

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

Yaochu Jin and Bernhard Sendhoff

Future Technology Research
Honda R&D Europe (D) GmbH
63073 Offenbach/Main, Germany
yaochu.jin@de.hrdeu.com

Abstract. In real-world applications, it is often desired that a solution is not only of high performance, but also of strong robustness. In evolutionary optimization, robust optimal solutions can be obtained either by averaging the fitness in the neighborhood or by perturbing the design variables in fitness evaluations. Unfortunately, only one solution can usually be obtained from one run of optimization using the existing methods for searching robust solutions. Besides, the user knows little about the performance degradation due to the improvement of robustness of a solution using these methods. This paper suggests 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. Thus, a trade-off between optimality and robustness can be realized with the help of evolutionary multiobjective optimization. Simulation studies have been conducted to verify the proposed method. Finally, the possibility of using this method for detecting multiple optima of multimodal functions is briefly discussed.

1 Introduction

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.

Usually, two measures can be used to enhance the robustness of an optimal solution [1, 2].

- To optimize the expectation of the objective function in a neighborhood of a target point. The expectation of the function is also known as the effective evaluation function in [3].

- To minimize the second order moment (variance) or higher order moments of the objective function.

Unfortunately, using either of the two robustness measures only cannot guarantee that a robust optimal solution can be found. If optimization of expectation is chosen as the only objective, it can happen that positive and negative deviations of the function cancel each other in the neighborhood of a target point. Thus, an undesirable solution will be considered to be robust. On the other hand, if optimization of the variance is chosen as the only objective, the optimizer may find a plateau of the original objective function, which may not be optimal. Therefore, it is sometimes insufficient to optimize the expectation or to minimize the variance only in the search of robust optimal solutions. In other words, the search of robust optimal solutions should be basically addressed as a trade-off between optimality and robustness. As suggested in [1], the trade-off can be realized by simultaneously

- optimizing the expectation and optimizing the original function, for example in [1];
- optimizing the expectation and minimizing the variance, for example in [4];
- optimizing the original function and minimizing the variance. The approach adopted in this work can be categorized into this class, however, instead of directly minimizing the variance of the function, the variance of the design variables are also taken into account.

In evolutionary optimization, main efforts have been made to obtain optimal solutions that are insensitive to small changes in the design variables. Evolutionary approaches to the search of robust optimal solutions are largely based on the optimization of the expectation, i.e., the average of the objective function. Existing methods for the expectation-based search of robust solutions in evolutionary optimization can generally be divided into two categories.

- Averaging. To find a robust solution, the fitness of a solution (\mathbf{x}) is calculated by averaging several points in its neighborhood [5–8]:

$$f(\mathbf{a}, \mathbf{x}) = \frac{\sum_{i=1}^N w_i f(\mathbf{a}, \mathbf{x} + \Delta \mathbf{x}_i)}{\sum_{i=1}^N w_i}, \quad (1)$$

where \mathbf{a} and \mathbf{x} denote a vector of environmental parameters and a vector of design variables, respectively, $i = 1, 2, \dots, N$ is the number of points to be evaluated. Usually, N should be larger than 2 but not too large due to the increasing computational cost with the increasing number of evaluations. $\Delta \mathbf{x}_i$ 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.

Several ideas have been proposed to use the information in the current or previous populations [9] 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 the points in the neighborhood using the historical data [10].

- Perturbation. A more efficient method for the search of robust solutions is to introduce stochastic perturbations in the design variables during fitness evaluation [3]:

$$f(\mathbf{a}, \mathbf{x}(t)) = f(\mathbf{a}, \mathbf{x}(t) + \Delta\mathbf{x}(t)), \quad (2)$$

where t is the generation index and $\Delta\mathbf{x}(t)$ is the random perturbation added to the design variables in generation t . Recall that the perturbations are introduced only in fitness evaluations and they have no direct influence on the genotype. Usually, $\Delta\mathbf{x}(t)$ need to be regenerated randomly in each generation. This method for obtaining robust solutions is justified due to the fact that the expected returned fitness value on each point is equivalent to the mean value over its neighborhood when the variables of the fitness function are perturbed randomly in each fitness evaluation. It has been shown under uncertain assumptions in [3] that if the population size is infinitely large, the perturbed evaluation on the fitness function $f(\mathbf{a}, \mathbf{x})$ is equivalent to the evaluation on its expected value, which is known as the effective evaluation function of $f(\mathbf{a}, \mathbf{x})$.

It is worth mentioning that if the number of points to be evaluated (N) is set to 1 and if the small variation is chosen randomly in each generation, then the averaging method is equivalent to the perturbation method. Thus, from the implementation point of view, the perturbation method can be treated as a special case of the averaging method. Nevertheless, the perturbation method and the averaging method were originated from quite a different background, and the perturbation method is computationally more efficient.

A general drawback of the aforementioned methods for the search of robust solutions is that only one objective has been used. On the one hand, this method may fail to find a robust solution in some cases as previously discussed. On the other hand, it may be impossible to find a single, ideal solution that is both optimal and robust. For a robust solution obtained using the above methods, it is difficult to know how robust it is and how much performance has been sacrificed for the increase in robustness. In real-world applications, a trade-off between the optimality and robustness often occurs. In this case, it is more desirable 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 has been suggested in [3] for detecting multiple robust solutions using the sharing method suggested in [11]. 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.

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.

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 from the multiobjective point of view in [12]. A separate fitness function that averages two points around the design point is used:

$$f^R(\mathbf{a}, \mathbf{x}) = f(\mathbf{a} + \Delta\mathbf{a}, \mathbf{x}) + f(\mathbf{a} - \Delta\mathbf{a}, \mathbf{x}), \quad (3)$$

where $f^R(\cdot)$ is a measure for the robustness of the solutions against the change of environmental parameters, $\Delta\mathbf{a}$ is a deterministic vector set up according to the robustness requirement. Obviously, the resulting optimal solution may not be robust to the variations of the design variables.

This paper considers the search of robust solutions as a multiobjective optimization problem. To this end, a measure for robustness based on the variance of the objective function in the presence of noises has been introduced, which is used as a separate objective in evolutionary optimization. To the best of our knowledge, no work has been reported in evolutionary optimization that uses a variance-based robustness measure to deal with the search of robust solutions as a multiobjective problem. More importantly, the paper suggests two methods for estimating the robustness measure using the information in the current population solely. Thus, no additional evaluations of the fitness function are necessary. To achieve the trade-off solutions, the evolutionary dynamic weighted method suggested in [13, 14] will be employed.

The remainder of the paper is organized as follows. Section 2 introduces a measure for robustness of optimal solutions. Two methods for estimating the robustness measure using individuals in the current population are suggested. The robustness measure is then applied in multiobjective optimization in Section 3 to generate a set of Pareto-optimal solutions. Two test functions have been used to show 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.

2 Measures for Robustness

The search for robust optimal solutions has been widely investigated in the field of engineering design [15]. Consider the following unconstrained minimization problem:

$$\text{minimize } f = f(\mathbf{a}, \mathbf{x}), \quad (4)$$

where \mathbf{a} and \mathbf{x} are vectors of environmental parameters and design variables. If the robustness with respect to both the environmental parameters and the design variables are to be considered, then these two different types of robustness problems can be reduced to one:

$$f = f(\mathbf{x}'), \quad (5)$$

where $\mathbf{x}' = [\mathbf{a}^T, \mathbf{x}^T]^T$ is a vector 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 \mathbf{x} .

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

$$f = f(\mu_{x_1}, \mu_{x_2}, \dots, \mu_{x_n}) + \sum_{i=1}^n \left[\frac{\partial f}{\partial x_i}(\mu_{x_1}, \mu_{x_2}, \dots, \mu_{x_n}) \right] \cdot (x_i - \mu_{x_i}), \quad (6)$$

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

$$\sigma_f^2 = \sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} \right)^2 \sigma_{x_i}^2 + \sum_{i \neq j} \sum_{j=1}^n \left(\frac{\partial f}{\partial x_i} \right) \left(\frac{\partial f}{\partial x_j} \right) \sigma_{x_i x_j}, \quad (7)$$

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 should be evaluated using the mean value of the variables. If the design variables are independent of each other, the resulting approximated variance is

$$\sigma_f^2 = \sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} \right)^2 \sigma_{x_i}^2. \quad (8)$$

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

$$f^R = \frac{1}{n} \sum_{i=1}^n \frac{\sigma_f}{\sigma_{x_i}}. \quad (9)$$

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 of 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 (8) in a local sense. Take a one-dimensional function $f(x)$ for example, as shown in Fig. 1. 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. 1. 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.

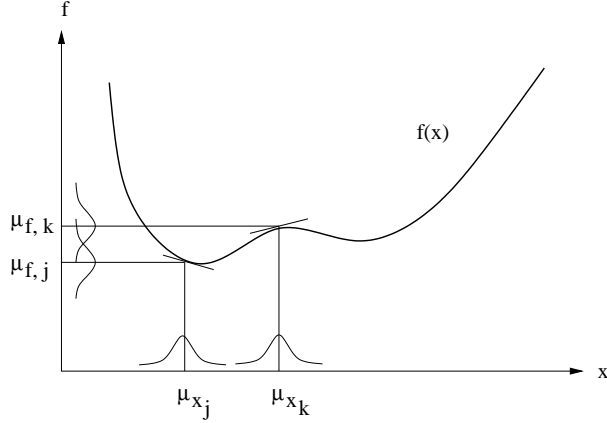


Fig. 1. Illustration of error propagation under local variations in the design variables.

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 in the neighborhood of the j -th individual. Thus, the robustness of the j -th individual can be approximated by

$$\text{Robust measure 1: } f_j^R = \frac{1}{n} \sum_{i=1}^n \frac{\bar{\sigma}_{f,j}}{\sigma_{x_i}}, \quad (10)$$

where $\bar{\sigma}_{f,j}$ is an estimation of the variance of the j -th individual according to equation (8):

$$\bar{\sigma}_{f,j}^2 = \frac{1}{N_j} \sum_{k \in D_j} \left(\frac{f_j - f_k}{x_{i,j} - x_{i,k}} \right)^2 \sigma_{x_i}^2, k \neq j, \quad (11)$$

where $x_{i,j}$ and $x_{i,k}$ denote the i -th element of \mathbf{x} of the j -th and k -th individuals, and D_j denotes a set of the individuals that are 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 $\mathbf{x}_k, k = 1, 2, \dots, \lambda$ and the j -th individual \mathbf{x}_j :

$$D_j : k \in D_j, \text{ if } d_{jk} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{i,j} - x_{i,k})^2} \leq d, 1 \leq k \leq \lambda, \quad (12)$$

where , $k = 1, 2, \dots, \lambda$ 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:

$$\mu_{f,j} = \frac{1}{N_j} \sum_{k \in D_j} f_k, \quad (13)$$

$$\sigma_{f,j}^2 = \frac{1}{N_j - 1} \sum_{k \in D_j} (f_k - \mu_{f,j})^2, \quad (14)$$

where $\mu_{f,j}$ and $\sigma_{f,j}^2$ 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:

$$\text{Robustness measure 2: } f_j^R = \frac{\sigma_{f,j}}{\bar{\sigma}_{\mathbf{x},j}}, \quad (15)$$

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

$$\bar{\sigma}_{\mathbf{x},j} = \frac{1}{n} \sum_{i=1}^n \sigma_{x_i,j}. \quad (16)$$

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:

$$\mu_{x_i,j} = \frac{1}{N_j} \sum_{k \in D_j} x_{i,k}, \quad (17)$$

$$\sigma_{x_i,j}^2 = \frac{1}{N_j} \sum_{k \in D_j} (x_{i,k} - \mu_{x_i,j})^2. \quad (18)$$

Note that the individuals in the neighborhood can be seen as a small and most probably biased sample of the local distribution around the concerned point. The sample is biased in the sense that it is chosen from the individuals in the current population of the evolutionary algorithm rather than drawn randomly from the local distribution, refer to Fig. 2. Therefore, the estimation of the local mean and variance is a rough approximation of the true local distribution. Nevertheless, the results obtained in the simulation show that the estimations are sufficient for the search of robust solutions.

With the robustness measures defined above, it is then possible to explicitly treat the search of robust optimal solutions as a multiobjective optimization problem. The main advantage of the proposed multiobjective approach to the search of robust optimal solutions over the existing ones is that the user is able to made a choice among a set of solutions and select those that can best deal with the problem at hand.

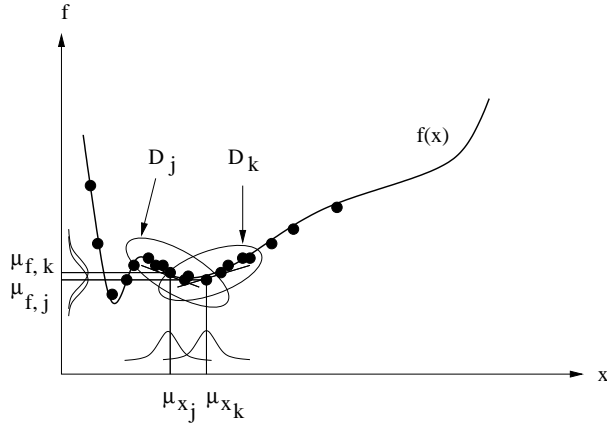


Fig. 2. Samples of the local statistics of the objective function on the basis of the current population of the evolutionary algorithm. The black dots represent the individuals in the current population.

Some remarks can be made on the robustness measure defined by equation (10) and equation (15). 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.

3 Dynamic Weighted Aggregation for Multiobjective Optimization

3.1 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:

$$\mathbf{x}(t) = \mathbf{x}(t-1) + \tilde{\mathbf{z}}, \quad (19)$$

$$\sigma_i(t) = \sigma_i(t-1) \exp(\tau' z) \exp(\tau z_i); i = 1, \dots, n, \quad (20)$$

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

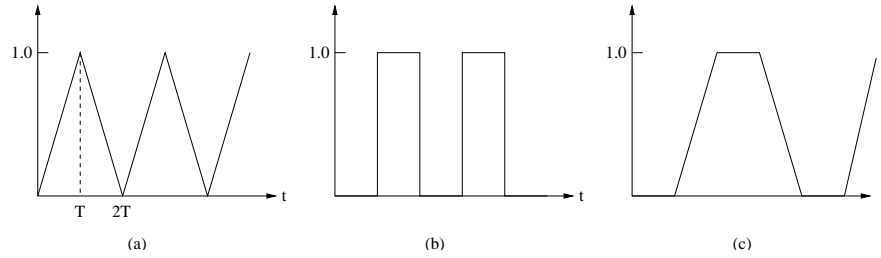


Fig. 3. Patterns of dynamic weight change. (a) Gradual change; (b) Bang-bang switching; (c) Combined.

the strategy parameters, where σ_i is mutated as in equation (20) and τ, τ' are constants as follows:

$$\tau = \left(\sqrt{2\sqrt{n}} \right)^{-1} ; \tau' = \left(\sqrt{2n} \right)^{-1} \quad (21)$$

3.2 Dynamic Weighted Aggregation

The classical approach to multiobjective optimization using weighted aggregation of objectives has often been criticized. However, it has been shown [13, 14] 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. 3(a). The first period of the function can be described by:

$$w_1(t) = \begin{cases} \frac{t}{T}, & 0 \leq t \leq T, \\ -\frac{t}{T} + 2, & T \leq t \leq 2T. \end{cases} \quad (22)$$

$$w_2(t) = 1 - w_1(t), \quad (23)$$

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. 3(b). The BWA has shown to be very effective in approximating concave Pareto fronts [14]. A combination of the two methods will also be very practical, as shown in Fig. 3(c).

4 Simulation Studies

4.1 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 exists. The function can be

described as follows, which is illustrated in Fig. 4.

$$f(x) = 2.0 \sin(10 \exp(-0.08x)x) \exp(-0.25x), \quad (24)$$

where $0 \leq x \leq 10$. From Fig. 4, 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.

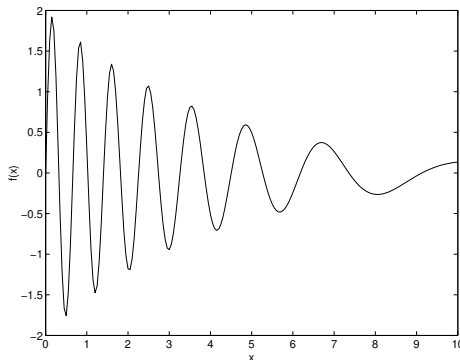


Fig. 4. The one-dimensional function of test problem 2.

At first, robustness measure 1 in equation (10) 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. 5. 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 in equation (15) is presented in Fig. 6. Although the absolute values of the robustness are different, the shape of the Pareto front is similar to the one obtained using robustness measure 1.

In the following, we extend the test function in equation (24) into a two-dimensional one. The 2-dimensional test function is shown in Fig. 7(a). It can be seen that a large number of minima with a different degree of robustness exist.

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

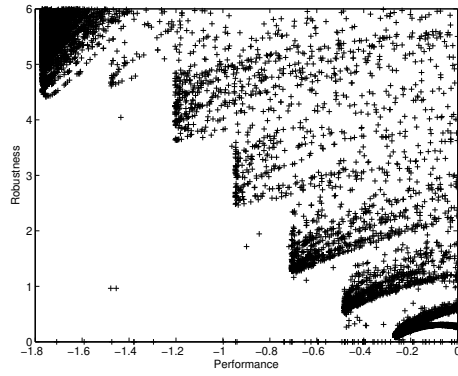


Fig. 5. The trade-off between performance and robustness of test problem 1 based on robustness measure 1.

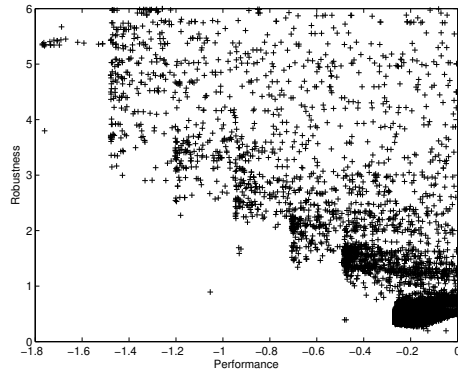


Fig. 6. The trade-off between performance and robustness of test problem 1 based on robustness measure 2.

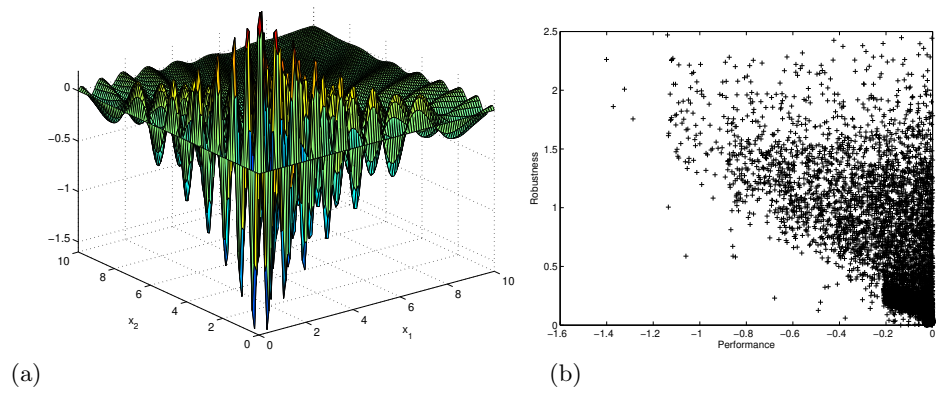


Fig. 7. (a) The 2-dimensional function of the test problem 1. (b) The Pareto front obtained using robust measure 2.

4.2 Test Problem 2

The second test problem is taken from reference [16]. The original objective function to minimize is as follows:

$$f(\mathbf{x}) = (x_1 - 4.0)^3 + (x_1 - 3.0)^4 + (x_2 - 5.0)^2 + 10.0, \quad (25)$$

subject to

$$g(\mathbf{x}) = -x_1 - x_2 + 6.45 \leq 0, \quad (26)$$

$$1 \leq x_1 \leq 10, \quad (27)$$

$$1 \leq x_2 \leq 10. \quad (28)$$

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

$$\sigma_f(\mathbf{x}) = \sigma_x \sqrt{(3.0(x_1 - 4.0))^2 + 4.0(x_1 - 3.0)^3)^2 + (2.0(x_2 - 5.0))^2}, \quad (29)$$

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

$$\sigma_x = \frac{1}{3} \Delta x, \quad (30)$$

where Δx is the maximal variation of x_1 and x_2 . According to [16], the search of robust optimal solutions can be formulated as follows, assuming the maximal deviation of the both variables is 1:

$$\text{minimize } f_1 = \frac{f}{\mu_f^*}, \quad (31)$$

$$f_2 = \frac{\sigma_f}{\sigma_f^*}, \quad (32)$$

$$\text{subject to } g(\mathbf{x}) = -x_1 - x_2 + 8.45, \quad (33)$$

$$2 \leq x_1 \leq 9, \quad (34)$$

$$2 \leq x_2 \leq 9, \quad (35)$$

where $\mu_f^* = 5.1046$ and $\sigma_f^* = 0.4168$ are the ideal solutions for the mean and deviation of the function[4]. We call the objective for robustness in equation (32) 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 [13, 14].

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). Since only

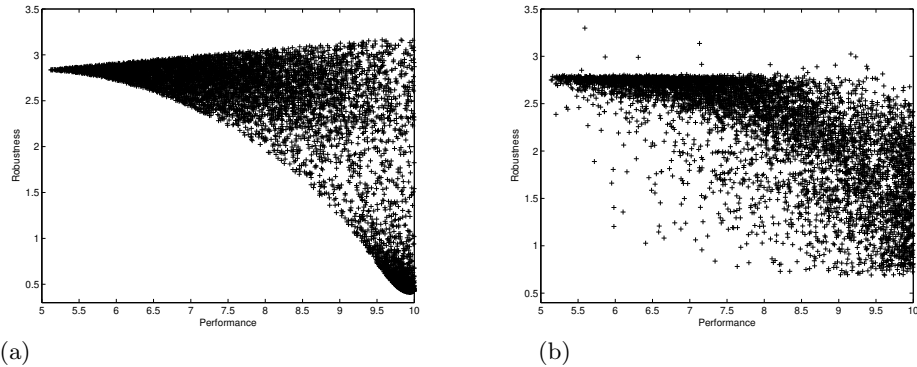


Fig. 8. (a) The Pareto front of test problem 2 using the theoretical robustness measure. (b) The approximated Pareto front using the estimated standard deviation as the robustness measure.

information in the current population is used, no additional fitness evaluations are needed, which is essential if the fitness evaluation is highly time-consuming as in many real-world applications.

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

5 Conclusion and Discussions

Two robustness measures based on the variance of function have been introduced. Methods for estimating the robustness using the information within the current population have been suggested. The robustness measure is then used as a separate objective so that the search of robust optimal solutions can be formulated as a multiobjective optimization problem to find the solutions that trade off between the performance and the robustness. The main advantage of the method is that by exploiting the available information in the current population, the robustness of a solution can be estimated without additional fitness evaluations. The basic idea is to define a neighborhood of a solution and thus to estimate the local mean and variance of a solution. With the trade-off solutions at hand, a user can easily make a decision on which solution is to be used to deal with the variations of design variables and environmental parameters in real-world applications. The method has 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 [11]. To show this capability, we consider the central two peak trap function studied in [17]. The function is modified to be a minimization

problem and rescaled as shown in Fig. 9(a)

$$f(x) = \begin{cases} -0.16x & \text{if } x < 10, \\ -0.4(20 - x) & \text{if } x > 15, \\ -0.32(15 - x) & \text{otherwise.} \end{cases} \quad (36)$$

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

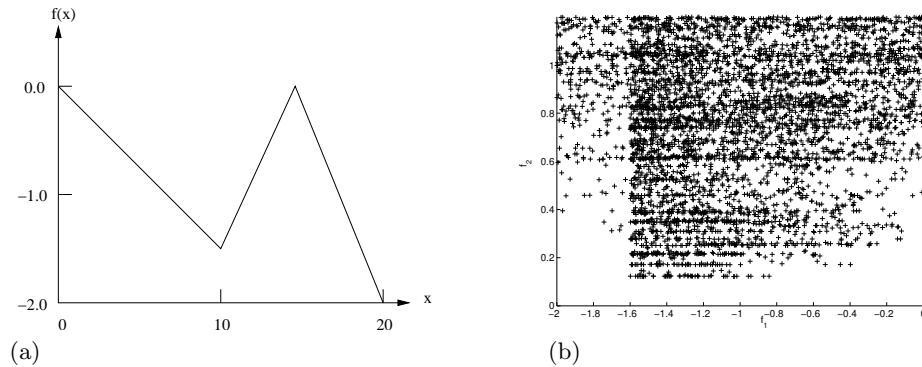


Fig. 9. (a) The trap function. (b) The detected minima.

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.

A few 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 in the 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.

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. 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.
6. H. Greiner. Robust optical coating design with evolution strategies. *Applied Optics*, 35(28):5477–5483, 1996.
7. 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.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. Y. Jin, T. Okabe, and B. Sendhoff. Adapting weighted aggregation for multiobjective evolution strategies. In *Proceedings of First International Conference on Evolutionary Multi-Criterion Optimization*, Leture Notes in Computer Science, pages 96–110, Zurich, March 2001. Springer.
14. 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.
15. K.-L. Tsui. An overview of Taguchi method and newly developed statistical methods for robust design. *IIE Transactions*, 24(5):44–57, 1992.
16. 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.
17. D. Beasley, D.B. Bull, and R.R. Martin. A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1(2):101–125, 1993.