f(x_i^1), f(x_i^2), ..., f(x_i^S)
            (using MCS or Taylor's series expansion)
        evaluate objective function (w_μ μ_f + w_σ σ_f^2)_i
                    and constraints (μ_f − nσ_f − LSL)_i (⩾0)
                                    (μ_f + nσ_f − USL)_i (⩽0)
        evaluate fitness value F(x_i) from objective and constraint functions
            (using PBCH technique
                    Michaleswicz's nonlinear function)
    end

    select N parents based on F(x_i) for i = 1, 2, ..., N
        (using RWS or SUS)
    reproduce N children from N parents
        (using BLX-0.5
                random mutation or uniform mutation (rate = 20 %))
    replace current N individuals with N better individuals
        (using Best-N selection)

    g = g + 1
end
```

a) DFSS optimization using SOEA.

```
g = 0
for i = 1, 2, ..., N
    generate random individual x_i
end

while g ⩽ G
    for i = 1, 2, ..., N
        for p = 1, 2, ..., S
            generate sample points x_i^p around x_i
                (following the normal distribution)
            evaluate f(x_i^p)
        end

        estimate statistics μ_fi and σ_fi from f(x_i^1), f(x_i^2), ..., f(x_i^S)
            (using MCS or Taylor's series expansion)
        treat μ_fi and σ_fi as objective functions
        evaluate fitness value F(x_i) from objective functions
            (using Fonseca and Fleming's Pareto-ranking method
                    Fonseca and Fleming's fitness sharing
                    Michaleswicz's nonlinear function)
    end

    select N parents based on F(x_i) for i = 1, 2, ..., N
        (using RWS or SUS)
    reproduce N children from N parents
        (using BLX-0.5
                random mutation or uniform mutation (rate = 20 %))
    replace current N individuals with N better individuals
        (using Best-N selection)

    g = g + 1
end
```

b) DFMOSS optimization using MOEA.

Fig. 6 Pseudocodes used in the present robust optimizations.

are assigned to each individual by using the Pareto-optimality-based constraint-handling (PBCH) technique [11]. The PBCH technique is an extension of the Fonseca and Fleming's Pareto-ranking method [12]; this assigns ranks to each solution based on feasibility as well as Pareto-optimality by considering dominance relations in both objective and constraint function spaces, while the normal Pareto-ranking method does it by considering dominance relations in an objective function space only. In the second problem, the PBCH technique is used to treat six constraints, too. Based on the ranks, fitness value $F(\boldsymbol{x}_i)$ of each individual $i = 1, 2, \ldots, N$ are evaluated by using the Michaleswicz's nonlinear function [13] (fitness value of individuals with rank $= 1$ is set to be 0.1).

In DFMOSS optimization using MOEA, on the other hand, μ_f and σ_f for each individual are evaluated as two objective functions. Fitness value $F(\boldsymbol{x}_i)$ is evaluated by using the Fonseca and Fleming's Pareto-ranking method, as well as the Fonseca and Fleming's fitness sharing [7,12] (sharing parameter is set to be 1). The fitness sharing aims to ensure uniformity of resulting non-dominated solutions in the case of multi-objective optimization.

Then in each optimization, N parents are selected following the evaluated fitness values $F(\boldsymbol{x}_i)$ $(i = 1, 2, \ldots, N)$. This study adopts the roulette-wheel selection (RWS) [9] in the first and the second problems, while the stochastic universal sampling (SUS) [14] is used in the third problem. This is because the third problem has more design variables than the first and the second problems, and the SUS can ensure better variety of selected parents (i.e., more global solution search) than the RWS. From the selected N parents, N children are reproduced by the blended crossover (BLX-0.5) [15]. For the children, mutation is given at a 20% rate. The mutation range is set to be a whole design space (so-called random mutation [13]) in the first and the second problems, while the range is set to be 10% of the design space (so-called uniform mutation [7]) in the third problem. This is because large perturbation may lead to unpractical design variables (e.g., strange airfoils in which upper and lower surfaces intersect) in the third problem.

In each subsequent optimization, the alternation of generations is performed by the Best-N selection [16,17]. This method selects better N individuals as a new population for the next generation at $G+1$, from $2N$ individuals in which N individuals at the current generation at G and N children are included. Therefore, the Best-N selection ensures elitism. Then, G is updated to $G+1$, and the above process is iterated until G reaches a pre-set maximum number (population size N, sample size S, and maximum number of generations G are set to be different values in each problem, and those specific values are described in Sec. IV.B–VI.D).

In each DFSS or DFMOSS optimization for the first and the second problems, a total of 30 trials with different initial population are implemented, because the capabilities of DFSS and DFMOSS are fairly compared in a statistical manner among the trials.

B. Test Function Problem

1. Problem Definition

The present test function $f(x)$, which is developed by the current authors, is defined as

$$f(x) = -\exp\left(-\frac{|x|}{5}\right)\cos\left(\frac{2\pi x}{|x|^{0.1}}\right) \tag{7}$$

where the range of design variable is $-0.5 \leqslant x \leqslant 5$. For robust optimizations using DFSS and DFMOSS, convert this problem to the following robust optimization problems:

1) Robust optimization using DFSS. When x disperses around the design point following the normal distribution with its standard deviation of 0.1
   a. minimize: $w_\mu \mu_f + w_\sigma {\sigma_f}^2$
   b. subject to: $\mu_f + n\sigma_f \leqslant -0.2$
2) Robust optimization using DFMOSS. When x disperses around the design point following the normal distribution with its standard deviation of 0.1
   a. maximize: μ_f
   b. minimize: σ_f

where μ_f and σ_f are the mean value and the standard deviation of the dispersive objective function $f(x)$, and w_μ and w_σ are weighting factors.

Figure 7 shows the test function distribution ($f(x)$ against x). In a normal optimization problem where the objective function $f(x)$ should be minimized, it is clear that the optimal solution would be the point A ($x = 0$). In a robust optimization problem, however, the tradeoff between optimality and robustness should also be considered.

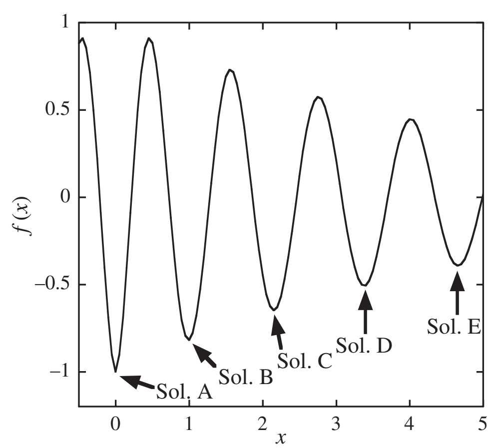
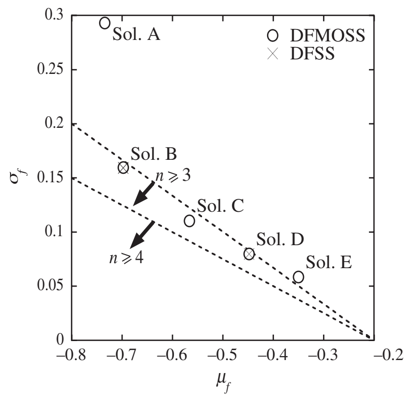


Fig. 7 Test function distribution.



DFSS			
$w_\mu : w_\sigma$	x	μ_f	σ_f
1000 : 1	3.3864	−0.44811	0.079737
100 : 1	3.3863	−0.44812	0.079737
10 : 1	0.99987	−0.69769	0.1592
1 : 1	0.99808	−0.69776	0.15927
1 : 10	0.99724	−0.69777	0.15934
1 : 100	0.99713	−0.69777	0.15935
1 : 1000	0.99712	−0.69777	0.15935
DFMOSS			
	x	μ_f	σ_f
	0.001737	−0.73406	0.29317
	1.0023	−0.69747	0.15923
	2.159	−0.56617	0.11038
	3.3835	−0.44815	0.079798
	4.6596	−0.3499	0.058399

Fig. 8 Comparison of robust optimal solutions obtained through the first trial of DFSS and DFMOSS optimizations in the test function problem.

Accordingly, the point A is the best in terms of optimality, but not in terms of robustness because the objective function $f(x)$ disperses widely with the dispersion of design variable x around this. The optimality becomes worse but the robustness becomes better at hollows for larger values of x, because the hollow becomes shallower and gentler. Therefore, the five points A, B, C, D, and E ($x = 0$, 1, 2.16, 3.38, and 4.66) depicted in Fig. 7 are all robust optimal solutions for this robust optimization problem.

In both DFSS and DFMOSS optimizations, population size N, sample size S, and maximum number of generations G are 32, 1000, and 100, respectively. In the DFSS optimization, the sigma level n is prespecified as 3σ. Seven optimization runs with different weighting factors ($w_\mu : w_\sigma = 1000:1$, $100:1$, $10:1$, $1:1$, $1:10$, $1:100$, and $1:1000$) have been performed with DFSS, while only one optimization run has been performed with DFMOSS without any prespecification of weighting factors and sigma level. Here, note that computational time taken by one optimization run using DFSS is nearly equal to that using DFMOSS, because N, S, and G are the same in both optimizations. This means that the seven optimization runs using DFSS took about seven times as much total computational time as one optimization run using DFMOSS.

2. Results

The robust optimal solutions (σ_f against μ_f) obtained through the first trial of DFSS and DFMOSS optimizations are depicted in Fig. 8. For the present test function problem, one can expect to find three robust optimal solutions with more than 3σ robustness quality ($x = 1$, 2.16, and 3.38). However, only two robust optimal solutions with more than 3σ robustness quality ($x = 1$ and 3.39) were obtained through DFSS, even though a total of seven optimization runs with different combinations of weighting factors have been performed. This indicates that the capability of DFSS to find robust optimal solutions depends heavily on the earlier setting of input parameters. In this case, the setting of sigma level as 3σ was appropriate by chance, but it is not always guaranteed that the DFSS will obtain the robust optimal solutions according to the advance setting of sigma level.

The DFMOSS, on the other hand, was able to find all five robust optimal solutions ($x = 0$, 1, 2.16, 3.38, and 4.66) in a single optimization run with no prior setting of input parameters. In addition, it can be easily understood in the postevaluation that the maximum sigma level which the robust optimal solutions have is actually more than 3σ.

Next, the robust optimal solutions obtained through all trials of DFSS and DFMOSS optimizations are listed in Table 1. The DFSS optimization fails in searching the solution C in three trials (1, 5, and 10). On the other hand, the DFMOSS can find all the solutions A, B, C, D, and E in all trials. It indicates that the DFMOSS ensures good consistency with the results, independent of the initial population.

Table 1 Comparison of robust optimal solutions obtained through all trials of DFSS and DFMOSS optimizations in the test function problem

DFSS

$w_\mu : w_\sigma$	Trial 1	2	3	4	5	6	7	8	9	10	11	...	30
1000 : 1	B	B	B	B	B	B	B	B	B	B	B		B
100 : 1	B	B	B	B	B	B	B	B	B	B	B		B
10 : 1	B	B	B	B	B	B	B	B	B	B	B		B
1 : 1	B	B	B	B	B	B	B	B	B	B	B	...	B
1 : 10	B	C	C	C	B	C	C	C	C	B	C		C
1 : 100	D	D	D	D	D	D	D	D	D	D	D		D
1 : 1000	D	D	D	D	D	D	D	D	D	D	D		D

DFMOSS

	Trial 1	2	3	4	5	6	7	8	9	10	11	...	30
	A	A	A	A	A	A	A	A	A	A	A		A
	B	B	B	B	B	B	B	B	B	B	B		B
	C	C	C	C	C	C	C	C	C	C	C	...	C
	D	D	D	D	D	D	D	D	D	D	D		D
	E	E	E	E	E	E	E	E	E	E	E		E

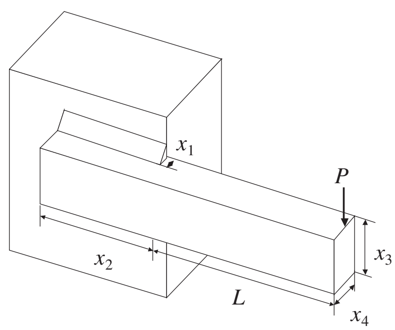


Fig. 9 Welded beam structure.

C. Welded Beam Design Problem [18,19]

1. Problem Definition

The welded beam structure is shown in Fig. 9. It consists of a beam and a weld required to secure the beam to the member. The objective is to find a feasible set of dimensions $\boldsymbol{x} = [x_1, x_2, x_3, x_4]^T$ to carry a certain load (P) and still have a minimum total fabrication cost. The objective function $f(\boldsymbol{x})$ is the total fabrication cost which mainly comprises setup cost, welding labor cost, and material cost

$$f(\boldsymbol{x}) = (1 + c_1)\, x_1^2 x_2 + c_2 x_3 x_4\, (L + x_2) \tag{8}$$

where c_1 and c_2 are the cost of unit volume of weld material and bar stock, respectively. The associated functional constraints are

$$\text{Subject to: } \tau_s(\boldsymbol{x}) - \tau_{s\max} \leqslant 0 \tag{9a}$$

$$\tau_b(\boldsymbol{x}) - \tau_{b\max} \leqslant 0 \tag{9b}$$

$$x_1 - x_4 \leqslant 0 \tag{9c}$$

$$c_1 {x_1}^2 + c_2 x_3 x_4 (L + x_2) - 5 \leqslant 0 \tag{9d}$$

$$\delta(\boldsymbol{x}) - \delta_{\max} \leqslant 0 \tag{9e}$$

$$P - P_c(\boldsymbol{x}) \leqslant 0 \tag{9f}$$

where

$$\tau_s(\boldsymbol{x}) = \sqrt{\tau_s'^2 + 2\tau_s'\tau_s''\frac{x_2}{2R} + \tau_s''^2}, \quad \tau_s' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau_s'' = \frac{PR}{J}\left(L + \frac{x_2}{2}\right),$$

$$R = \sqrt{\frac{{x_2}^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \quad J = 2\left\{\sqrt{2}x_1x_2\left[\frac{{x_2}^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\},$$

$$\tau_b(\boldsymbol{x}) = \frac{6PL}{x_4{x_3}^2}, \quad \delta(\boldsymbol{x}) = \frac{4PL^3}{E_v{x_3}^3x_4}, \quad P_c(\boldsymbol{x}) = \frac{4.013E_v\sqrt{{x_3}^2{x_4}^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E_v}{4E_h}}\right)$$

and

$$c_1 = 0.10471, \quad c_2 = 0.04811, \quad P = 6 \times 10^3 \text{ lb}, \quad L = 14 \text{ in}, \quad E_v = 3 \times 10^7 \text{ psi},$$

$$E_h = 1.2 \times 10^7 \text{ psi}, \quad \delta_{\max} = 0.25 \text{ in}, \quad \tau_{s\max} = 1.36 \times 10^4 \text{ psi}, \quad \tau_{b\max} = 3 \times 10^4 \text{ psi}$$

where $\tau_s(\boldsymbol{x})$, $\tau_b(\boldsymbol{x})$, $\delta(\boldsymbol{x})$, and $P_c(\boldsymbol{x})$ are weld shear stress, bar bending stress, bar end deflection, and bar buckling load, respectively. The ranges of design variables are $0.125 \leqslant x_1 \leqslant 5$ in, $0.1 \leqslant x_2 \leqslant 10$ in, $0.1 \leqslant x_3 \leqslant 10$ in, and $0.1 \leqslant x_4 \leqslant 5$ in. For robust optimizations using DFSS and DFMOSS, convert this problem to the following robust optimization problems:

1) Robust optimization using DFSS. When $\boldsymbol{x}$ disperses around the design point following the normal distribution with its standard deviation of 0.01
   a. minimize: $w_\mu \mu_f + w_\sigma {\sigma_f}^2$
   b. subject to: $\mu_f + n\sigma_f \leqslant 3$
2) Robust optimization using DFMOSS. When $\boldsymbol{x}$ disperses around the design point following the normal distribution with its standard deviation of 0.01
   a. maximize: μ_f
   b. minimize: σ_f

where μ_f and σ_f are the mean value and the standard deviation of dispersive fabrication cost $f(\boldsymbol{x})$, and w_μ and w_σ are weighting factors.

In both DFSS and DFMOSS optimizations, population size N, sample size S, and maximum number of generations G are 50, 1000, and 1000, respectively. In the DFSS optimization, the sigma level n is prespecified as 6σ, and seven optimization runs were performed with different weighting factors ($w_\mu : w_\sigma = 1000 : 1, 100 : 1, 10 : 1, 1 : 1, 1 : 10, 1 : 100$, and $1 : 1000$). In the DFMOSS optimization, a single optimization run was performed with no presetting of sigma level or weighting factors. Similar to the first problem, it means that the seven optimization runs using DFSS took about seven times as much total computational time as one optimization run using DFMOSS.

2. *Results*

The robust optimal solutions (σ_f against μ_f) obtained through DFSS and DFMOSS are depicted in Fig. 10, for the welded beam design problem. Although optimal solutions with more than 6σ robustness quality could be found by DFSS, these solutions distribute very locally despite optimization runs being performed seven times with different values of weighting factors. Essentially, these solutions include only two nondominated solutions. This indicates that the capability of DFSS to find robust optimal solutions globally is limited and dependent on the earlier setting of input parameters. Also in this case, the setting of sigma level as 6σ was appropriate by chance, but it is not always guaranteed that the DFSS will obtain the robust optimal solutions according to the earlier setting of the sigma level.

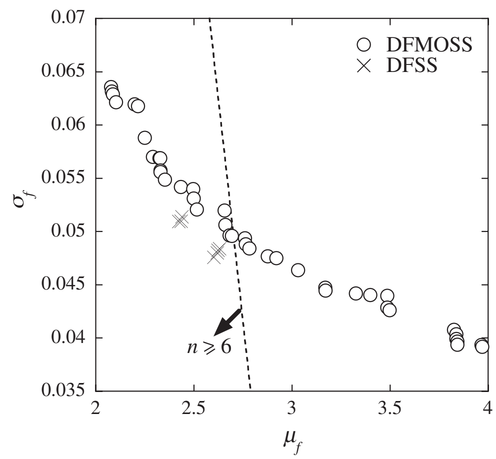


Fig. 10 Comparison of robust optimal solutions obtained through the first trial of DFSS and DFMOSS optimizations in the welded beam design problem.

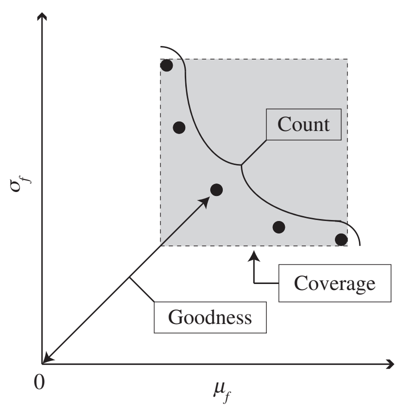


Fig. 11 Performance measures used in the welded beam design problem.

The DFMOSS, on the other hand, found many robust optimal solutions, distributed globally and uniformly, in a single optimization run. Also, the postevaluation revealed the maximum sigma level to be actually more than 6σ. Specifically, a total of 21 robust optimal solutions have been revealed with robustness quality superior to 6σ.

Next, the following three performance measures of nondominated solutions with more than 6σ robustness quality obtained through all trial of DFSS and DFMOSS optimizations are compared:

1) Count: number of nondominated solutions (larger is better);
2) Coverage: minimum area of a rectangle, which includes all non-dominated solutions in a μ_f–σ_f space (larger is better);
3) Goodness: minimum distance between a non-dominated solution and the origin [0,0] in a μ_f–σ_f space (smaller is better).

A definition of each measure is illustrated in Fig. 11.

Table 2 lists the best and the averaged values of the three measures among all trials. Regarding the "goodness," the DFSS and DFMOSS have almost the same performance. Regarding the "count" and the "coverage," on the other hand, the DFMOSS has better performance than the DFSS. These results indicate that the DFMOSS can search more robust optimal solutions, which distribute more widely in μ_f–σ_f space, than the DFSS. Also, the DFMOSS can obtain a set of robust optimal solutions, which has almost the same minimum distance from the origin in μ_f–σ_f space as that of DFSS.

Table 2 Comparison of performance measures of robust optimal solutions obtained through all trials of DFSS and DFMOSS optimizations in the welded beam design problem

		Count	Coverage	Goodness
DFSS	Best	7	1.5066×10^{-2}	1.7307
	Averaged	4.8	4.8577×10^{-3}	2.1232
DFMOSS	Best	41	2.7637×10^{-2}	1.7572
	Averaged	18.567	8.6375×10^{-3}	2.131

D. Airfoil Design Problem for Mars Airplane

1. Problem Definition

DFMOSS is applied to a robust airfoil design optimization for a future Mars airplane, to investigate the efficiency and capability of DFMOSS to reveal tradeoff design information considering optimality and robustness of aerodynamic performance against the variation of flight Mach number. The flight conditions adopted are those of NASA's "Airplane for Mars Exploration (AME);" [20] i.e., Reynolds number based on root chord length of 1.0×10^5, the angle of attack of 2.0 [deg], and freestream Mach number $M_\infty = 0.4735$. In addition, it is assumed that M_∞ disperses around the mean value of 0.4735 in a normal distribution with a standard deviation of 0.1. Here, the value of 0.1 as the standard deviation of M_∞ is nearly equal to the daily and seasonal variations of 22 m/s in the speed of westerly winds at the altitude of several kilometers over Mars [21], where the airplane is assumed to fly. For robust optimization considering robustness of lift to drag ratio l/d (l is lift and d is drag) using DFSS and DFMOSS, convert this problem to the following robust optimization problems:

1) Robust optimization using DFSS. When M_∞ disperses around 0.4735 following the normal distribution with its standard deviation of 0.1
   a. maximize: $-w_\mu \cdot$ (mean value of l/d) $+ w_\sigma \cdot$ (variance of l/d)
   b. subject to: (mean value of l/d) $- n \cdot$ (standard deviation of l/d) $\geqslant 42$
2) Robust optimization using DFMOSS. When M_∞ disperses around 0.4735 following the normal distribution with its standard deviation of 0.1,
   a. maximize: mean value of l/d
   b. minimize: standard deviation of l/d

These optimizations aim at finding the airfoil configuration which can assure the expected flight range even when the flight Mach number disperses widely around its design point.

An airfoil configuration is defined by the B-spline curves, with three fixed points corresponding to the leading (one point) and trailing (two points) edges, plus six control points whose coordinates can be specified flexibly, as shown in Fig. 12 (here, C is the airfoil chord length). The design variables are chordwise (X) and vertical (Y) coordinates of the six control points, therefore the number of design variables is twelve. Such definition based on the B-spline curves has some advantages; e.g., the second-order derivatives of the coordinates along the B-spline curves are continuous, various airfoil configurations can be expressed, and the definition of an initial design space is intuitive [22]. This paper aims to discuss only the aerodynamic effect. A consideration of additional constraints besides the aerodynamics may confuse the current discussions due to the interaction between the aerodynamics and other disciplines.

Aerodynamic performance of an airfoil is evaluated by computational fluid dynamics (CFD) simulation. The governing equations for CFD simulation are two-dimensional Farve-averaged compressible thin-layer Navier–Stokes equations. The lower-upper alternate directional implicit factorization algorithm [23] is used for the time integration. The inviscid terms of numerical fluxes are evaluated by the simple high-resolution upwind scheme [24]. In the inviscid terms, high-order accuracy is obtained by the third-order upwind-biased monotone upstream-centered schemes for conservation law interpolation [25] based on the primitive variables with van Albada differentiable limiter [26]. The viscous terms are evaluated by the second-order central differencing, and the turbulent viscosity is modeled by the Baldwin-Lomax algebraic turbulence model [27]. In the present study, C type grid as shown in Fig. 13 is used. The number of grid points is 251 in the direction around the airfoil (211 points over the airfoil surface), 51 in the direction normal to the airfoil surface, and the total number of grid points is 12,801.

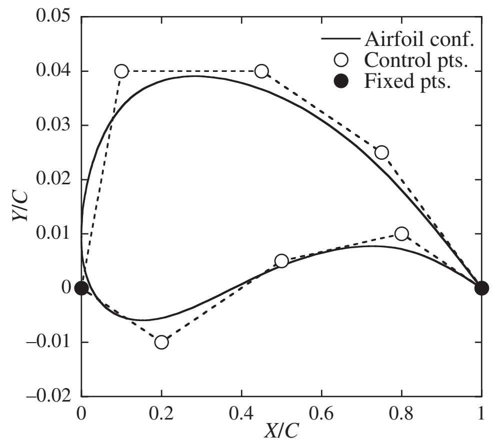


Fig. 12 Definition of airfoil configuration based on the B-spline curves.

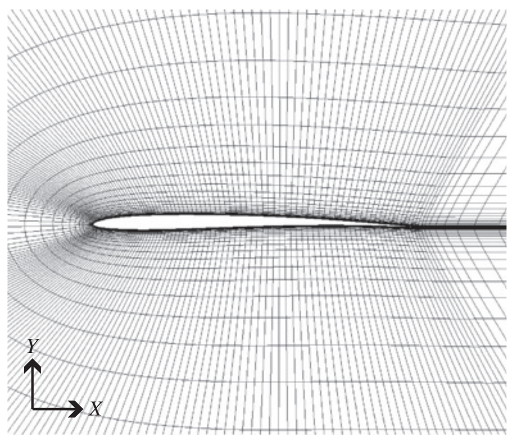


Fig. 13 Grid distribution.

In both DFSS and DFMOSS optimizations, population size N and maximum number of generations G are 64 and 100, respectively. This problem has a dispersive design variable corresponds to the flight Mach number M_∞, i.e., $m = 1$, $x = M_\infty$, $\mu_x = 0.4735$, and $\sigma_x = 0.1$ in Eqs. (6a) and (6b). Therefore, μ_f and σ_f are estimated from $f(x)$ evaluated at three conditions $x = M_\infty = 0.3735$, 0.4735, and 0.5735 (i.e., $S = 3$). In the DFSS optimization, the sigma level n is set to 3σ, and three optimizations are performed with different weighting factors ($w_\mu : w_\sigma = 10 : 1$, $1 : 1$, and $1 : 10$). In the DFMOSS optimization, on the other hand, only one optimization is performed without any prespecification of weighting factors and sigma level.

The computational time required for one evaluation of aerodynamic performance of an airfoil using the CFD simulation is about 5 min with one processor of NEC SX-6 computing system owned by the Institute of Space and Astronautical Science of Japan Aerospace Exploration Agency. In the present study, the optimizer distributes the multiple evaluators corresponding to the multiple individuals of EA into 32 processors of this computing system in parallel. Therefore, the total computation time required for one present robust aerodynamic design optimization using DFMOSS can be reduced to about 56 h, while it takes about 135 h to perform the present three robust optimizations using DFSS.

2. Results

The robust optimal solutions (standard deviation of l/d against mean value of l/d) obtained through DFSS and DFMOSS are depicted in Fig. 14, for the airfoil design problem. The DFSS found three robust optimal solutions

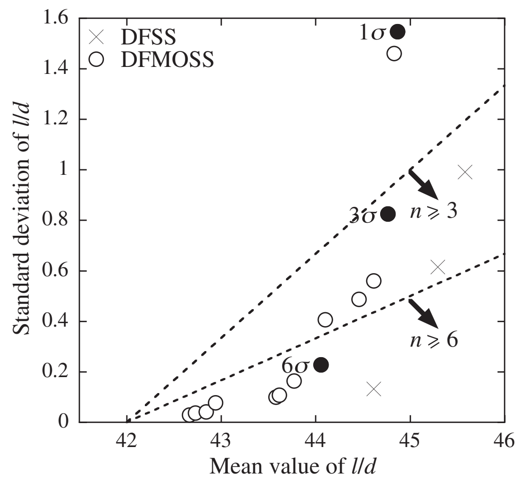
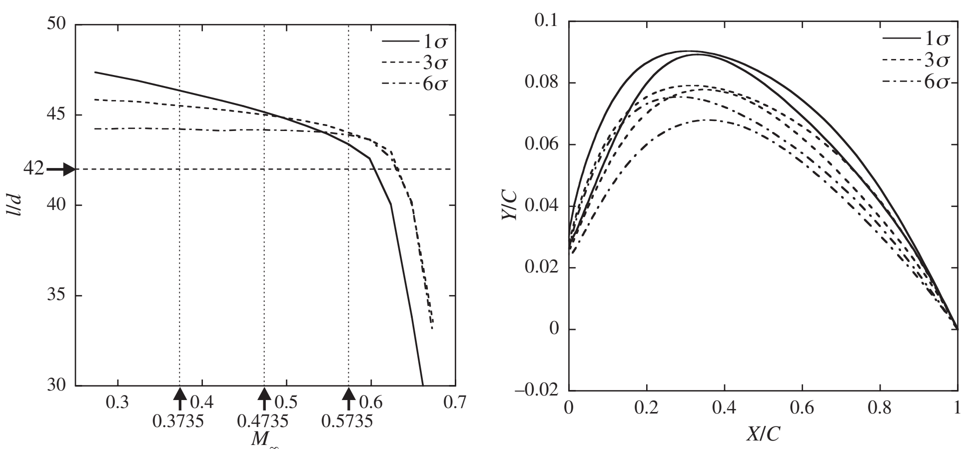


Fig. 14 Comparison of robust optimal solutions obtained through DFSS and DFMOSS in the airfoil design problem.

with more than 3σ robustness quality, but these solutions distribute very narrowly and sparsely. This indicates again that the capability of DFSS to reveal global tradeoff relation between optimality (mean value of l/d) and robustness (standard deviation of l/d) is limited and more optimization runs with different combinations of weighting factors are required to obtain more detailed tradeoff information. Fortunately the pre-specified value of sigma level as 3σ was appropriate this time too, but it is not always guaranteed for the DFSS to obtain the robust optimal solutions according to the pre-specified sigma level. On the other hand, DFMOSS found multiple a total of 18 robust optimal solutions distributed globally and uniformly in the design space in a single optimization run. From this distribution of robust optimal solutions obtained through DFMOSS, global tradeoff information between optimality and robustness can be easily understood; e.g., the maximum sigma level of l/d of the obtained solutions is more than 6σ when the lower specification limit of l/d is set to 42, and the standard deviation of l/d increases drastically when the mean value of l/d becomes larger than 44.5. In the present case, the robust optimization using DFSS found better robust optimal solutions (located in the lower right section of Fig. 14) than that using DFMOSS. This is because the DFMOSS searched an unexpectedly larger design space, producing impractical solutions with good robustness but extremely bad l/d optimality. However, this can be easily avoided with additional constraints that eliminate impractical regions in the design space.



a) l/d variation over $M\infty$ b) airfoil configurations

Fig. 15 Comparison of three robust optimal solutions obtained through DFMOSS in the airfoil design problem.

Hereafter, three robust optimal solutions with l/d robustness quality of 1σ, 3σ, and 6σ obtained through DFMOSS (shown by closed circles in Fig. 14) will be compared and discussed. First, Fig. 15a shows the variation of l/d against M_∞ for these three robust optimal solutions. In the robust optimal solution with 1σ robustness quality, l/d decreases more sharply than the other two solutions when M_∞ is incremented beyond the design point; and it falls below the lower specification limit of 42 sooner than the solutions with 3σ and 6σ robustness quality. In other words, more robust optimal solutions do have a slightly smaller l/d at the design point $M_\infty = 0.4735$, but on the other hand they present a more stable behavior of l/d against increments in M_∞. These results demonstrate the DFMOSS's capability to find the multiple airfoil design configurations with various optimality and robustness qualities in a single optimization run.

Next, Fig. 15b shows the airfoil configurations of these three robust optimal solutions with 1σ, 3σ, and 6σ robustness qualities obtained through DFMOSS. It indicates that maximum camber is one of the major tradeoff factors between l/d and robustness improvements. The reason is that an airfoil with a smaller maximum camber realizes a smaller increment in pressure drag owing to a shock wave, and eventually improves the robustness in l/d against the increment in M_∞.

V. Concluding Remarks

A new robust optimization approach for robust design, DFMOSS, has been developed and applied to three robust optimization problems. The DFMOSS couples the ideas of DFSS and MOEA for a more practical and efficient robust design optimization. The three robust design optimization problems (test function problem, welded beam problem, and airfoil design problem for a Mars airplane) indicated that DFMOSS has superior characteristics than the DFSS regarding the global search for multiple compromised solutions without any prior input parameter tuning. The DFMOSS can also reveal detailed tradeoff information between optimality and robustness of performance in a single optimization run. Actually, for the airfoil design problem, the obtained tradeoff information has even led to practical knowledge on airfoil design concept; i.e., an airfoil with a smaller maximum camber has improved robustness of lift to drag ratio against the variation of flight Mach number.

In future works, handling techniques of DFMOSS for many-objective problems should be considered and demonstrated. In addition, some techniques for reducing function evaluation time (e.g., surrogate models) is also expected for an efficient DFMOSS robust optimization, which involves many-point sampling for statistics evaluation. Through such considerations, the DFMOSS will enable its application to various real-world design problems.

References

[1] Gunawan, S., and Azarm, S., "Multi Objective Robust Optimization Using a Sensitivity Region Concept," *Structural and Multidisciplinary Optimization*, Vol. 29, No. 1, 2005, pp. 50–60.
doi: 10.1007/s00158-004-0450-8

[2] Li, M., Azarm, S., and Aute, V., "A Multi-Objective Genetic Algorithm for Robust Design Optimization," *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, ACM Press, New York, NY, June 2005, pp. 771–778.

[3] Deb, K., and Gupta, H., "Searching for Robust Pareto-Optimal Solutions in Multi-objective Optimization," *Evolutionary Multi-Criterion Optimization, 2nd International Conference*, edited by C. A. Coello Coello, A. H. Aguirre, and E. Zitzler, Lecture Notes in Computer Science 3410, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 150–164.

[4] Ong, Y.-S., Nair, P. B., and Lum, K. Y., "Max–Min Surrogate-Assisted Evolutionary Algorithm for Robust Design," *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 4, August 2006, pp. 392–404.

[5] Ray, T., "Constrained Robust Optimal Design using a Multi-objective Evolutionary Algorithm," *Proceedings of the 2002 Conference on Genetic and Evolutionary Computation*, Vol. 1, IEEE Publications, Piscataway, NJ, May 2002, pp. 419–424.

[6] Jin, Y., and Sendhoff, B., "Trade-off between Performance and Robustness: An Evolutionary Multi-objective Approach," *Evolutionary Multi-Criterion Optimization, 2nd International Coference*, edited by C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, Lecture Notes in Computer Science 2632, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 237–251.
doi: 10.1007/3-540-36970-8_17

[7] Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley, Chichester, UK, 2001.

[8] *iSIGHT Reference Guide Version 7.1*, Engineous Software, 2002, pp. 220–233.

[9] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Longman, Reading, MA, 1989.

[10] Saliby, E., “Descriptive Sampling: A Better Approach to Monte Carlo Simulation,” *Journal of the Operational Research Society*, Vol. 41, No. 12, 1990, pp. 1133–1142.
doi: 10.1038/sj/jors/0411205

[11] Oyama, A., Shimoyama, K., and Fujii, K., “New Constraint-Handling Method for Multi-Objective Multi-Constraint Evolutionary Optimization and Its Application to Space Plane Design,” *EUROGEN 2005*, edited by R. Schilling, W. Hasse, J. Periaux, H. Baier, and G. Bugeda, FLM, Munich, September 2005.

[12] Fonseca, C. M., and Fleming, P. J., “Genetic Algorithms for Multi-objective Optimization: Formulation, Discussion and Generalization,” *Proceedings of the 5th International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, 1993, pp. 416–423.

[13] Michalewicz, Z., *Genetic Algorithms + Data Structure = Evolution Programs*, 3rd revised and extended ed., Springer-Verlag, Berlin, Heidelberg, New York, 1996.

[14] Baker, J. E., “Reducing Bias and Inefficiency in the Selection Algorithm,” *Proceedings of the 2nd International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, 1987, pp. 41–49.

[15] Eshelman, L. J., and Schaffer, J. D., “Real-coded Genetic Algorithms and Interval Schemata,” *Foundations of Genetic Algorithms 2*, Morgan Kaufmann, San Mateo, CA, 1993, pp. 187–202.

[16] Eshelman, L. J., “The CHC Adaptive Search Algorithm: How to Have Safe When Engaging in Nontraditional Genetic Recombination,” *Foundations of Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, 1991, pp. 265–283.

[17] Tsutsui, S., and Fujimoto, Y., “Forking Genetic Algorithms with Blocking and Shrinking modes (fGA),” *Proceedings of the 5th International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, 1993, pp. 206–213.

[18] Deb, K., “Optimal Design of a Welded Beam via Genetic Algorithms,” *AIAA Journal*, Vol. 29, No. 11, November 1991, pp. 2013–2015.
doi: 10.2514/3.10834

[19] Mezura-Montes, E., Coello Coello, C. A., and Landa-Becerra, R., “Engineering Optimization Using a Simple Evolutionary Algorithm,” *Proceedings of the 15th International Conference on Tools with Artificial Intelligence*, IEEE Computer Society, Washington, DC, Sacramento, CA, November 2003, pp. 149–156.

[20] Hall, D. W., Parks, R. W., and Morris, S., “Airplane for Mars Exploration,” Tech. rep., NASA Ames Research Center, Moffett Federal Airfield, CA, May 1997, http://www.redpeace.org/final_report.pdf [retrieved 28 June 2008].

[21] Smith, M. D., Pearl, J. C., Conrath, B. J., and Christensen, P. R., “Thermal Emission Spectrometer Results: Mars Atmospheric Thermal Structure and Aerosol Distribution,” *Journal of Geophysical Research*, Vol. 106, No. E10, October 2001, pp. 23929–23945.
doi: 10.1029/2000JE001321

[22] Oyama, A., Obayashi, S., Nakahashi, K., and Hirose, N., “Fractional Factorial Design of Genetic Coding for Aerodynamic Optimization,” AIAA, Reston, VA, 1999; AIAA Paper 99–3298.

[23] Fujii, K., and Obayashi, S., “Practical Applications of New LU–ADI Scheme for the Three-Dimensional Navier–Stokes Computation of Transonic Viscous Flows,” AIAA, Reston, VA, 1986, AIAA Paper 86–0513.

[24] Shima, E., and Jounouchi, T., “Role of CFD in Aeronautical Engineering (No. 14): AUSM Type Upwind Schemes,” *Proceedings of the 14th NAL Symposium on Aircraft Computational Aerodynamics*, NAL SP–34, National Aerospace Laboratory of Japan, Tokyo, January 1997, pp. 7–12.

[25] van Leer, B., “Towards the Ultimate Conservation Difference Scheme. V. A Second-Order Sequel to Godunov’s Method,” *Journal of Computational Physics*, Vol. 32, 1979, pp. 101–136.

[26] Anderson, W. K., Thomas, J. L., and van Leer, B., “Comparison of Finite Volume Flux Vector Spllittings for the Euler Equations,” *AIAA Journal*, Vol. 24, No. 9, September 1986, pp. 1453–1460.
doi: 10.2514/3.9465

[27] Baldwin, B. S., and Lomax, H., “Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows,” AIAA Reston, VA, 1978, AIAA Paper 78–257.

Christopher Rouff
Associate Editor