Trade-Off between Performance and Robustness: An Evolutionary Multiobjective Approach

Yaochu Jin and Bernhard Sendhoff

Honda Research Institute Europe GmbH
63073 Offenbach/Main, Germany
{yaochu.jin,bs}@honda-ri.de

Abstract. In real-world applications, it is often desired that a solution is not only of high performance, but also of high robustness. In this context, a solution is usually called robust, if its performance only gradually decreases when design variables or environmental parameters are varied within a certain range. In evolutionary optimization, robust optimal solutions are usually obtained by averaging the fitness over such variations. Frequently, maximization of the performance and increase of the robustness are two conflicting objectives, which means that a trade-off exists between robustness and performance. Using the existing methods to search for robust solutions, this trade-off is hidden and predefined in the averaging rules. Thus, only one solution can be obtained. In this paper, we treat the problem explicitly as a multiobjective optimization task, thereby clearly identifying the trade-off between performance and robustness in the form of the obtained Pareto front. We suggest two methods for estimating the robustness of a solution by exploiting the information available in the current population of the evolutionary algorithm, without any additional fitness evaluations. The estimated robustness is then used as an additional objective in optimization. Finally, the possibility of using this method for detecting multiple optima of multimodal functions is briefly discussed.

1 Motivation

The search for robust optimal solutions is of great significance in real-world applications. Robustness of an optimal solution can usually be discussed from the following two perspectives:

- The optimal solution is insensitive to small variations of the design variables.
- The optimal solution is insensitive to small variations of environmental parameters. In some special cases, it can also happen that a solution should be optimal or near-optimal around more than one design point. These different points do not necessarily lie in one neighborhood.

Mostly, two methods have been used to increase the robustness of a solution [1,2].

C.M. Fonseca et al. (Eds.): EMO 2003, LNCS 2632, pp. 237–251, 2003.

- Optimization of the expectation of the objective function in a neighborhood around the design point. If the neighborhood is defined using a probability distribution $\phi(\boldsymbol{z})$ of a variation parameter $\boldsymbol{z}$, an effective evaluation [3] function can be defined using the original evaluation function f as

$$f_{eff} = \int_{-\infty}^{\infty} f(\boldsymbol{x}, \boldsymbol{z})\, \phi(\boldsymbol{z})\, d\boldsymbol{z}, \tag{1}$$

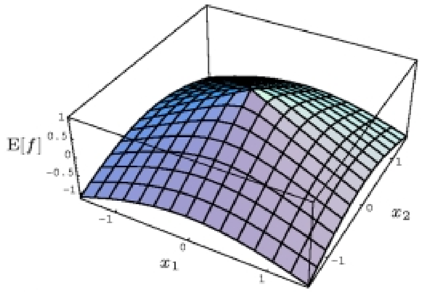
  where $\boldsymbol{x}$ is the design variable.
- Optimization of the second order moment or higher order moments of the evaluation function. For example, minimization of the variance of the evaluation function over the neighborhood around the design point has been used to maximize the robustness of a solution.

Unfortunately, the expectation based measure does not sufficiently take care of fluctuations of the evaluation function as long as these fluctuations are symmetric around the average value. At the same time, a purely variance based measure does not take the absolute performance of the solution into account. Thus, it can only be employed in combination with the original quality function or with the expectation based measure. Different combinations of objectives are possible:

- maximizing the expectation and maximizing the original function, for example in [1];
- maximizing the expectation and minimizing the variance, for example in [4];
- maximizing the original function and minimizing the variance.

Since robustness and performance (even if its measure is expectation based), are often exclusive objectives, see Figure 1 for an example, it makes sense to analyze this problem in the framework of multicriteria optimization. In this way, the user can get a better understanding of the relation between robustness and performance for the optimization problem at hand. Besides, the Pareto front can provide the user with valuable information about the stability of the solutions.

In this paper, we employ two variance based measures which are outlined in Section 3 as robustness objectives. Both measures use the information which is already available within the population to estimate the variance. Thus, no additional fitness evaluations are necessary, which is very important when fitness evaluation is computationally expensive, such as in aerodynamic design optimization problems [5]. In Section two, we will briefly review some of the expectation based approaches to searching for robust optimal solutions. The multiobjective optimization algorithm used in this paper, the dynamic weighted aggregation method proposed in [6,7], is described in Section 4. Simulation results on two test problems are presented in Section 5 to demonstrate the effectiveness of the proposed method. A summary of the method and a brief discussion of future work conclude the paper, where a simple example of detecting multiple optima using the proposed method is also provided.



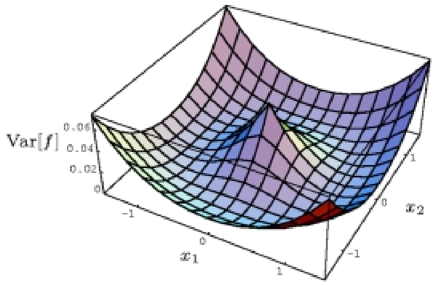


Fig. 1. Example for the trade-off between average and variance. Figure (a) shows the average and (b) the variance of function $f(\mathbf{x}) = a - (z+1)\|\mathbf{x}\|^{\alpha} + z, z \sim \mathcal{N}(0, \epsilon_z^2), a = 5, \alpha = 1, \epsilon_z = 0.25$. The maximum of the average is given for $\mathbf{x} = (0, 0)$, whereas the variance is minimized for $\|\mathbf{x}\| = 1$

2 Expectation-Based Search for Robust Solutions

In evolutionary optimization, efforts have been made to obtain optimal solutions that are insensitive to small changes in the design variables. Most of these approaches are mainly based on the optimization of the expectation of the fitness function. The calculation of the expected performance is usually not trivial in many real-world applications.

To estimate the expected performance, one straightforward way is to calculate the fitness of a solution ($\boldsymbol{x}$) by averaging several points in its neighborhood [8,9,10,11,12]:

$$\tilde{f}(\boldsymbol{x}) = \frac{\sum_{i=1}^{N} w_i f(\boldsymbol{x} + \boldsymbol{\Delta x}_i)}{\sum_{i=1}^{N} w_i}, \tag{2}$$

where $\boldsymbol{x}$ denotes a vector of design variables and possibly some environmental parameters, $i = 1, 2, ..., N$ is the number of points to be evaluated. $\Delta\boldsymbol{x}_i$ is a vector of small numbers that can be generated deterministically or stochastically and w_i is the weight for each evaluation. In the simplest case, all the weights are set equally to 1. If the $\boldsymbol{\Delta x}_i$ are random variables $\boldsymbol{z}$ and are drawn according to a probability distribution $\phi(\boldsymbol{z})$, we obtain in the limit $N \to \infty$, the effective evaluation function f_{eff}, equation 1.

One problem of the averaging method for estimating the expected performance is the increased computational cost. To alleviate this problem, several ideas have been proposed to use the information in the current or in previous populations [12] to avoid additional fitness evaluations. Note that throughout this paper, the terminology *population* is used as defined in evolutionary algorithms[1]. An alternative is to construct a statistical model for the estimation of

[1] In statistics, a population is defined as any entire collection of elements under investigation, while a sample is a collection of elements selected from the population.

the points in the neighborhood using the historical data [13]. Statistical tests have been used to estimate how many samples are needed to decide which solutions should be selected for the next generation [14].

Besides the averaging methods, it has been showed in [3] that the "perturbation" of design variables in each fitness evaluation leads to the maximization of the effective fitness function, equation 1, under the assumption of linear selection (in [3] the schema theorem is used as a basis for the mathematical proof, however, it can be shown that the important assumption is the linearity of the selection operator and an infinite population. Note that the "perturbation" method is equivalent to the averaging method for $N = 1$ and stochastic $\boldsymbol{\Delta x}$.

Whereas most methods in evolutionary optimization consider the robustness with respect to the variations of design variables, the search for robust solutions that are insensitive to environmental parameters has also been investigated [15]. An additional objective function has been defined at two deterministic points symmetrically located around the design point. Let a define an environmental parameter, and the design point is given by $a = a_1$. Thus, the first objective (performance) is defined by $f(\boldsymbol{x}, a_1)$. As a second objective f_2, the following deterministic function is used:

$$f_2(\boldsymbol{x}) = f(\boldsymbol{x}, a_1 + \Delta a) + f(\boldsymbol{x}, a_1 - \Delta a). \tag{3}$$

Note that deterministic formulations of a robustness measure like in equation 3 seems only sensible in the special case when a fixed number of design conditions can be determined. In most cases where parameters can vary within an interval, a stochastic approach seems to be more sensible.

A general drawback of the expectation based methods is that (with the exception of the last example) only one objective has been used. As discussed in Section 1, it is necessary to combine two objectives to search for robust solutions, where a trade-off between performance and robustness exists, as often occurs in many real-world applications. In this case, it is able to present a human user with a set of solutions trading off between the robustness and the optimality, from which the user has to make a choice according to the need of the application. A method for achieving multiple robust solutions has been suggested in [3] using the sharing method suggested in [16]. However, no information on the relative robustness increase and performance decrease of the solutions is available and thus, no trade-off decisions can be made on the obtained solutions.

3 Variance-Based Measures for Robustness

The search for robust optimal solutions has been widely investigated in the field of engineering design [17]. Consider the following unconstrained minimization problem:

$$\text{minimize } f = f(\boldsymbol{a}, \boldsymbol{x}), \tag{4}$$

where $\boldsymbol{a}$ and $\boldsymbol{x}$ are vectors of environmental parameters and design variables. For convenience, we will not distinguish between environmental parameters and design variables and hereafter, both are called design variables denoted uniformly with $\boldsymbol{x}$.

Now consider the function $f(\boldsymbol{x}) = f(x_1, x_2, ..., x_n)$, where the x_i's are n design variables and function f is approximated using its first-order Taylor expansion about the point $(\mu_{x_1}, \mu_{x_2}, ..., \mu_{x_n})$:

$$f \approx f(\mu_{x_1}, \mu_{x_2}, ..., \mu_{x_n}) + \sum_{i=1}^{n} \left[\frac{\partial f}{\partial x_i}(\mu_{x_1}, \mu_{x_2}, ..., \mu_{x_n}) \right] \cdot (x_i - \mu_{x_i}), \tag{5}$$

where $\mu_{x_i}, i = 1, 2, ..., n$ is the mean of x_i. Thus, the variance of the function can be derived as follows:

$$\sigma_f^2 = \sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} \right)^2 \sigma_{x_i}^2 + \sum_{i=1}^{n} \sum_{j=1, i \neq j}^{n} \left(\frac{\partial f}{\partial x_i} \right) \left(\frac{\partial f}{\partial x_j} \right) \sigma_{x_i x_j}, \tag{6}$$

where $\sigma_{x_i}^2$ is the variance of x_i and $\sigma_{x_i x_j}$ is the covariance between x_i and x_j. Recall that the function has to be evaluated using the mean value of the variables. If the design variables are independent of each other, the resulting approximated variance is

$$\sigma_f^2 = \sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} \right)^2 \sigma_{x_i}^2. \tag{7}$$

A measure for robustness of a solution can be defined using the standard deviation of the function and that of the design variables as:

$$f^R = \frac{1}{n} \sum_{i=1}^{n} \frac{\sigma_f}{\sigma_{x_i}}. \tag{8}$$

It should be pointed out that with this definition of robustness, the smaller the robustness measure, the more robust the solution is. In other words, the search for robust optimal solutions can now be formulated as a multiobjective optimization problem where both the fitness function and the robustness measure are to be minimized.

In robust design, the variation of the objective function in the presence of small variations in the design variables is the major concern. Therefore, it is reasonable to discuss the variance of the function defined in equation (7) in a local sense. Take a one-dimensional function $f(x)$ for example, as shown in Fig. 2. If the robustness of a target point x_j is considered, the function is then expanded in a Taylor series about $x = \mu_{x_j} = x_j$, which assumes that the variations of the design variable are zero-mean. Similarly, if the robustness of x_k is to be evaluated, the function will be expanded about $x = \mu_{x_k} = x_k$, refer to Fig. 2. In the figure, $\mu_{f,j}$ and $\mu_{f,k}$ denote the mean of the function calculated around the point x_j and x_k, respectively.

In the following, an estimation of the robustness measure based on the fitness evaluations in the current population will be proposed. Suppose the population size is λ, and N_j $(1 \leq N_j \leq \lambda)$ individuals are located in the neighborhood of

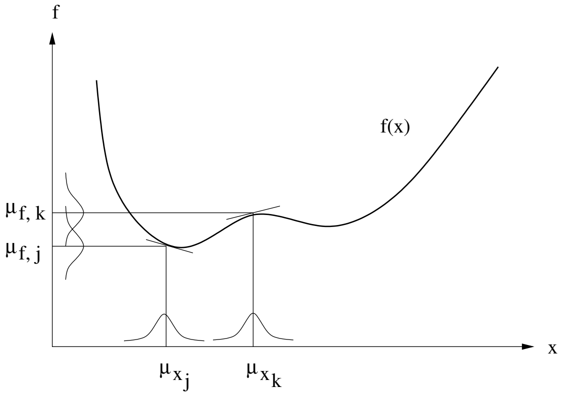


Fig. 2. Illustration of the local variations in the design variables

the j-th individual. The robustness of the j-th individual can be approximated by

$$\text{Robust measure 1: } f_j^R = \frac{1}{n}\sum_{i=1}^{n}\frac{\bar{\sigma}_{f,j}}{\sigma_{x_i}}, \tag{9}$$

where $\bar{\sigma}_{f,j}$ is an estimation of the variance of the j-th individual according to equation (7):

$$\begin{aligned}\bar{\sigma}_{f,j}^2 &= \sum_{i=1}^{n}\left(\frac{\partial f}{\partial x_i}\right)^2 \sigma_{x_i}^2 \\ &\approx \sum_{i=1}^{n}\left(\frac{1}{N_j}\sum_{k\in D_j}\frac{f_j - f_k}{x_{i,j} - x_{i,k}}\right)^2 \sigma_{x_i}^2, \mathrm{k} \neq j,\end{aligned} \tag{10}$$

where $x_{i,j}$ and $x_{i,k}$ denote the i-th element of $\boldsymbol{x}$ of the j-th and k-th individuals, and D_j denotes a set of the individuals that belong to the neighborhood of the j-th individual. The neighborhood of j-th individual D_j is defined using the Euclidean distance between the individual $\boldsymbol{x}_k, k = 1, 2, ..., \lambda$ and the j-th individual $\boldsymbol{x}_j$:

$$D_j : k \in D_j, \text{ if } d_{jk} \leq d^2, \quad 1 \leq k \leq \lambda, \; d_{jk} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{i,j} - x_{i,k})^2}, \tag{11}$$

where $k = 1, 2, ..., \lambda$ is the index for the k-th individual, λ is the population size of the evolutionary algorithm, d_{jk} is the Euclidean distance between individual j and k, and d is a threshold to be specified by the user according to the requirements in real applications. This constant should be the same for all individuals.

Actually, a more direct method for estimating the robustness measure can be used. Using the current population and the definition of the neighborhood, the robustness measure of the j-th individual can be estimated by dividing the local standard deviation of the function by the average local standard deviation of the variables. Assume N_j $(1 \leq N_j \leq \lambda)$ is the number of individuals in the neighborhood of the j-th individual in the current population, then the local variance of the function corresponding to the j-th individual in the population can be estimated as follows:

$$\mu_{f,j} = \frac{1}{N_j} \sum_{k \in D_j} f_k, \tag{12}$$

$$\sigma^2_{f,j} = \frac{1}{N_j - 1} \sum_{k \in D_j} (f_k - \mu_{f,j})^2, \tag{13}$$

where $\mu_{f,j}$ and $\sigma^2_{f,j}$ are the local mean and variance of the function calculated from the individuals in the neighborhood of the j-th individual. Thus, the robustness of the j-th individual can be estimated in the following way:

$$\text{Robustness measure 2: } f_j^R = \frac{\sigma_{f,j}}{\bar{\sigma}_{\boldsymbol{x},j}}, \tag{14}$$

where $\bar{\sigma}_{\boldsymbol{x},j}$ is the average of the standard deviation of x_i estimated in the j-th neighborhood:

$$\bar{\sigma}_{\boldsymbol{x},j} = \frac{1}{n} \sum_{i=1}^{n} \sigma_{x_i,j}. \tag{15}$$

The calculation of the mean and variance of x_i in the j-th neighborhood is similar to the calculation of the local mean and variance of the j-th individual as follows:

$$\mu_{x_i,j} = \frac{1}{N_j} \sum_{k \in D_j} x_{i,k}, \tag{16}$$

$$\sigma^2_{x_i,j} = \frac{1}{N_j - 1} \sum_{k \in D_j} (x_{i,k} - \mu_{x_i,j})^2. \tag{17}$$

Note that the individuals in the neighborhood can be seen as a small sample of a probability distribution around the concerned point, i.e., the current solution. If this probability distribution would coincide with $\phi(z)$ in equation 1, the approximation of the variance would be exact in the sense of the given "variation rule" $\phi(z)$. For example, if we assume that manufactoring tolerance of the final solution leads to a "noise" term which is normally distributed with a given standard deviation, the estimation of the robustness is exact if the sub-sample of the population represents the same normal distribution. Of course, this will not be the case in general. In other words, the sample will usually not be able to reproduce exactly the given distribution $\phi(z)$. Nevertheless, the results obtained in the simulations in the next section demonstrate that the estimations seem to be sufficient for a qualitative search for robust solutions.

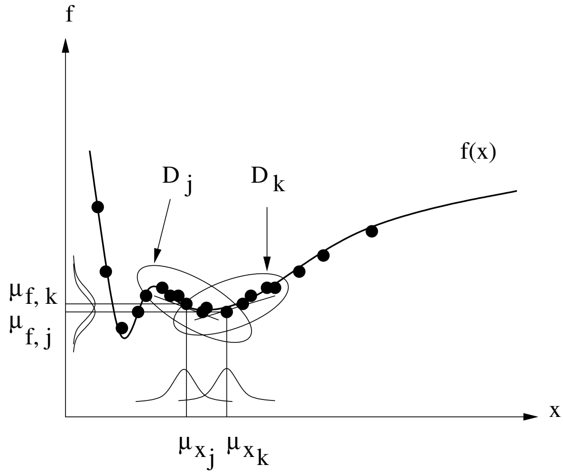


Fig. 3. Samples of the local statistics of the objective function on the basis of the current population of the evolutionary algorithm. The black dots represent the individuals in the current population

With the robustness measures defined above, it is then possible to explicitly treat the search for robust optimal solutions as a multiobjective optimization problem.

Some remarks can be made on the robustness measures defined by equations (9) and (14). The former definition is based on an approximation of the partial derivative of the function with respect to each variable. Theoretically, the smaller the neighborhood, the more exact the estimation will be. However, the estimation may fail if two individuals are too close in the design space due to numerical errors. In this method, neither the variance of the function nor the variance of the variables needs to be estimated. In contrast, the latter definition directly estimates the local variance of the variables and the function using the individuals in the neighborhood.

In this section, we have discussed possible ways to estimate a variance based second criterion for the integration of robustness in the evaluation of solutions. These criteria can now be used in a multi-objective evolutionary algorithm to visualize the trade-off between performance and robustness with the help of the Pareto front. We will employ the Dynamic Weighted Aggregation (DWA) method due to its simplicity and ease of use with evolution strategies.

4 Dynamic Weighted Aggregation for Multiobjective Optimization

4.1 Evolution Strategies

In the standard evolution strategy (ES), the mutation of the object parameters is carried out by adding an $N(0, \sigma_i^2)$ distributed random number. The standard deviations, σ_i's, usually known as the step sizes, are encoded in the genotype together with the object parameters and are subject to mutations. The standard ES can be described as follows:

$$\boldsymbol{x}(t) = \boldsymbol{x}(t-1) + \tilde{\boldsymbol{z}}, \tag{18}$$

$$\sigma_i(t) = \sigma_i(t-1)\exp(\tau' z)\exp(\tau z_i); i = 1, ..., n, \tag{19}$$

where $\boldsymbol{x}$ is an n-dimensional parameter vector to be optimized, $\tilde{\mathbf{z}}$ is an n-dimensional random number vector with $\tilde{\boldsymbol{z}} \sim N(\mathbf{0}, \boldsymbol{\sigma}(t)^2)$, z and z_i are normally distributed random numbers with $z, z_i \sim N(0,1)$. Parameters τ, τ' and σ_i are the strategy parameters, where σ_i is mutated as in equation (19) and τ, τ' are constants as follows:

$$\tau = \left(\sqrt{2\sqrt{n}}\right)^{-1}; \ \tau' = \left(\sqrt{2n}\right)^{-1} \tag{20}$$

4.2 Dynamic Weighted Aggregation

The classical approach to multiobjective optimization using weighted aggregation of objectives has often been criticized. However, it has been shown [6,7] through a number of test functions as well as several real-world applications that the shortcomings of the weighted aggregation method can be addressed by changing the weights dynamically during optimization using evolutionary algorithms. Two methods for changing the weights have been proposed. The first method is to change the weights gradually from generation to generation. For a bi-objective problem, an example for the periodical gradual weight change is illustrated in Fig. 4(a). The first period of the function can be described by:

$$w_1(t) = \begin{cases} \frac{t}{T}, & 0 \le t \le T, \\ -\frac{t}{T} + 2, & T \le t \le 2T. \end{cases} \tag{21}$$

$$w_2(t) = 1 - w_1(t), \tag{22}$$

where T is a constant that controls the speed of the weight change.

A special case of the gradual weight change method described above is to switch the weights between 0 and 1, which has been termed the bang-bang weighted aggregation (BWA) method, as shown in Fig. 4(b). The BWA has shown to be very effective in approximating concave Pareto fronts [7]. A combination of the two methods will also be very practical, as shown in Fig. 4(c).

5 Simulation Studies

5.1 Test Problem 1

The first test problem is constructed in such a way that it exhibits a clear trade-off between the performance and robustness. The function can be described as follows, which is illustrated in Fig. 5.

$$f(x) = 2.0 \sin(10 \exp(-0.08x)x) \exp(-0.25x), \tag{23}$$

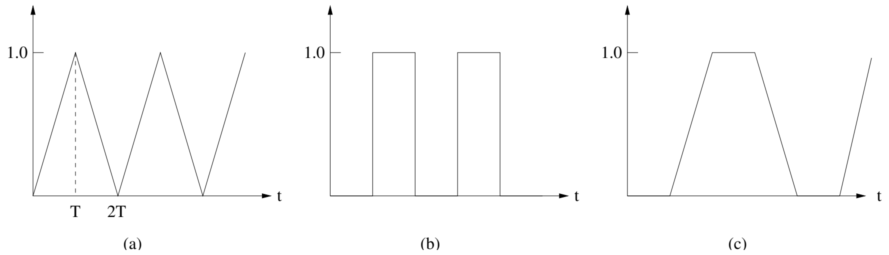


Fig. 4. Patterns of dynamic weight change. (a) Gradual change; (b) Bang-bang switching; (c) Combined

where $0 \leq x \leq 10$. From Fig. 5, it is seen that there is one global minimum together with six local minima in the feasible region. Furthermore, the higher the performance of a minimum, the less robust it is. That is, there is a trade-off between the performance and robustness and the Pareto front should consist of seven separated points.

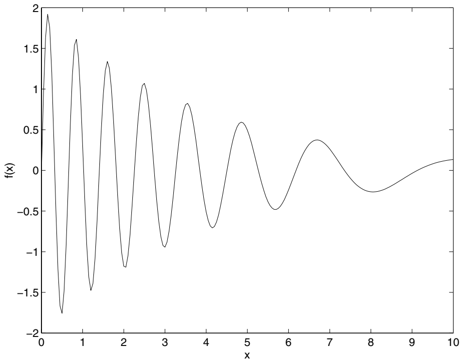
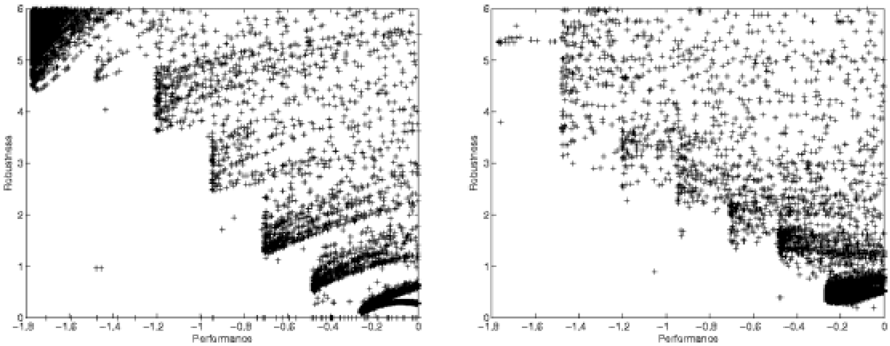


Fig. 5. The one-dimensional function of test problem 2

At first, robustness measure 1 defined by equation (9) is used. That is to say, the individuals in the neighborhood are used to estimate the partial derivatives. The obtained Pareto front is given in Fig. 6(a). It can be seen that an obvious trade-off between the performance and the robustness of the minima has been correctly reflected. Thus, it is straightforward for a user to make a choice among the trade-off solutions according to the problem at hand.

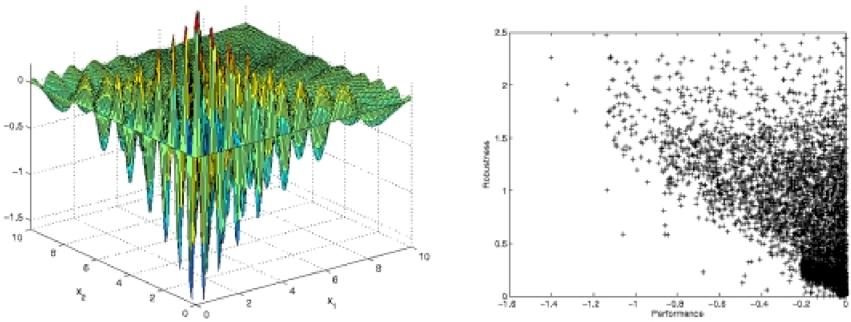
The result using the robustness measure defined by equation (14) is presented in Fig. 6(b). The Pareto fronts of both Figures 6(a) and 6(b) are qualitatively the same and the robustness values at the corners of the Pareto fronts share very similar values even quantitatively.



(a) (b)

Fig. 6. The trade-off between performance and robustness of test problem 1 based on (a) robustness measure 1, eq. (9) and (b) robustness measure 2, eq. (14)

In the following, we extend the test function in equation (23) to a two-dimensional one. The two-dimensional test function is shown in Fig. 7(a). It can be seen that there are a large number of minima with a different degree of robustness.



(a) (b)

Fig. 7. (a) The 2-dimensional function of the test problem 1. (b) The Pareto front obtained using robust measure 2, eq. (14)

The trade-off between performance and robustness is shown in Fig. 7(b) using the robustness measure 2. It can be seen that the Pareto front seems to be continuous due to the large number of minima and the small robustness difference between the neighboring minima. Nevertheless, the result provides a qualitative picture about the trade-off between performance and robustness, from which a user can make a decision and choose a preferred solution.

5.2 Test Problem 2

The second test problem is taken from reference [18]. The original objective function to minimize is as follows:

$$f(\boldsymbol{x}) = (x_1 - 4.0)^3 + (x_1 - 3.0)^4 + (x_2 - 5.0)^2 + 10.0, \tag{24}$$

subject to

$$g(\boldsymbol{x}) = -x_1 - x_2 + 6.45 \leq 0, \tag{25}$$
$$1 \leq x_1 \leq 10, \tag{26}$$
$$1 \leq x_2 \leq 10. \tag{27}$$

The standard deviation of the function can be derived as follows, assuming the standard deviation of x_1 and x_2 are the same:

$$\sigma_f(\boldsymbol{x}) = \sigma_x \sqrt{(3.0(x_1 - 4.0)^2 + 4.0(x_1 - 3.0)^3)^2 + (2.0(x_2 - 5.0))^2}, \tag{28}$$

where σ_x is the standard deviation of both x_1 and x_2, which is set to:

$$\sigma_x = \frac{1}{3}\Delta x, \tag{29}$$

where Δx is the maximal variation of x_1 and x_2. According to [18], the search for robust optimal solutions can be formulated as follows, assuming the maximal deviation of both variables is 1:

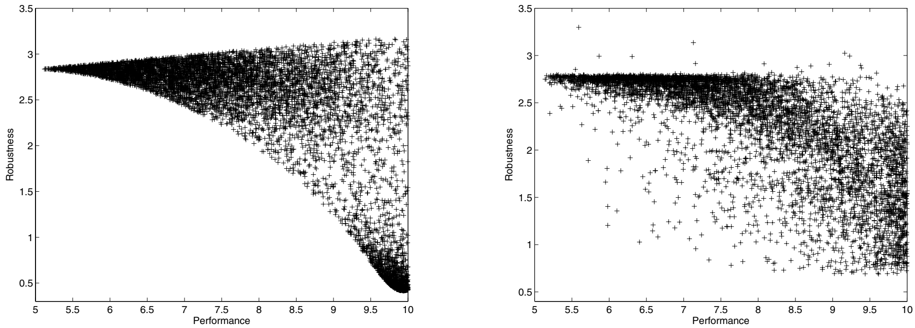
$$\text{minimize} \quad f_1 = f, \tag{30}$$
$$f_2 = \sigma_f, \tag{31}$$
$$\text{subject to} \quad g(\boldsymbol{x}) = -x_1 - x_2 + 8.45, \tag{32}$$
$$2 \leq x_1 \leq 9, \tag{33}$$
$$2 \leq x_2 \leq 9. \tag{34}$$

We call the objective for robustness in equation (31) the theoretical robustness measure, which is explicitly derived from the original fitness function.

The dynamic weighted aggregation method with a $(15, 100)$-ES is used to solve the multiobjective optimization problem. The obtained Pareto front is shown in Fig. 8(a), which is obviously concave. Note, that no archive of the non-dominated solutions has been used in the optimization, which also indicates that the success of the dynamic weighted aggregation method for multiobjective optimization has nothing to do with the archive that has been used in [6,7].

An estimated local standard deviation is used as the robustness measure so that the obtained Pareto front is comparable to the one in Fig. 8(a).

The optimization result is provided in Fig. 8(b). It is seen that although the Pareto front is quite "noisy", it does provide a qualitative approximation of the theoretical trade-off between performance and robustness.



(a) (b)

Fig. 8. (a) The Pareto front of test problem 2 using the theoretical robustness measure. (b) The approximated Pareto front using the estimated standard deviation as the robustness measure

6 Conclusion and Discussions

In this paper, we discussed the important property of robustness of solutions in the light of multiobjective optimization. Robustness measures were used as additional objectives together with the usual performance measure. With the help of the obtained Pareto fronts the relation between robustness and performance becomes visible and the decision by the application engineer for a particular solution will be more grounded.

In order to minimize the computational cost involved in estimating the robustness of solutions, we suggested two robustness measures based on the "local" variance of the evaluation function have been introduced. For both methods only information available within the current population has been used, thus, no additional fitness evaluations were necessary. In the case of computationally expensive evaluation functions this property can be essential. The basic idea is to define a neighborhood of a solution and thus to estimate the local mean and variance of a solution. The methods have been applied to two test problems and encouraging results have been obtained.

Although the proposed method is originally targeted at achieving trade-off optimal solutions between performance and robustness, it is straightforward to imagine that the method can also be used in detecting multiple optima of multimodal functions [16]. To show this capability, we consider the central two peak trap function studied in [19]. The function is modified to be a minimization problem and rescaled as shown in Fig. 9(a)

$$f(x) = \begin{cases} -0.16x & \text{if } x < 10, \\ -0.4(20-x) & \text{if } x > 15, \\ -0.32(15-x) & \text{otherwise.} \end{cases} \tag{35}$$

The function has two minima and is believed to be deceptive because values of x between 0 and 15 lead toward the local minima.

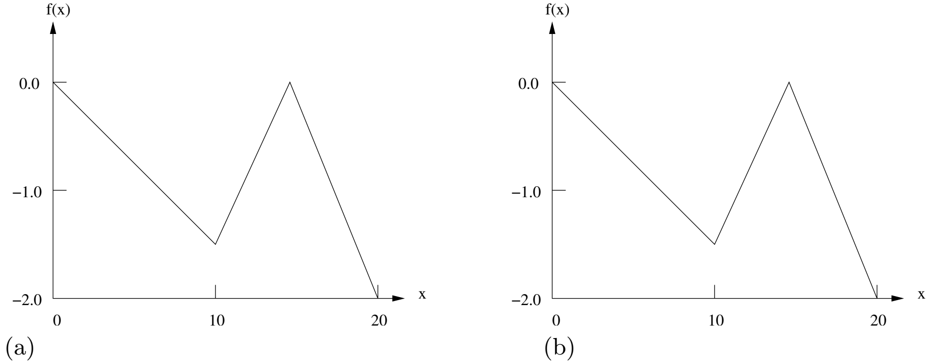


Fig. 9. (a) The trap function. (b) The detected minima

The proposed method is employed to detect the two minima of the function and the result is shown in Fig. 9(b). It can be seen that both minima have successfully been detected.

Of course, it will be difficult to distinguish different optima using the proposed method either if the function values of the optima are very similar or if the robustness values of the optima are very similar.

Several issues still deserve further research efforts. For example, how to improve the quality of the robustness estimation. Currently, the robustness estimation is quite noisy, which to some extent, degrades the performance of the algorithms. Meanwhile, it may be desirable to use the information not only in the current generation, but also from previous generations. Finally, the current algorithm is based on evolution strategies. It will be interesting to extend the method to genetic algorithms.

Acknowledgments. The authors would like to thank E. Körner for his support and M. Olhofer for discussions on robustness.

References

1. I. Das. Robustness optimization for constrained nonlinear programming problems. *Engineering Optimization*, 32(5):585–618, 2000.
2. H.-G. Beyer, M. Olhofer, and B. Sendhoff. On the behavior of $(\mu/\mu_i, \lambda)$-es optimizing functions disturbed by generalized noises. In *Foundations of Genetic Algorithms*, 2002.
3. S. Tustsui and A. Ghosh. Genetic algorithms with a robust solution searching scheme. *IEEE Transactions on Evolutionary Computation*, 1(3):201–208, 1997.
4. W. Chen, J. K. Allen, K.-L. Tsui, and F. Mistree. A procedure for robust design: Minimizing variations caused by noise factors and control factors. *ASME Journal of Mechanical Design*, 118:478–485, 1996.
5. Y. Jin, M. Olhofer, and B. Sendhoff. Managing approximate models in evolutionary aerodynamic design optimization. In *IEEE Congress on Evolutionary Computation*, volume 1, pages 592–599, Seoul, South Korea, 2001.

6. Y. Jin, T. Okabe, and B. Sendhoff. Adapting weighted aggregation for multi-objective evolution strategies. In *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 96–110, Zurich, March 2001. Springer.
7. Y. Jin, M. Olhofer, and B. Sendhoff. Evolutionary dynamic weighted aggregation for multiobjective optimization: Why does it work and how? In *Genetic and Evolutionary Computation Conference*, pages 1042–1049, San Francisco, CA, 2001.
8. A.V. Sebald and D.B. Fogel. Design of fault tolerant neural networks for pattern classification. In D.B. Fogel and W. Atmar, editors, *Proceedings of 1st Annual Conference on Evolutionary Programming*, pages 90–99, 1992.
9. H. Greiner. Robust optical coating design with evolution strategies. *Applied Optics*, 35(28):5477–5483, 1996.
10. D. Wiesmann, U. Hammel, and T. Bäck. Robust design of multilayer optical coatings by means of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 2(4):162–167, 1998.
11. A. Thompson. On the automatic design of robust electronics through artificial evolution. In *Proceedings on 2nd International Conference on Evolvable Systems*, pages 13–24. Springer, 1998.
12. J. Branke. Creating robust solutions be means of evolutionary algorithms. In *Parallel Problem Solving from Nature*, volume V of *Lecture Notes in Computer Science*, pages 119–128. Springer, Berlin, 1998.
13. J. Branke, C. Schmidt, and H. Schmeck. Efficient fitness estimation in noisy environment. In *Proceedings of Genetic and Evolutionary Computation Conference*, pages 243–250, San Francisco, 2001. Morgan Kaufmann.
14. Peter Stagge. Averaging efficiently in the presence of noise. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN V*, pages 188–197. Springer Verlag, 1998.
15. Y. Yamaguchi and T. Arima. Multiobjective optimization for transonic compressor stator blade. In *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2000.
16. D.E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In J.J. Grefenstette, editor, *Proceedings of the 2nd International Conference on Genetic Algorithms*, pages 41–49, 1987.
17. K.-L. Tsui. An overview of Taguchi method and newly developed statistical methods for robust design. *IIE Transactions*, 24(5):44–57, 1992.
18. W. Chen, M. Wiecek, and J. Zhang. Quality utility: A compromise programming approach to robust design. *Journal of Mechanical Design*, 121(2):179–187, 1999.
19. D. Beasley, D.B. Bull, and R.R. Martin. A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1(2):101–125, 1993.