Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 196 (2007) 3190–3218
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Robust optimization – A comprehensive survey

Hans-Georg Beyer [a,*], Bernhard Sendhoff [b]

[a] Vorarlberg University of Applied Sciences, Dornbirn, Austria
[b] Honda Research Institute Europe GmbH, Offenbach, Germany

Received 4 September 2006; received in revised form 28 February 2007; accepted 4 March 2007

Abstract

This paper reviews the state-of-the-art in robust design optimization – the search for designs and solutions which are immune with respect to production tolerances, parameter drifts during operation time, model sensitivities and others. Starting with a short glimps of Taguchi's robust design methodology, a detailed survey of approaches to robust optimization is presented. This includes a detailed discussion on how to account for design uncertainties and how to measure robustness (i.e., how to evaluate robustness). The main focus will be on the different approaches to perform robust optimization in practice including the methods of mathematical programming, deterministic nonlinear optimization, and direct search methods such as stochastic approximation and evolutionary computation. It discusses the strengths and weaknesses of the different methods, thus, providing a basis for guiding the engineer to the most appropriate techniques. It also addresses performance aspects and test scenarios for direct robust optimization techniques.

Keywords: Direct search methods; Evolutionary computation; Handling design uncertainties; Mathematical programming; Noisy optimization; Robust design; Robust optimization

1. Introduction

In order to design and manufacture high quality products at a minimum of costs, techniques are needed which are able to find those designs which meet the requirements usually specified by objectives (goal functions) at the beginning of the design process. Provided that the general system design has been fixed (e.g., the type of product and its desired basic properties are given), it is the engineer's task to choose the design parameters $\mathbf{x}$ according to an (or some) objective function(s) $f(\mathbf{x})$. These objective functions may be given by verbal descriptions, mathematical models, simulation models, or physical models. The process of finding the right design parameters is usually referred to as *optimization*. Typically, the optimization has also to account for design constraints imposed on the design parameters $\mathbf{x}$. Such constraints can be modeled by inequalities and/or equalities restricting the design space (search space). In mathematical terms a general optimization task can be stated as

$$\left.\begin{array}{lllll} \text{optimize:} & f(\mathbf{x}), & & & \text{(a)} \\ \text{subject to:} & g_i(\mathbf{x}) \leqslant 0, & i = 1,\ldots,I, & & \text{(b)} \\ & h_j(\mathbf{x}) = 0, & j = 1,\ldots,J, & & \text{(c)} \end{array}\right\} \tag{1}$$

where (1b) represents the set of inequality constraints and (1c) the set of equality constraints.

There are principle problems that might prevent us from identifying the optimum of $f(\mathbf{x})$ in (1), like NP-hardness in discrete search spaces or multi-modality in continuous search spaces. However, one might also ask whether the formulation of the optimization problem in Eq. (1) is as general and practical as it seems. The question arises whether it is desirable to locate *isolated, singular* design points with a high precision:

* Corresponding author. Tel.: +43 55727927102; fax: +43 55727929510.
E-mail addresses: Hans-Georg.Beyer@fhv.at (H.-G. Beyer), bs@honda-ri.de (B. Sendhoff).

0045-7825/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2007.03.003

(1) The global optimal design clearly depends on the goal (objective) function(s) and constraints in (1), however, these functions *always* represent models and/or approximations of the real world. As long as one does not have detailed knowledge of the error function of the model, one cannot be certain the model optimum can be mapped to the true optimum. Thus, being too precise in the model might waste resources, which could be better used at a later design stage.
(2) Even if one were able to map the model optimum to the true optimum, one might not be able to build the true optimum either because of manufacturing uncertainties [1] or because the required precision during the manufacturing stage would be too costly. There is always an economical trade-off between a potentially more complex manufacturing process and the performance gain by the new design.
(3) The formulation of the optimization problem in Eq. (1) is inherently static. Reality is dynamic: environmental parameters fluctuate (temperature, Reynolds number for gas turbine design), materials wear down, parts of a complete system might be replaced. Since the constraints on which the original design process was based change, Eq. (1) is only correct for a limited time span.
(4) Life cycle costs have to be taken into account for many engineering designs. Life cycle engineering [2,3] focuses on the whole life span of a design, e.g., easier maintenance (system design to enable a cheap disassembly and assembly process, e.g., for gas turbines), longer maintenance intervals, effect of attrition during operation, or environmentally friendly disposal, e.g., recycling capability.

Systems (1) optimized in the classical sense can be very sensitive to small changes. Changes which are likely to occur as we have just argued. A better target for a design is one that provides a high degree of *robustness*. Marczyk writes "*Optimization is actually just the opposite of robustness*" [4, p. 3]. Although there is some truth in this statement, it does make sense to re-consider the current optimization algorithm philosophy and the test functions and instances used to evaluate these in the framework of robustness. We will come back to Marczyk's statement later in the paper. As a result, again one will search for optimal solutions, however, for *robust solutions*. The procedure of finding such solutions is referred to as *robust design optimization*. The appeal of robust design optimization is that its solutions and performance results remain relatively unchanged when exposed to uncertain conditions.

The quest for robust design techniques is not only motivated by coping with sensitivities of goal or utility functions with respect to design parameters or environmental uncertainties. Whenever the design parameters describe only a part of the overall system, the engineer has to make assumptions on the optimal operating point of each subsystem. Either the integrated system must be simulated [5] or optimal (robust) operating intervals have to be defined. A similar problem arises in the field of *complex systems design* where multidisciplinary teams develop a complex system by independently optimizing subsystems [6]. Due to the complexity of the whole system and the time constraints (time-to-market restrictions), the teams must optimize their subsystems without full information about the outputs of the other subsystems (which appear as the inputs of the subsystem to be optimized). Such types of optimization are referred to as *multidisciplinary design optimization.* Coping with uncertainties in such optimizations is subject of *robust multidisciplinary design*, see e.g., [6–8], and may be regarded as a new application field of robust optimization.

Concepts of robustness and robust design optimization have been developed independently in different scientific disciplines, mainly in the fields of operations research (OR) and engineering design.

While the methods of *stochastic* (*linear*) *programming*[1] may be regarded as a first approach to deal with uncertainties treating robustness as a side effect only, the notion of *robust optimization* gained focus in OR after the publication of [10] (for an introductory paper, see also [11]).

Robust design and optimization has even deeper roots in engineering. There it is inextricably linked with the name of Taguchi [12] who initiated a highly influential design philosophy (see Section 2). Due to the advent of high-speed computers and its exponentially increasing FLOPS-rates (floating point operations per second), robust design optimization has gained increasing interest in the past few years. This is reflected in, e.g., a special issue of the ASME *Journal of Mechanical Design* (July issue 2006, vol. 128), which is devoted to robust and reliability-based design, and also in an overview article of Park et al. [13].

While [13] represents a first resume of what has been done in robust engineering design optimization so far, this survey tries to present a broader perspective taking the activities in the OR field into account. Furthermore, as indicated in [14], modeling robustness aspects is only one side of the coin, solving the related optimization problems is the other often computationally demanding side, and only a few computational tools have been developed so far. Therefore, also the different algorithmic approaches will be reviewed.

This survey is organized as follows. First, we will provide a short introduction into the "Taguchi method" of robust design. The seminal work of Taguchi marks the beginning of systematic design methods taking robustness into account. In Section 3, we will shortly review the different sources and kinds of uncertainties that can be encountered when facing design problems. Then, in order to incorporate uncertainties in design optimization, robust counterparts to the original design objectives will be defined – the robustness measures. Section 4 gives an

[1] For an introductory book see, e.g., [9].

overview over how the robustness measures can be optimized using different approaches from mathematical programming, nonlinear optimization, and direct (randomized) search methods including evolutionary algorithms. Most often robust design optimization will rely on simulation programs evaluating the different designs. Moreover, the outcome of such simulations is often noisy. In such cases direct search methods might be the means of choice. However, it is not clear which of the various direct search algorithms are the most efficient ones. To this end, it would be helpful to have a collection of *scalable* test function in order to compare the different algorithms. Section 5 makes a first attempt to propose such a test bed. Being based on the results of this review, the concluding Section 6 aims to identify promising future research areas.

2. Taguchi's robust design methodology

Early attempts to account for design uncertainties in the framework of quality engineering are closely connected with Taguchi, the "father of robust design" who envisioned a three-stage design methodology comprising [15]:

(1) *Systems design*: determines the basic performance parameters of the product and its general structure.
(2) *Parameter design*: optimizes the design parameters in order to meet the quality requirements.
(3) *Tolerance design*: fine-tuning of the design parameters obtained in the second stage.

From viewpoint of mathematical optimization, the differentiation between the second and the third stage seems superfluous since both stages differ only in the granularity by which design parameters are treated (of course practically the classification might be important because stage two and three can occur under very different constraints, e.g., design time vs. operation time). That is why we will only concentrate on the second stage.

The main difference of Taguchi's method compared to ordinary optimization lies in the accounting for performance variations due to noise factors beyond the control of the designer. That is, there are two kinds of parameters entering the objective function: *control parameters* $\mathbf{x}$, which are to be tuned to optimality, and *noise factors* $\boldsymbol{\xi}$, such as environmental conditions (e.g., temperature, pressure, etc.) and production tolerances (e.g., weight and length variations, purity of material used, etc.) difficult to be controlled by the designer.

Depending on the design objective, Taguchi proposed so-called signal-to-noise measures. Let $y_i = y(\mathbf{x}, \boldsymbol{\xi}_i)$ be the quality value of a single sample (keeping $\mathbf{x}$ constant), the *Mean Square Deviation*

$$\mathrm{MSD}_1 := \frac{1}{\kappa}\sum_{i=1}^{\kappa}(y(\mathbf{x}, \boldsymbol{\xi}_i) - \hat{y})^2 \tag{2}$$

measures the deviation of y from the desired target value $\hat{y}$. If the design goal is to have y close to zero, Eq. (2) simplifies to

$$\mathrm{MSD}_2 := \frac{1}{\kappa}\sum_{i=1}^{\kappa} y(\mathbf{x}, \boldsymbol{\xi}_i)^2, \tag{3}$$

and if the design goal is to have y as large as possible, Taguchi proposed

$$\mathrm{MSD}_3 := \frac{1}{\kappa}\sum_{i=1}^{\kappa} y(\mathbf{x}, \boldsymbol{\xi}_i)^{-2}. \tag{4}$$

Using these MSD-functions, Taguchi defined the "signal-to-noise ratio"

$$\mathrm{SNR} := -10\log_{10}(\mathrm{MSD}) \tag{5}$$

which is to be maximized w.r.t. $\mathbf{x}$.

Taguchi does not really use an automated optimization procedure in order to maximize (5). Instead he uses *design of experiments* (DOE)[2] in order to evaluate different designs (i.e., different $\mathbf{x}$ values). To this end, the design parameters are systematically changed taking values on a predefined (orthogonal) lattice, the so-called "control array" (also "inner array"). At each design point $\mathbf{x}$, the noise variables $\boldsymbol{\xi}$ are systematically changed according to an "outer array" $(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_\kappa)$. Thus, one obtains for each $\mathbf{x}$ a set $(y_1, \ldots, y_\kappa)$ which can be used to calculate $\mathrm{MSD}(\mathbf{x})$ and finally an array of $\mathrm{SNR}(\mathbf{x})$. A statistical data analysis applied to this array allows to identify the $\mathbf{x}$ producing the best performance.

An example using Taguchi's method for finding an optimal robust design of an automobile rear-view mirror has been presented in [18]. A decisive point for a successful application of this method concerns the identification of the principal design variables, the number of which must be kept small (see below).

From viewpoint of optimization efficiency, Taguchi's optimization approach suffers from the "curse of dimensions". If we consider an N-dimensional design vector $\mathbf{x}$ we need at least two points per dimension, we already have 2^N designs and thus we have to perform a minimum of $\kappa 2^N$ experiments. As pointed out by Trosset [19]: "*the Taguchi approach violates a fundamental tenet of numerical optimization – that one should avoid doing too much work until one nears a solution.*"

Besides these efficiency considerations there are other aspects of Taguchi's method which are subject to controversial debates summarized in [20] including the disputable definition of MSD and SNR functions (2)–(5). In the following, we will take a broader view on robustness, however, keeping in mind the merits of Taguchi in developing a philosophy of robust design.

[2] For an state-of-the-art introduction into DOE, the reader is referred to [16,17].

3. Robustness concepts and measures

This section discusses two topics. First, we will identify the sources of uncertainties that can be encountered in the design and optimization process. Thereafter, the second section provides measures for evaluating the effects of these uncertainties and means to incorporate them in the objective functions.

3.1. Uncertainties in the parameter design process

There are different possibilities to classify uncertainties which the engineer has to face during the design process. Firstly, we will start from a system theoretical point of view and thereafter present an alternative epistemologically motivated typology often used in mechanical engineering.

Consider the "system" in Fig. 1. This system is to be designed in such a way that it generates (desired) outputs $\mathbf{f}$ which depend on input quantities $\boldsymbol{\alpha}$ provided by the environment the system is embedded in. The output behavior of the system can be controlled to a certain extend by the design variables $\mathbf{x}$

$$\mathbf{f} = \mathbf{f}(\mathbf{x}, \boldsymbol{\alpha}) \tag{6}$$

to be tuned in such a way that designer-defined performance or quality measures are fulfilled. While this parameter design task matches perfectly the classical optimization scenario, real-world system design has to face different kinds of uncertainties which are usually beyond the (direct) control of the designer (see Fig. 1)[3]:

(A) *Changing environmental and operating conditions.* These uncertainties enter the system via the $\boldsymbol{\alpha}$-variables in Eq. (6). Examples are the angle of attack in airfoil design, operating temperature, pressure, humidity, changing material properties and drift, etc. This kind of uncertainties is also referred to as *Type I variations* in [21].

(B) *Production tolerances and actuator imprecision.* The design parameters of a product can be realized only to a certain degree of accuracy. High precision machinery is expensive, therefore, a design less sensitive to manufacturing tolerances reduces costs. This type of uncertainties enters the $\mathbf{f}$ functions in terms of perturbations $\boldsymbol{\delta}$ of the design variables $\mathbf{x}$, i.e., one has to consider the function

$$\mathbf{f} = \mathbf{f}(\mathbf{x} + \boldsymbol{\delta}, \boldsymbol{\alpha}). \tag{7}$$

This kind of uncertainties is referred to as *Type II variations* in [21]. Note, even though $\boldsymbol{\delta}$ enters $\mathbf{f}$ in (7) via an additive coupling with $\mathbf{x}$, in the general case $\boldsymbol{\delta}$ may also depend on $\mathbf{x}$. For example, $\boldsymbol{\delta} = \varepsilon\mathbf{x}$ models relative manufacturing tolerances.

[3] Note, that one is not restricted to continuous spaces, $\mathbf{x}$ can also contain discrete components.

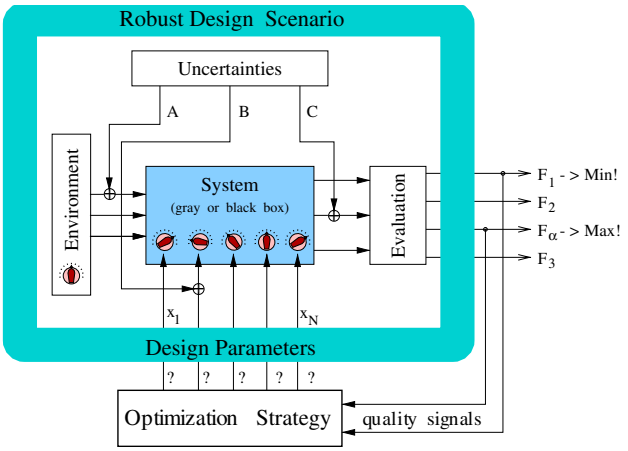

Fig. 1. Robust system design has to account for different classes of uncertainties: (A) uncertain operating conditions, (B) design parameter tolerances, (C) uncertainties concerning the observed system performance.

(C) *Uncertainties in the system output.* These uncertainties are due to imprecision in the evaluation of the system output and the system performance. This kind of uncertainty includes measuring errors and all kinds of approximation errors due to the use of models instead of the real physical objects (model errors). That is, the actually observed output/performance $\tilde{\mathbf{f}}$ is a (random) functional of $\mathbf{f}$

$$\tilde{\mathbf{f}} = \tilde{\mathbf{f}}[\mathbf{f}(\mathbf{x} + \boldsymbol{\delta}, \boldsymbol{\alpha})]. \tag{8}$$

(D) *Feasibility uncertainties.* Uncertainties concerning the fulfillment of constraints the design variables must obey. This kind of uncertainty is different to (A)–(C) in that it does not consider the uncertainty effects on $\mathbf{f}$ but on the design space. In real-world applications it often appears together with the uncertainty types (A) and (B).

Since the uncertainties of type (A) and (B) are closely related to the sensitivity of the objective function f w.r.t. specific changes, the corresponding robustness has also been termed *sensitivity robustness* [22].

There are different possibilities to quantify the uncertainties subsumed under (A)–(D) mathematically. Basically, the uncertainties can be modeled deterministically, probabilistically, or possibilistically:

(1) The *deterministic type* defines parameter domains in which the uncertainties $\boldsymbol{\alpha}$, $\boldsymbol{\delta}$, etc. can vary,
(2) the *probabilistic type* defines probability measures describing the likelihood by which a certain event occurs, and
(3) the *possibilistic type* defines fuzzy measures describing the possibility or membership grade by which a certain event can be plausible or believable.

These three different types of uncertainties are mathematically modeled usually by crisp sets, probability

distributions, and fuzzy sets [23], respectively. Having four different classes (A–D) of uncertainties and three ways to quantify them, one ends up with up to 12 different concepts of robustness that can be encountered in the analysis of real-world problems. The most important ones that are frequently found in the literature will be reviewed in more detail below.

Especially in mechanical engineering disciplines, a somewhat complementary classification scheme of uncertainties is in common use. It is taken from an epistemological perspective differentiating the uncertainties into objective and subjective ones. Objective uncertainties, also called *aleatory* [24] or *random* uncertainties, are of intrinsically irreducible stochastic nature. That is, these kinds of uncertainties are of physical nature, e.g., the noise in electrical devices (resistor noise, $1/f$-noise), seismic and wind load, humidity, temperature, server load, material parameters (steel yielding strength, stiffness, conductivity), stochastic imperfections in the design of mesostructures of materials [25] and quantum mechanical effects to name but a few. These uncertainties cannot be removed. The designer has to "live with them" and to optimize his design according to this reality. Due to the probabilistic nature, probability distributions are the adequate means for the mathematical description of these uncertainties. Thus aleatory uncertainties basically belong to item (2) (probabilistic type) uncertainties.

In contrast to the objective character of aleatory uncertainties, *epistemic* uncertainties reflect the lack of knowledge a designer has about the problem of interest. This kind of uncertainty is regarded as subjective, because it is due to a lack of information that could, in principle, be reduced by increased efforts.[4] Epistemic uncertainties include uncertainties about the model used to describe the reality, its boundary and operation conditions, also referred to as *model form errors* [26], and also the errors introduced by the numerical solution methods used (e.g., discretization error, approximation error, convergence problems). Such uncertainties can be modeled by type (1) and (3) techniques.

It should be also mentioned that the differentiation into aleatory and epistemic uncertainties does not cover all aspects of robust design: Consider the performance of an airfoil design with respect to different angles of attack, i.e., performance of a system under different operating conditions (which are well known, there is no lack of information). This is a type A(1) uncertainty in the taxonomy presented above, however, it neither belongs to the aleatory nor to the epistemic uncertainty class.

3.2. Accounting for uncertainties – robustness measures

Robust design can be regarded as an optimization approach which tries to account for uncertainties that have been defined in the previous section. There are different approaches to incorporate the uncertainties. Generally one tries to derive/construct robust counterparts F of the original performance or objective function f.[5] In the following an overview over the different robustness measures starting with the treatment of the deterministic uncertainties. Thereafter, three sections are devoted to probabilistic uncertainties and finally, a fifth section concerns possibilistic uncertainties.

3.2.1. Coping with deterministic uncertainties: the robust counterpart approach

In this section, robustness with respect to deterministically defined uncertainty sets is considered. Given an objective function $f(\mathbf{x})$ to be *minimized*, the robust counterpart function $F_B(\mathbf{x};\epsilon)$ is defined as

$$F_B(\mathbf{x};\epsilon) = \sup_{\boldsymbol{\xi}\in \boldsymbol{X}(\mathbf{x},\epsilon)} f(\boldsymbol{\xi}), \tag{9}$$

where $\boldsymbol{X}(\mathbf{x},\epsilon)$ is a neighborhood of the design $\mathbf{x}$ the size of which depends on the *regularization* parameter ϵ and

$$\lim_{\epsilon\to 0} F_B(\mathbf{x};\epsilon) = f(\mathbf{x}). \tag{10}$$

This kind of accounting for uncertainties of the design variables (i.e., type B uncertainties) presents a worst case scenario in that it considers the maximal f-value within the neighborhood of the design $\mathbf{x}$. This technique is also referred to as *robust regularization* in [27]. Minimizing the regularized f resembles basically the idea of "minimax" optimization used in context of Tschebyshev approximation. This approach can also be found in [19]. In the *evolutionary computation* field,[6] it has been used to evolve robust wing-boxes in [35]. A genetic algorithm utilizing this idea for robust job shop scheduling has been proposed in [36]. Robust regularization is not restricted to type B uncertainties, it can also be defined for type A uncertainties. In [37] it has been used to find robust solutions to least-squares problems with uncertain data.

In order to see the effect of robust regularization on a simple example, consider the one-dimensional function

$$f(x) := \begin{cases} -x, & \text{if } x < 0, \\ \sqrt{x}, & \text{if } x \geqslant 0. \end{cases} \tag{11}$$

which is displayed as solid curve in Fig. 2 and which is to be (robustly) minimized. Let us consider the design neighborhood $\xi \in [x-\epsilon, x+\epsilon]$. The robust regularization reads

$$\begin{aligned} F_B(x;\epsilon) &= \max_{\xi}\{f(\xi)|x-\epsilon \leqslant \xi \leqslant x+\epsilon\} \\ &= \max_{\xi}\{f(\xi)||x-\xi| \leqslant \epsilon\}. \end{aligned} \tag{12}$$

[4] This is of course a somewhat philosophical point of view and increasing the efforts might be economically not feasible, thus "subjectivity" can be also a consequence of fully "objective" constraints.

[5] Note, in order to simplify notations, we will restrict our considerations to scalar-valued objective functions $f(\mathbf{x})$.

[6] For introductory material to evolutionary algorithms, such as evolution strategies and genetic algorithms, the reader is referred to, e.g., [28–34].

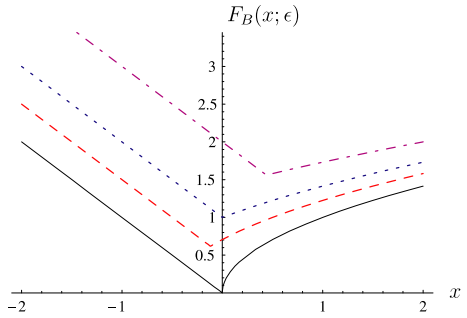

Fig. 2. On the effect of robust regularization of function (11). The regularized function depends on the regularization parameter ϵ. From bottom to top $\epsilon = 0$, 0.5, 1.0, and 2.0.

Having a look at the graph of (11) in Fig. 2 (the continuous curve), one sees that the maximum in this constraint optimization problem is attained at the interval boundaries $x - \epsilon$ and $x + \epsilon$, respectively. Shifting successively the $x \pm \epsilon$ interval from large negative x-values to large positive x-values, the maximum is first governed by the left branch, i.e., by $-(x - \epsilon)$, and then for $x \geqslant \hat{x}$ by the right branch, i.e., $\sqrt{x + \epsilon}$. The transition point $\hat{x}$ is given by equality of the left and right branch maximum, i.e.,

$$-(\hat{x} - \epsilon) = \sqrt{\hat{x} + \epsilon}. \tag{13}$$

This quadratic equation can be solved for $\hat{x}$ yielding

$$\hat{x} = \frac{1 + 2\epsilon - \sqrt{1 + 8\epsilon}}{2} \tag{14}$$

and the robust regularized f becomes

$$F_B(x;\epsilon) = \begin{cases} \epsilon - x, & \text{if } x < \hat{x}, \\ \sqrt{\epsilon + x}, & \text{if } x \geqslant \hat{x}. \end{cases} \tag{15}$$

Fig. 2 shows the graphs of $F_B(x;\epsilon)$ for different values of the regularization parameter ϵ. The optimal robust design can be easily determined. Taking into account that the monotonously decreasing part in Eq. (15) corresponds to the left branch in Fig. 2, it becomes clear that $F_B(x;\epsilon)$ takes its minimum at $x = \hat{x}$ which is given by Eq. (14). This optimal design depends on the regularization parameter ϵ displayed in Fig. 3. As one can see, the location of the robust minimum does in general *not* coincide with the minimum of f (continuous curve). This is a typical property of robust optimization.

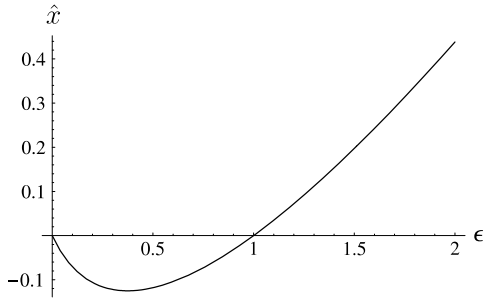

Fig. 3. The optimal value $\hat{x}$ of the design variable of the robust regularized function depends on the regularization parameter ϵ.

There is no reason to restrict robust regularization to design parameters $\mathbf{x}$ as has been done in [27,19]. It can also be defined for the $\boldsymbol{\alpha}$ parameters in (6), thus providing a regularization w.r.t. type A uncertainties

$$F_A(\mathbf{x};\epsilon) = \sup_{\boldsymbol{\alpha} \in \mathbf{A}(\epsilon)} f(\mathbf{x}, \boldsymbol{\alpha}), \tag{16}$$

where $A(\epsilon)$ is a neighborhood that defines the operating conditions of the system.

Combinations of type A and type B regularizations are also imaginable.

The regularization w.r.t. feasibility constraints, i.e., type D uncertainties, has been considered in the context of *robust linear, conic, and quadratic programming* [38,39] under the label "*robust counterpart approach*". This approach usually incorporates also type A uncertainties. In the case of linear constrained minimization

$$\min_{\mathbf{x}} \{\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{x} | \mathbf{A}\mathbf{x} - \mathbf{b} \geqslant \mathbf{0}\}, \tag{17}$$

$\boldsymbol{\alpha}$ as well as the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ can be sources of uncertainties, i.e., they are only known to belong to a "*uncertainty set*" $\mathscr{U}$. Regarding the constraints $\mathbf{A}\mathbf{x} - \mathbf{b} \geqslant \mathbf{0}$ as hard constraints, only *robust feasible candidate solutions* $\mathbf{x}$ are allowed to be considered. That is, the $\mathbf{x}$ must fulfill *all* possible realizations of $\mathscr{U}$. Thus, the regularized counterpart of f becomes

$$F_D(\mathbf{x}) := \sup_{(\boldsymbol{\alpha},\mathbf{A},\mathbf{b}) \in \mathscr{U}} \{\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{x} | \mathbf{A}\mathbf{x} - \mathbf{b} \geqslant \mathbf{0}\}. \tag{18}$$

Since maximizing $\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{x}$ is equivalent to finding the smallest t for which $\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{x} \leqslant t$, the robust counterpart of the linear minimization problem (17) can alternatively be expressed as

$$\min_{\mathbf{x},t} \{t | \forall (\boldsymbol{\alpha}, \mathbf{A}, \mathbf{b}) \in \mathscr{U} : (t \geqslant \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{x}) \wedge (\mathbf{A}\mathbf{x} - \mathbf{b} \geqslant \mathbf{0})\}. \tag{19}$$

The inequalities $t - \mathbf{a}^{\mathrm{T}}\mathbf{x} \geqslant 0$ and $\mathbf{A}\mathbf{x} - \mathbf{b} \geqslant \mathbf{0}$ must be satisfied for all $\mathbf{a}, \mathbf{A}, \mathbf{b} \in \mathscr{U}$ (worst case scenario). This is equivalent to demanding $\min_{\mathbf{a} \in \mathscr{U}}(t - \mathbf{a}^{\mathrm{T}}\mathbf{x}) \geqslant 0$ and $\min_{\mathbf{A},\mathbf{b} \in \mathscr{U}}(\mathbf{A}\mathbf{x} - \mathbf{b}) \geqslant 0$, respectively. Thus, we alternatively obtain

$$\left.\begin{array}{ll} \min & t \\ \text{s.t.} & \min\limits_{\mathbf{a} \in \mathscr{U}}(t - \mathbf{a}^{\mathrm{T}}\mathbf{x}) \geqslant 0, \\ & \min\limits_{\mathbf{A},\mathbf{b} \in \mathscr{U}}(\mathbf{A}\mathbf{x} - \mathbf{b}) \geqslant 0, \end{array}\right\} \tag{20}$$

a linear objective function with (arbitrarily) difficult constraints due to nonlinearity of the functions produced by the minimum operator. Therefore, the robust counterpart of the linear programming task (17) is typically *not* a linear programming task. This can easily be seen considering an example where $\mathscr{U}$ contains only three different $\boldsymbol{\alpha}$ vectors. For sake of simplicity we consider a two-dimensional case. Fig. 4 shows a typical case where there is a large uncertainty about the $\boldsymbol{\alpha}$ vectors resulting in ascending and descending $f(x)$ scenarios. Taking the supremum of the

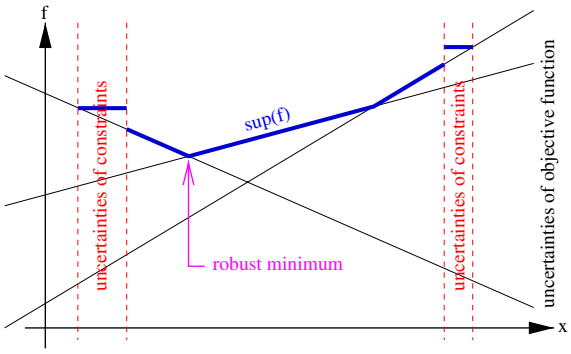

Fig. 4. Example of robust linear optimization scenario where the robust counterpart (bold line) does *not* represent a linear optimization task.

three possible $f(x)$-functions within the constraints given, we see that the resulting robust regularization function $F = \sup(f)$ is not linear within the constraints given. This has considerable consequences, both from viewpoint of solving such tasks and the general properties of the minimizer: It is well known from linear programming theory (see, e.g., [40]) that the optimum states are located on the feasibility boundary. As one can see in Fig. 4, this does not necessarily hold for the robust counterpart, the optimal solution can be located deeply in the feasible domain. This is also different to linear *stochastic programming* (see, e.g., [9]), where uncertainties are aggregated in a linear function resulting in an ordinary linear programming problem.

The field of robust mathematical programming has received increasing interest during the last five years [41–46]. While we have seen that the robust counterparts will usually turn the original linear or quadratic problem into nonlinear ones, the focus of recent research is on the question to identify conditions and uncertainty sets resulting in systems (20) which can be solved efficiently, i.e., the original problem class should be conserved or should at least not go beyond *second-order cone problems*. Of course, this is bought at the expense of a higher effort modeling/approximating the original problem in such a way that it fits into the concepts developed.

3.2.2. Expectancy measures of robustness

The robust regularization approach taken in Section 3.2.1 may be considered as a *worst case* philosophy: According to Eq. (9), the robust counterpart F of a function f to be minimized is the maximal value of f in a user-defined domain $\mathscr{X}$ around the design point $\mathbf{x}$. This can be regarded as a conservative setting, and choosing the domain $\mathscr{X}$ too large can result in robust solutions with very poor performance such that the resulting design is rendered useless (e.g., the stability of an aircraft design is guaranteed, but due to its heavy weight the aircraft cannot take off). In such situations it might be better to consider robustness measures based on probability. That is, we regard $\boldsymbol{\delta}$, $\boldsymbol{\alpha}$, and $\tilde{f}$ in (7) and (8), respectively, as random variables obeying user-provided distribution functions reflecting the knowledge about the uncertainties. As a result, the function f itself becomes a *random function*. Dealing with these random functions can be done in two ways: (a) using aggregation approaches, and (b) evaluating the induced distribution function of f. We will consider the latter case in the next section.

The *aggregation approach* yields one (or more) *integral* measure(s) of robustness – the *expectancy measures*. Introducing a *utility function* $U(f)$, the robust counterpart of f can be defined as the *conditional expectation* of $U(f)$

$$F_U(\mathbf{x}) := \mathrm{E}[U(f)|\mathbf{x}]. \tag{21}$$

Depending on the choice of U, one can define different robustness measures.[7] Using

$$U(f) = \mathrm{sign}(f)|f|^k, \tag{22}$$

one obtains *momentum measures* introduced in the context of robust evolution strategy optimization [49].[8] The special case $k = 1$ yields with (7)

$$F_1(\mathbf{x}) = \int f(\mathbf{x}+\boldsymbol{\delta},\boldsymbol{\alpha})p(\boldsymbol{\delta},\boldsymbol{\alpha})\,\mathrm{d}\boldsymbol{\delta}\,\mathrm{d}\boldsymbol{\alpha}, \tag{23}$$

where $p(\boldsymbol{\delta},\boldsymbol{\alpha})$ is the joint density of the uncertainties. Accounting for type B uncertainties only, one obtains

$$F_1(\mathbf{x},\boldsymbol{\alpha}) = \int f(\mathbf{x}+\boldsymbol{\delta},\boldsymbol{\alpha})p(\boldsymbol{\delta})\,\mathrm{d}\boldsymbol{\delta} \tag{24}$$

a standard robustness measure often proposed in literature, e.g., [19,50,51]. In [52] the quantity F_1 has been called "effective fitness" (see also [53, p. 127f]). Power coefficients $k \neq 1$ can be used to put different emphasis on the utility of f-values. If $k > 1$, extreme values of the random variate f are amplified. In the opposite case $0 < k < 1$, the influence of (statistical) outliers will be dampened.

Other utility functions than the power functions (22) can be of interest or may be additionally used to define robustness aspects in a specific application. For example, finding plateau-like regions in quality or fitness landscapes suggests the introduction of *dispersion measures* such as

$$F_d(\mathbf{x}) = \int (f(\mathbf{x}+\boldsymbol{\delta}) - f(\mathbf{x}))^2 p(\boldsymbol{\delta})\,\mathrm{d}\boldsymbol{\delta} \tag{25}$$

proposed in [50, p. 7]. Considering the *conditional variance* of f

$$\mathrm{Var}[f|\mathbf{x}] = \mathrm{E}[(f - \mathrm{E}[f|\mathbf{x}])^2|\mathbf{x}] = \mathrm{E}[f^2|\mathbf{x}] - (F_1(\mathbf{x}))^2, \tag{26}$$

where

$$\mathrm{E}[f^2|\mathbf{x}] = \int (f(\mathbf{x}+\boldsymbol{\delta}))^2 p(\boldsymbol{\delta})\,\mathrm{d}\boldsymbol{\delta} \tag{27}$$

provides an alternative to the dispersion measure (25).

[7] A similar proposal can be found in [47, p. 6f], and it turns out that this is a recurring idea. In stochastic programming it appears as a means of handling "risk in decision-making", see e.g., [48, p. 4].

[8] Note, in contrast to [49, p. 311] we have changed f^k to $\mathrm{sign}(f)|f|^k$ in order to treat the even k cases for $f(\mathbf{x})$-functions with negative f-values correctly.

The search for optimal robust designs appears often as a multiple criteria decision problem. For example, optimizing the conditional expectation (21) may lead to a design with large dispersion F_d, Eq. (25), or variance (26). This is not always desirable. For example, provided that the optimum is located in a plateau-like region of the f-function, decreasing its variance might be more important than just searching for the best mean value behavior. However, also the opposite case might be of interest: An increased "phenotypic" variance of a product can be desirable. At least in nature, additional variance in the genotype or phenotype of animals and plants makes creatures more resistant to parasites, bacteria, and viruses.[9] In all these cases there is a trade-off between maximal performance and variance and a compromise must be found. Thus, we are entering the field of *robust multi-objective optimization*.

As a simple example, consider the objective function

$$f(\mathbf{x},\alpha) = \alpha - (\alpha - 1)\|\mathbf{x}\|^2, \quad \alpha \in \mathbb{R},\ \mathbf{x} \in \mathbb{R}^N \tag{28}$$

to be minimized under a type A uncertainty, where α is assumed to be normally distributed $\alpha \sim \mathcal{N}(0,1)$. Using the first moment of f and the variance as robustness measures, the multi-objective optimization task becomes

$$\left.\begin{aligned} \mathrm{E}[f|\mathbf{x}] &= \|\mathbf{x}\|^2 && \to \ \min \\ \mathrm{Var}[f|\mathbf{x}] &= (1-\|\mathbf{x}\|^2)^2 && \to \ \min \end{aligned}\right\}. \tag{29}$$

As one can see, $\mathrm{E}[f|\mathbf{x}]$ and $\mathrm{Var}[f|\mathbf{x}]$ have different minimizers and therefore they represent conflicting goals. Dealing with such scenarios, as a first approach one may aggregate the different objectives in a single function using a weighted sum of the objective functions (see e.g., [10, p. 265; 55, p. 138f; 56])

$$(1-\beta)\mathrm{E}[f|\mathbf{x}] + \beta\,\mathrm{Var}[f|\mathbf{x}] \to \min, \quad \beta \in [0,1]. \tag{30}$$

Alternatively, one can consider the set of *Pareto-optimal* solutions (for the example (28), (29), see Fig. 5). There are different techniques for determining the *Pareto-front*,[10] such as *compromise programming* (in context of robust optimization, see [58,59]) and others (see e.g., [50]). In the field of *evolutionary computation* the method of "dynamic weighted aggregation" has been proposed and applied to the bi-objective optimization problem of the mean-variance trade-off of some test functions in [60]. These authors also introduced *relative variance robustness measures* in that they divide the standard deviation of f, i.e., the square root of (26), by aggregated standard deviations of the design variable uncertainties. There is a growing number of papers on evolutionary algorithms for finding solutions to robust design optimization problems using approaches other than the aggregated sum technique in order to locate the Pareto front, see e.g., [61–64].

[9] There is evidence that increased genetic diversity reduces disease damage in natural ecosystems, see e.g., [54].

[10] Also referred to as "efficient front" in finance applications, see [57] and the references presented there.

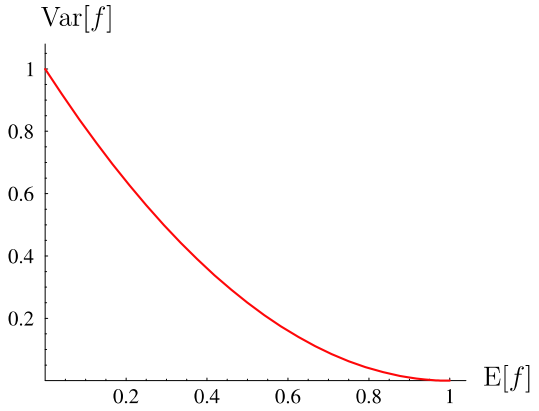

Fig. 5. Pareto-front of example function (28) using the goals (29): The curve represents the non-dominating solutions of the multi-objective optimization problem, i.e., those $(\mathrm{E}[f],\mathrm{Var}[f])$-values for which the improvement of one component results necessarily in a deterioration of the other component.

3.2.3. Probabilistic threshold measures of robustness

Instead of considering the expected value of utility functions $U(f)$ one can consider the distribution of the f-variate directly. In the case of *minimization* one is interested in obtaining f realizations sufficiently small. That is, given a *threshold* q, one is aiming at a maximum number of samples of f for which $f \leqslant q$. Given a fixed number of samples n, the expected number of samples fulfilling $f \leqslant q$ is $n\Pr[f \leqslant q]$. Therefore, one yields a threshold dependent criterion for robustness optimization

$$\Pr[f \leqslant q|\mathbf{x}] \to \max, \tag{31}$$

where the lhs in (31) measures the relative frequency of desired designs. Clearly, this relative frequency should be as large as possible. Since the lhs of (31) is the conditional cumulative distribution function P of the random variate f, $P(q|\mathbf{x}) = \Pr[f \leqslant q|\mathbf{x}]$, this criterion for *robust minimization* based on the threshold measure reads

$$F_q(\mathbf{x}) = P(q|\mathbf{x}) \to \max. \tag{32}$$

The optimal value $\hat{\mathbf{x}}$ of the design parameter(s) $\mathbf{x}$ depends on the choice of the threshold q. That is, as a result, $\hat{\mathbf{x}}$ is a function of q, i.e., $\hat{\mathbf{x}} = \hat{\mathbf{x}}(q)$. This is in contrast to the expectation measures introduced in Section 3.2.2 where, given a fixed utility function $U(f)$, the optimizer of (21) is usually a single state $\hat{\mathbf{x}}$.

Similarly, one obtains the threshold measure for robust *maximization*

$$\Pr[f > q|\mathbf{x}] \to \max. \tag{33}$$

Using $P(f)$ one gets $1 - P(q|\mathbf{x}) \to \max$. Therefore, the criterion for *robust maximization* based on the threshold measure reads

$$F_q(\mathbf{x}) = P(q|\mathbf{x}) \to \min. \tag{34}$$

There is an interesting connection between the threshold measures and the expectancy measures. The former can be considered as expectancy measures using *parameterized*

utility functions $U(f,q)$. Considering the case of maximization, Eq. (33) can be recovered from (21) using the utility function

$$U(f,q) = \Theta(f-q), \tag{35}$$

where $\Theta(t)$ is the step function

$$\Theta(t) := \begin{cases} 0, & t \leqslant 0, \\ 1, & t > 0. \end{cases} \tag{36}$$

This can easily be shown starting from (21) using (36)

$$\begin{aligned} \mathrm{E}[\Theta(f-q)|\mathbf{x}] &= \int \Theta(f-q)p(f|\mathbf{x})\,\mathrm{d}f \\ &= \int_{f>q} p(f|\mathbf{x})\,\mathrm{d}f = \Pr[f > q|\mathbf{x}]. \end{aligned} \tag{37}$$

Using rectangular or bump-like parameterized utility functions, one can extend the idea of threshold measures to define certain robustness ranges for f the probability of which is to be maximized. For example

$$\Pr[q_1 \leqslant f \leqslant q_2|\mathbf{x}] \to \max. \tag{38}$$

Applications of the proposed probabilistic threshold robustness measures have *not* been found in literature. There are likely two basic reasons for their non-appearance:

(a) Determining $P(f|\mathbf{x})$ analytically will almost always be excluded except for simple toy problems (see Section 5.2 for examples). On the other hand, calculating $P(f|\mathbf{x})$ numerically will usually be computationally expensive due to the need of Monte-Carlo simulations. Since $P(f|\mathbf{x})$ is to be optimized w.r.t. $\mathbf{x}$, these Monte-Carlo simulations are in the inner loop of an optimization algorithm. That is, the overall computational complexity increases considerably.
(b) While the robustness function $P(q|\mathbf{x})$ contains in principle all relevant (statistical) information of a design $\mathbf{x}$, its interpretation can be rather difficult: Maximizing $P(q|\mathbf{x})$ w.r.t. $\mathbf{x}$ yields an optimum design $\hat{\mathbf{x}}$ which depends on the threshold q. Unless the user has a clear picture of the meaning of q, fixing the threshold q can be a difficult decision problem.

However, there is a certain demand of using the probabilistic threshold measures in theoretical investigations. Especially in those cases where the statistical momentum measures *do not exist*, e.g., when Cauchy-noise is involved, these measures present a means to find robust optimizer states.

3.2.4. Statistical feasibility robustness

Considering uncertainty type D, a natural way of handling (general nonlinear) constraints

$$\mathbf{g}(\mathbf{x},\boldsymbol{\alpha}) \leqslant \mathbf{0} \tag{39}$$

is to guarantee these inequalities probabilistically. That is, e.g., demanding

$$\Pr[\mathbf{g}(\mathbf{x},\boldsymbol{\alpha}) \leqslant \mathbf{0}] \geqslant P_0, \tag{40}$$

where P_0 is the confidence probability and $\boldsymbol{\alpha}$ is a random variable (or vector) with probability density function (pdf) $p(\boldsymbol{\alpha})$. This way of expressing feasibility robustness is usually referred to as "*probabilistic* or *chance constraints* programming" in the stochastic programming literature [9].

Design optimization problems incorporating probabilistic inequalities (40) are also referred to as *reliability-based design optimization* (RBDO) problems in mechanical engineering literature (see e.g., [65–71]). However, reviewing the literature, there is no general consensus that RBDO should not be regarded as part of the robust optimization methodology. Therefore, agreeing with Park et al. [13], we also consider it in this survey.

Formally, inequality (40) can be expressed by integrals. Considering a single constraint $g_i(\mathbf{x},\boldsymbol{\alpha}) \geqslant 0$, this leads to

$$\Pr[g_i(\mathbf{x},\boldsymbol{\alpha}) \leqslant 0] = \int_{g_i(\mathbf{x},\boldsymbol{\alpha})\leqslant 0} p(\boldsymbol{\alpha})\,\mathrm{d}\boldsymbol{\alpha} \geqslant P_{0i}. \tag{41}$$

Since this integral depends on $\mathbf{x}$, we again end up with a nonlinear constraint of the form $\tilde{g}_i(\mathbf{x},\boldsymbol{\alpha}) \geqslant P_{0i}$. Thus, the *probabilistic* or *chance constraints* approach does not define a new class of optimization problems from the mathematical point of view. However, calculating $\tilde{g}_i$ appears as a technically involved problem analytically tractable for the simplest cases only (see e.g., [9, pp. 103–109]). In practice one has to resort to numerical approximations such as First-Order Reliability Methods (FORM) and Second-Order Reliability Methods (SORM, see also Section 4.2.1.1) or Monte-Carlo simulations. An overview and a comparison of feasibility modeling techniques can be found in [72].

Statistical feasibility robustness is often considered in conjunction with type (A) and (B) uncertainties. This kind of problems is also referred to as *reliability-based robust design optimization* [73,67,74].

3.2.5. Possibilistic uncertainties: getting the optimum under subjectivity

So far we have discussed the crisp and probabilistic characterization of uncertainty sets. While the former uniquely characterizes a design variable $\mathbf{x}$ or parameters $\boldsymbol{\alpha}$ as a feasible design alternative or not, the latter one does this probabilistically. Both kinds of characterization, however, are based on *complete* information about the design problem and its modeling. If this information is not available, various kinds of error sources have to be taken into account, such as the error introduced by the model used,[11] the uncertainties about the model's input parameters (boundary and operating condition) and its feasibility constraints. In all these cases one has to rely on experts' estimates which are intrinsically subjective. This is the domain of *epistemic* methods for characterizing and handling uncertainties.

[11] These kinds of errors are also referred to as *model form errors*, see e.g., [26].

Coping with epistemic uncertainties has not gained much attention in literature. There are basically two directions:

(a) iterative improvements of probability models using bootstrap methods based on Bayesian statistics [68] (this concerns the question how to incorporate new information), and
(b) treatment of uncertainty sets using fuzzy logic and evidence theory [23].

The fuzzy set methodology offers a mathematical rigorous way to quantify the membership of a design solution to the feasible solution space. That is, rather than characterizing a design $\mathbf{x}$ as feasible or infeasible, it is mapped to a membership function $\mu_D(\mathbf{x}) \in [0,1]$ describing the degree of membership to the feasible set. Here the extreme value $\mu_D = 1$ means that $\mathbf{x}$ is feasible and $\mu_D = 0$ that it is not feasible. The advantage is that this approach allows for intermediate membership grades $0 < \mu_D < 1$. Thus, the feasibility constraints of the crisp case (39) and of the probabilistic case (40) can be replaced by

$$\mu_D(\mathbf{x}, \boldsymbol{\alpha}) \geqslant \mu_0. \tag{42}$$

This concept can be extended to model uncertainties of the quality information of the objective function as well [75,76]: The idea is to associate the unconstrained optimum $f(\mathbf{x}^*)$ with the membership value $\mu_f(\mathbf{x}^*) = 1$ and the worst performing design with $\mu_f = 0$ and assigning intermediate values in between. Finding the optimal robust design $\hat{\mathbf{x}}$ is then done by maximizing the membership grade of the minimum of the membership grade of the constraints μ_D and of μ_f, i.e.,

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} [\min(\mu_D(\mathbf{x}), \mu_{\mathbf{f}}(\mathbf{x}))]. \tag{43}$$

The most radical approach to possibilistic modeling is to express the whole design problem using fuzzy numbers. To this end, the crisp problem formulation must be fuzzified. However, such an approach bears various subtleties when fuzzifying nonlinear functions, since this can be done using Zadeh's *extension principle* or via *α-cuts* [77,78] (with interval arithmetic) which can lead to different results [79]. Also, the numerical effort for solving the resulting min–max problems is computationally high when considering real applications.

Determining the concrete form of the membership functions and combining different fuzzy constraints is also a difficult part of this approach. Possibility theory provides the theoretical basis to quantify the notions of "belief" and "plausibility" and to aggregate opinions of different experts based on the Dempster–Shafer theory [80].

Optimization problems formulated in the possibility framework are also referred to as *evidence-based design optimization* [81] or *possibility-based design optimization* [82,83]. For an in-depth discussion of the possibilistic approach, the reader is referred to [84]. A simple application example using the concept of fuzzy numbers can be found in [85].

4. Robust optimization in practice

In Section 3, ideas for constructing robust counterparts F of objective functions f have been presented based on theoretical considerations without taking into account their usability in practical applications. Implementing these ideas, one has to face two problems: (a) determining $F(\mathbf{x})$ and (b) optimizing $F(\mathbf{x})$. There are different philosophies to handle these problems when facing real-world robust design optimization tasks. At the one end of the spectrum there is the *simplification strategy* aiming at a reduction (or transformation) of the problem to an optimization task that can be solved using standard techniques of mathematical programming. At the other end there are the *simulation optimization techniques* [86], predominantly used in engineering sciences where the function $F(\mathbf{x})$ to be optimized is usually not available analytically and/or the resulting optimization problem is not tractable by standard procedures from mathematics.

We will first review where mathematical programming can contribute to the field of robust optimization. Thereafter, deterministic numerical approximation techniques are considered. A strong focus will then be put on randomized approaches and direct search methods including a separate section devoted to evolutionary algorithms. These techniques allow for very flexible and direct robustness tests and evaluations.

4.1. Robust optimization using mathematical programming

By mathematical programming we mean exact (usually polynomial runtime) algorithms developed to solve convex constrained optimization problems. These are:

(P1) linear optimization problems (LP),
(P2) quadratically constrained quadratic optimization problems,
(P3) conic linear or quadratic optimization problems,
(P4) semidefinite optimization problems.

In [10] a robust counterpart approach for (P1) based on expectancy measures (cf. Section 3.2.2) has been proposed. It basically extends the idea of (linear) stochastic programming [9] by enhancing the linear objective function of the stochastic programming problem with a *nonlinear* regularization function penalizing constraint violations. In order to keep the resulting optimization problem tractable, the penalty function must be kept simple, e.g., quadratically. Approaches for treating more complicated nonlinearities do exist, see e.g., [11], however, these usually represent ad hoc approximations resulting in quadratic optimization problems with auxiliary variables the number of which increases linearly with the number of scenarios considered.

An approach based on robust regularization (worst case regularization, cf. Section 3.2.1) for least-square problems with uncertain data has been developed in [37]. Finding a solution $\mathbf{x}$ to the over-determined linear system $\mathbf{Ax} = \mathbf{b}$ can be formulated as a least-square minimization problem $\Delta := \|\mathbf{Ax} - \mathbf{b}\| \to \min$. The authors have shown that under certain conditions the robust optimum of Δ in the worst case scenario (w.r.t. uncertainties of $\mathbf{A}$ and $\mathbf{b}$) can be computed using convex quadratic cone and semidefinite programming techniques, respectively.

While the work of [37] is a special case of robust quadratic programming, the work of Ben-Tal and Nemirovski initiated increased research activities during the last ten years. Their theory is mainly based on the robust counterpart idea of Section 3.2.1. In various papers, e.g., [38,87,39], the authors developed a theory of robust linear, quadratic, and semidefinite programming considering systems with linear, conic quadratic, and linear matrix inequality constraints.

In [39] the authors also showed the significance of robust optimization for finding robust solutions in engineering applications such as robust antenna design, robust truss topology design, and control of uncertain dynamic systems. They also addressed investigations concerning the stability of solutions to linear programming test cases collected in the NETLIB library.[12] As have been observed by other authors long ago, the solutions of a considerable amount of these "real-world" test cases obtained by standard LP software are highly unreliable: Even very small perturbations of the solutions can result in severe constraint violations. Applying the robust counterpart methodology, the solutions to these test problems can be "immuned" against such perturbations.

Recently, first attempts have been made to extend the ideas developed for continuous search spaces to discrete optimization [88,43,46,44]. Unlike the case of continuous design spaces, the conditions on the uncertainty sets are rather restrictive to still ensure the efficient solvability of the robust counterpart. As has been shown in [44] general ellipsoidal uncertainty sets turn polynomially solvable problems, such as shortest path, minimum spanning tree, minimum cost assignment, and resource scheduling, into NP-hard counterparts. Therefore, approximation techniques based on piecewise linearization are currently the means of choice to tackle such problems.

Even though the methodology presented in this section represents a successful approach with strong theoretical background, its application domain seems somewhat restricted. Up until now, applications are mainly in the field of finances and portfolio management [89,57,90], but also engineering applications such as (linear) truss topology design, antenna design (basically a robust linear approximation problem) and robust (linear) control problems have been reported in [39]. In [91] the robust design of a truss has been modeled by a nonlinear semidefinite programming problem which has been solved by a sequence of semidefinite programs obtained by successive linearization.

The main disadvantage of this approach lies in the necessity to fit the real-world problem into a linear model with (at most) conic or quadratic constraints. Provided that the original real-world problem can be expressed analytically by an objective function $f(\mathbf{x}, \boldsymbol{\alpha})$, then there is a certain chance to construct approximate models. However, this is often bought at the expense of a system having a huge number of variables (because the nonlinearities must be approximated by piecewise linear or quadratic functions, etc.). Finding approximate models of tractable size is usually a non-trivial task. Nevertheless, the trade-off between modeling effort and optimization effort, but also the problem solver's preferences, finally decide whether to apply this approach or not.

However, if the values of the objective function $f(\mathbf{x}, \boldsymbol{\alpha})$ are not available as a closed mathematical expression and if they can only be obtained by simulations, the methodology presented in this section cannot be applied. Therefore, these techniques of robust optimization are restricted to cases where the original problem itself can be well approximated by linear or quadratic functions. In the other cases, one has to consider alternative approaches that we will discuss in the next section.

4.2. Robust optimization – the broader perspective

Reviewing the approaches to robust optimization taken in engineering science, one can differentiate two main classes:

(A) Methods which calculate the desired robustness measures $F(\mathbf{x})$ and the related (robust) constraints explicitly using numerical techniques. Thus, the resulting optimization problem is an ordinary one, to be solved using local or global optimization algorithms. This will be referred to as the *deterministic* approach to robust optimization.

(B) Treating the (probabilistic) uncertainties directly by optimizing *noisy functions and constraints*. This will be referred to as the *randomized* approach to robust optimization, sometimes referred to as Monte-Carlo techniques. Since the noisy information is usually obtained by simulation programs, this kind of optimization is also referred to as *simulation optimization* in OR literature [86].

The review of class (A) will concern methods on how to transform the robust optimization problem into an ordinary optimization problem, which can be solved by standard (and also non-standard) techniques of nonlinear programming. These optimization techniques usually rely on strong mathematical assumptions on the functions to be optimized, such as the availability of first- or second-order derivatives. Such information is not available in the class (B) approach. Therefore, class (B) is basically devoted to

[12] See http://cm.bell-labs.com/netlib/index.html and http://cm.bell-labs.com/netlib/lp/index.html.

direct search methods, i.e., algorithms which do not need explicit first- or second-order information, but only the $f(\mathbf{x})$-values to calculate the next search point. Class (B) also contains the *Evolutionary Algorithms* which will be reviewed in a separate third section. The algorithms collected in class (B) can – of course – also be used for the treatment of the ordinary optimization problems arising from class (A). The optimization algorithms from class (B) may be regarded as "general" problem solvers. This generality, however, usually has to be traded off against a lower efficiency when compared to standard nonlinear programming algorithms (e.g., quasi-Newton methods, augmented Lagrangian methods) which are designed by making strong assumptions on the structure of the optimization problem.

4.2.1. The deterministic approach to robust optimization

In this section, we will shortly review methods for determining robustness measures F using deterministic numerical techniques. It is assumed that the objective function $f(\mathbf{x},\boldsymbol{\alpha})$ and the constraints as well as its derivatives w.r.t. $\mathbf{x}$ and $\boldsymbol{\alpha}$, respectively, can be calculated analytically or numerically with sufficient accuracy. In literature, one finds almost exclusively methods for the expected utility models introduced in Section 3.2.2 and for feasibility robustness, Section 3.2.4.

4.2.1.1. Feasibility robustness. Methods for robust optimization using the expected value robustness measure and taking feasibility constraints into account can be found in [92], in which the authors focused mainly on the treatment of the constraints and presented three methods for incorporating the effect of uncertainties in the constraints. The method: "Constraints with Build-in Constraints Variation" presented there can be found in numerous follow-up papers where it is usually referred to as some kind of (linearized) *worst case analysis*. Considering the inequality constraint $g_i(\mathbf{x},\boldsymbol{\alpha}) \leqslant 0$, the effect of uncertainties $\boldsymbol{\delta}$ on $\mathbf{x}$ and $\Delta\boldsymbol{\alpha}$ on $\boldsymbol{\alpha}$ about a design point can be approximated by linear Taylor expansion as

$$g_i(\mathbf{x}+\boldsymbol{\delta},\boldsymbol{\alpha}+\Delta\boldsymbol{\alpha}) = g_i(\mathbf{x},\boldsymbol{\alpha}) + \sum_j \frac{\partial g_i}{\partial x_j}\delta_j + \sum_j \frac{\partial g_i}{\partial \alpha_j}\Delta\alpha_j + \cdots \leqslant 0. \tag{44}$$

Neglecting higher-order terms, design point $\mathbf{x}$ will surely fulfill (44) if it fulfills the *worst case* where the sum of the deviations in (44) assumes its maximum. Since the upper bound of these deviations is given by the absolute values of the summands, one obtains the inequality condition

$$g_i(\mathbf{x},\boldsymbol{\alpha}) + \sum_j \left|\frac{\partial g_i}{\partial x_j}\delta_j\right| + \sum_j \left|\frac{\partial g_i}{\partial \alpha_j}\Delta\alpha_j\right| \leqslant 0. \tag{45}$$

Alternatively, the inequality

$$g_i(\mathbf{x},\boldsymbol{\alpha}) + \sqrt{\sum_j \left(\frac{\partial g_i}{\partial x_j}\delta_j\right)^2 + \sum_j \left(\frac{\partial g_i}{\partial \alpha_j}\Delta\alpha_j\right)^2} \leqslant 0 \tag{46}$$

has been proposed in [92]. Considering (44) as a linear inequality at the nominal design point, the robust counterpart approach of mathematical programming (cf. Section 4.1) can be used to perform approximately robust optimization [93]. From this point of view, (45) and (46) are special cases. However, in order to apply both (45) and (46), the uncertainties must have bounded support (i.e., $0 \leqslant |\delta_i| < \hat{\delta}_i$ and $0 \leqslant |\Delta\alpha_j| < \hat{\alpha}_j$). Incorporating unbounded uncertainties can be done in the probabilistic framework using expected value considerations applied, e.g., to (46); this leads to the so-called *moment matching method* (see e.g., [94] or [72, p. 386f]).

In many applications (see below), the robust solutions have also to fulfill inequality or equality constraints probabilistically. Some sophisticated techniques for calculating expected value integrals taking constraints and uncertainties into account can be found in [50]. Simple techniques for handling constraints under uncertainties are also described in [72,95]. These are again based on worst case or confidence measures and replace the original constraints by (first-order) Taylor approximations at the nominal design point.

Numerical approximation techniques for calculating the acceptance probabilities in statistical feasibility models (40) have been developed in the field of reliability engineering and have been used in *reliability-based design optimization* (RBDO). Provided that the problem can be (approximately) transformed to an integral of the form

$$\Pr[g(\mathbf{x}) \leqslant 0] = \int_{g(\mathbf{x})\leqslant 0} p(\mathbf{x})\,\mathrm{d}\mathbf{x}, \tag{47}$$

where $p(\mathbf{x})$ is the standard normal distribution $\mathcal{N}(\mathbf{0},\mathbf{I})$. A first-order approximation to $\Pr[g(\mathbf{x}) \leqslant 0]$ is obtained as follows. First one has to find the *most probable point* (MPP) $\mathbf{x}_\mathrm{p}$ on the *limit state surface* $g(\mathbf{x}) = 0$ (i.e., the maximum of $p(\mathbf{x})$ on $g(\mathbf{x}) = 0$) by solving the constrained optimization problem. Since the joint density function $p(\mathbf{x})$ is spherically symmetric, this is equivalent to finding that point on $g(\mathbf{x}) = 0$ that is nearest to the coordinate origin

$$\mathbf{x}_\mathrm{p} = \arg\max_{\mathbf{x}:g(\mathbf{x})=0} \|\mathbf{x}\|, \tag{48}$$

to be determined by numerical techniques such as the Rackwitz–Fiessler algorithm (see, e.g., [96]). In a second step, $g(\mathbf{x})$ is expanded in a Taylor series around $\mathbf{x} = \mathbf{x}_\mathrm{p}$. In the case of the *First-Order Reliability Method* (FORM), the expansion is truncated after the linear terms resulting in

$$g(\mathbf{x}) \approx g(\mathbf{x}_\mathrm{p}) + \nabla^\mathrm{T} g(\mathbf{x}-\mathbf{x}_\mathrm{p}) = \nabla^\mathrm{T} g\mathbf{x} - \nabla^\mathrm{T} g\mathbf{x}_\mathrm{p}. \tag{49}$$

Eq. (49) represents the hyperplane separating the designs and $-\nabla g$ points into the direction of states $g(\mathbf{x}) < 0$. Due to the spherical symmetry, the problem reduces to an one-dimensional integration of the standard normal density function from $-\infty$ to the (signed) distance $\|\mathbf{x}_\mathrm{p}\|$. Thus, (47) can be approximated by

$$\Pr[g(\mathbf{x}) \leqslant 0] \approx \Phi(-\beta), \quad \text{where } \beta = \frac{-\nabla^{\mathrm{T}} g \mathbf{x}_{\mathrm{p}}}{\|\nabla g\|} \tag{50}$$

and $\Phi(z)$ is the cumulative distribution function of the standard normal variate. Considering quadratic terms in the Taylor expansion of $g(\mathbf{x})$ leads to *Second-Order Reliability Methods* (SORM), the reader is referred to [96] and the state-of-the-art review [97]. The embedding of these techniques in the reliability-based design framework and the different numerical approaches as single- vs. double-loop and decoupled optimization methods are reviewed in detail in [65].

4.2.1.2. Expected value robustness and related measures. Most of the applications on robust design (see below) use the expected value and variance measure to assess robustness of a design. However, as one can infer from Eqs. (23)–(27), calculating these measures analytically is almost always impossible. Therefore, approximation techniques must be used.

Almost all approximation techniques are based on Taylor expansions (see below). There is notably one exception in [14] where it has been proposed to calculate the expected value integrals (23) and (27) using numerical Gauss–Hermite integration. However, since the n-dimensional integrals are approximated by n-fold sums, one has to face the curse of dimensions. Therefore, this approach can only be applied for low-dimensional design spaces (in the case of [14] it was four-dimensional).

In [55], the authors presented an approach based on the Taylor expansion of the objective function $f(\mathbf{x}, \boldsymbol{\alpha})$. We will briefly sketch the approach for the case of type B uncertainties. Using $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_N)^{\mathrm{T}}$ the Taylor expansion of f including second-order terms reads

$$f(\mathbf{x} + \boldsymbol{\delta}) = f(\mathbf{x}) + \sum_{i=1}^{N} \frac{\partial f}{\partial x_i}\delta_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\partial^2 f}{\partial x_i \partial x_j}\delta_i\delta_j + \cdots \tag{51}$$

Let us assume that

$$\mathrm{E}[\boldsymbol{\delta}] = \mathbf{0}, \tag{52}$$

which is a reasonable assumption, since a systematic deviation $\mathrm{E}[\boldsymbol{\delta}] = \boldsymbol{\delta}_c \neq \mathbf{0}$ could be easily incorporated in the design. Considering the linear approximation first, the expected value of f w.r.t. $\boldsymbol{\delta}$ becomes

$$\mathrm{E}[f|\mathbf{x}]_1 = f(\mathbf{x}) \tag{53}$$

and the variance

$$\begin{aligned}\mathrm{Var}[f|\mathbf{x}]_1 &= \mathrm{E}\left[\left(f(\mathbf{x}) + \sum_{i=1}^{N}\frac{\partial f}{\partial x_i}\delta_i - \mathrm{E}[f|\mathbf{x}]_1\right)^2\right] \\ &= \mathrm{E}\left[\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial f}{\partial x_i}\frac{\partial f}{\partial x_j}\delta_i\delta_j\right] \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial f}{\partial x_i}\frac{\partial f}{\partial x_j}\mathrm{E}[\delta_i\delta_j].\end{aligned} \tag{54}$$

Introducing the covariance matrix $\mathbf{C}$

$$(\mathbf{C})_{ij} := \mathrm{E}[\delta_i\delta_j] \tag{55}$$

and the gradient vector ∇f, Eq. (54) can be expressed as

$$\mathrm{Var}[f|\mathbf{x}]_1 = \nabla^{\mathrm{T}} f \mathbf{C} \nabla f. \tag{56}$$

If the uncertainties are *not* correlated, $\mathbf{C}$ will be a diagonal matrix containing the δ_i-specific variances $\sigma_i^2 := \mathrm{Var}[\delta_i] = \mathrm{E}[\delta_i^2]$. In such cases (56) simplifies to

$$\mathrm{Var}[f|\mathbf{x}]_1 = \sum_{i=1}^{N}\left(\frac{\partial f}{\partial x_i}\right)^2\sigma_i^2. \tag{57}$$

Provided that there is a way to calculate the derivatives of f analytically or numerically, Eqs. (56) or (57) may be used to estimate the variance. Inserting (53), (56), (57), respectively, into Eq. (30) we arrive at the so-called *sensitivity robustness approach* [55, p. 139].

Applications of the *sensitivity robustness* approach with weighted sum of objectives can be found in (note, this is not an exhaustive collection, it just presents typical examples):

- [98] deals with the geometry optimization of a magnetic pole face of a motor.
- [56] presents nearly the same approach as in [98] applied to some simple toy problems in mechanical design.
- [99] uses this approach for the robust minimization of the negative square of the lift-to-drag ratio of a 3-D flexible wing.
- [100] presents a robust design optimization of the truss structure of a parabola antenna. It uses sequential quadratic programming to solve the resulting optimization problem.

The *sensitivity robustness* approach assumes the mean value of f to be equal to f at the nominal design point $\mathbf{x}$, Eq. (53). In Section 5.2.3, we will present test functions derived from a real-world problem where this assumption would lead to disastrous results. While not being a final solution to this problem, considering higher-order approximations can improve the situation. To this end, we take the quadratic terms in (51) into account. Calculation of the expected f value yields with (52) and (55) immediately

$$\begin{aligned}\mathrm{E}[f|\mathbf{x}]_2 &= f(\mathbf{x}) + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial^2 f}{\partial x_i \partial x_j}\mathrm{E}[\delta_i\delta_j] \\ &= f(\mathbf{x}) + \frac{1}{2}\mathrm{Tr}[\mathbf{HC}],\end{aligned} \tag{58}$$

where $\mathbf{H}$ is the Hessian of f at $\mathbf{x}$, i.e.

$$(\mathbf{H})_{ij} := \frac{\partial^2 f}{\partial x_i \partial x_j} \tag{59}$$

and the trace of a matrix $\mathbf{A}$ is defined as

$$\mathrm{Tr}[A] = \sum_i (\mathbf{A})_{ii}. \tag{60}$$

Calculating the variance using (51) and (58) yields

$$\begin{aligned}\mathrm{Var}[f|\mathbf{x}]_2 &= \mathrm{E}\left[\left(f(\mathbf{x}) + \sum_{i=1}^{N}\frac{\partial f}{\partial x_i}\delta_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial^2 f}{\partial x_i \partial x_j}\delta_i\delta_j - \mathrm{E}[f|\mathbf{x}]_2\right)^2\right] \\ &= \mathrm{E}\left[\left(\sum_{i=1}^{N}\frac{\partial f}{\partial x_i}\delta_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial^2 f}{\partial x_i \partial x_j}\left(\delta_i\delta_j - \mathrm{E}[\delta_i\delta_j]\right)\right)^2\right] \\ &= \sum_{ij}\frac{\partial f}{\partial x_i}\frac{\partial f}{\partial x_j}\mathrm{E}[\delta_i\delta_j] \\ &\quad + \sum_{ijk}\frac{\partial f}{\partial x_k}\frac{\partial^2 f}{\partial x_i \partial x_j}\left(\mathrm{E}[\delta_i\delta_j\delta_k] - \mathrm{E}[\delta_i\delta_j]\mathrm{E}[\delta_k]\right) \\ &\quad + \frac{1}{4}\sum_{ijkl}\frac{\partial^2 f}{\partial x_i \partial x_j}\frac{\partial^2 f}{\partial x_k \partial x_l}\left(\mathrm{E}[\delta_i\delta_j\delta_k\delta_l] - \mathrm{E}[\delta_i\delta_j]\mathrm{E}[\delta_k\delta_l]\right). \end{aligned} \tag{61}$$

This expression can be further simplified assuming *correlated Gaussian noise*, i.e.

$$\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) \tag{62}$$

with covariance matrix $(\mathbf{C})_{ij}$. Then it holds (see e.g., [101] or [102, p. 356f])

$$\mathrm{E}[\delta_i\delta_j\delta_k] = 0, \quad \mathrm{E}[\delta_i\delta_j\delta_k\delta_l] = C_{il}C_{jk} + C_{jl}C_{ik} + C_{kl}C_{ij} \tag{63}$$

and (61) finally becomes with (56) and (59)

$$\mathrm{Var}[f|\mathbf{x}]_2 = \mathrm{Var}[f|\mathbf{x}]_1 + \frac{1}{2}\mathrm{Tr}[(\mathbf{HC})^2]. \tag{64}$$

Inserting (64) and (58) into Eq. (30) we arrive at the so-called *mean-variance robustness approach* [55, pp. 139ff].

Applications of the *mean-variance robustness* approach can be found in (note, this is not an exhaustive collection, it just presents typical examples):

- [103], where the expected value and variance of f are obtained by weighted sums of f-samples taken according to a *fractional quadrature factorial design*.
- [104], where it is applied to aircraft design. The authors used an automatic differentiation tool to generate Fortran 77 code of the derivatives needed to determine mean and variance of a second-order model (Taylor approximation) which is then optimized using standard numerical optimization techniques.
- [105], dealing with robust shape optimization of an airfoil using a weighted sum of performance values (drag coefficients) to be minimized under operation constraints (angle of attack, Mach number) numerically solved by a variant of sequential linear programming.

4.2.2. The randomized approach to robust optimization

Methods subsumed under this section may be regarded as *direct* approaches to robust optimization in that they directly incorporate the uncertainties into the generic optimization problem (1). Provided that the uncertainties are of probabilistic type, the objective function $f(\mathbf{x}, \boldsymbol{\alpha})$ becomes a random function $\tilde{f}$ because of Eqs. (8) and (9). That is, in the direct approach, the effect of the uncertainties can directly be evaluated and one can use any simulation technique to obtain objective function values. The problem is, however, how to utilize these information. In order to perform robust optimization, the respective robustness measures of Section 3.2 must be calculated from the observed $\tilde{f}$ values. Generally, the question arises how to use the raw data obtained. There are basically three categories:

(A) Monte-Carlo (MC) strategies: Given a fixed design point $\mathbf{x}$, calculate the statistics (mean value, variance, etc.) and use the statistical parameters obtained as input for a deterministic (derivative free) numerical optimization algorithm. More elaborated approaches usually take into account the special problem structure and are referred to as *sample path optimization* strategies [86, p. 154].
(B) Meta-model approach: A meta-model is constructed using a set of design points $\mathbf{x}$ carefully chosen. The optimizer of the meta-model is used as an estimate of the real robust optimizer.
(C) Use the $\tilde{f}$ values directly as inputs of an optimization algorithm especially suited for noisy optimization.

An overview of the state-of-the-art of direct simulation optimization techniques can be found in [106]. Unfortunately, the author almost completely ignored the *Evolutionary Algorithms* (Evolution Strategies, Genetic Algorithms) widely used in engineering optimization. Therefore, we will present below our own brief review of these categories from viewpoint of robust design optimization.

If one is interested in expectancy robustness measures (cf. Section 3.2.2), averaging over a fixed number κ of $\tilde{f}(\mathbf{x})$ samples (keeping the design point $\mathbf{x}$ constant) represents a simple possibility to obtain robustness estimates. However, this naive MC approach is computationally expensive; and w.r.t. optimization it remains an open issue which degree of accuracy, controlled by κ, is actually needed to obtain a certain robust solution quality. With respect to assessment of statistical feasibility robustness, specially tailored Monte-Carlo methods, such as importance or Latin hypercube sampling [107], are an alternative if FORM/SORM approximations are suspected to yield wrong results due to the nonlinearity of the limit state surface [97]. Such an approach has been taken in conjunction with Evolution Strategies (see below) in order to find earthquake-resistant steel frame designs [108]. Further examples using MC techniques can be found in [109–112].

Option (B) uses the observed $\tilde{f}$ values to build a meta-model $F_M(\mathbf{x})$ of the robust counterpart of $f(\mathbf{x}, \boldsymbol{\alpha})$. The meta-model F_M is usually a simple function of $\mathbf{x}$ which depends on a set of model parameters $\boldsymbol{\beta}$ to be tuned (optimally) such that it predicts the observed data $\tilde{f}$ well. Having a model of the observed data, the optimal design point $\hat{\mathbf{x}}_M$ of the meta-model can be easily calculated and can serve as an approximation of the original robust optimum of f.

Meta-modeling techniques are reviewed in [113,114]. In the context of robust design optimization, the *response surface methodology*, *neural networks*, and *Kriging* models have

been proposed as meta-modeling techniques. In [115], these three techniques have been evaluated for a robust two-bar design problem with $N = 2$ unknown design parameters. In the application considered, Kriging models appeared as the method with the best results. However, apart from this study, comprehensive empirical investigations comparing the different meta-modeling techniques from viewpoint of robust optimization are still missing.

Even though there are several papers on robust optimization using meta-modeling techniques, especially concerning *response surface methodology*, these techniques *are not* well suited for large-scale robust optimization problems when the number of design variables N is large. There are basically two problems:

First, the model complexity: For example, a fully quadratic response surface $F_M(\mathbf{x}, \boldsymbol{\beta})$ comprises $\mathcal{O}(N^2)$ free β parameters. In order to determine these parameters, $\mathcal{O}(N^2)$ function evaluation of $\tilde{f}$ are needed – a prohibitive number when considering large-scale simulation models. Although sophisticated sampling techniques have been proposed such as the *Latin hypercube sampling* and *fractional factorial design* (see [113,116]), this information theoretical complexity limit can principally not be bypassed.

Second, similar to Newton strategies in numerical optimization, the (quadratic) model optimum will usually be a first approximation only. That is, the meta-model must be repeatedly applied in order to get closer to the robust optimum. However, unlike one of the standard assumption in Newton strategies ensuring the convergence to the optimizer, the meta-model is *not* allowed to shrink arbitrarily (because it still has to cover the uncertainties). Under such conditions, it appears very unlikely that the (quadratic) meta-model represents the robust counterpart adequately.

There is a third problem with the meta-model approach, especially for the *response surface methodology*, which is due to the data uncertainties which lead to uncertainties in the model parameters $\boldsymbol{\beta}$ itself. Finding the model parameters $\boldsymbol{\beta}$ should be regarded as a robust optimization problem itself. Actually, this has been proposed in [117], where the robust regularization approach of Section 3.2.1 has been used to derive a *robust* response surface.

Although the application of the response surface methodology and other meta-models in robust optimization has obvious weaknesses, it is often applied, e.g., in

- [21], where the authors optimized a solar powered irrigation system.
- [95], where the authors applied it to a multidisciplinary mechanical design (six-link function-generator linkage) consisting of two subsystems.
- [116], where the authors used *Latin hypercube sampling* strategy and a physically motivated utility function as robustness measure for the design optimization of the femoral component for total hip arthroplasty.
- [110], where the authors used a Kriging-based approximation model for the optimal mean-variance design of an on silicon micro gyroscope.

To summarize, the application of meta-modeling can be useful in cases where the design space dimensionality N is not too large. However, the computational efficiency of the meta-model approach is at least questionable. The critique given by Trosset [19] concerning the Taguchi method (cf. p. 2), which may be regarded as a meta-model approach as well, applies to any kind of meta-models. From a more mathematically rigorous point of view, it is even not clear what the response surface methodology really produces as iteration time $t \to \infty$: Up to now no proof has been presented that this methodology yields a robust optimum based on any of the robustness measures defined in Section 3.2.[13]

Option (C) represents the most direct usage of noisy information. To this end, the direct optimization algorithms[14] must be able to cope with the noise. In the remainder of this section, we will briefly review some of the techniques found in literature except the *Evolutionary Algorithms*, which are to be reviewed separately in the next section.

Four types of direct search methods in the presence of noise can be identified:

(I) gradient estimation techniques, usually referred to as *stochastic approximation methods* (see below),
(II) *pattern search methods* (see below), and
(III) optimization techniques based on *response surface methodology*. This type of methods iteratively generate sequences of response surfaces. The local optimum or improvement directions predicted by the response surfaces serve as predictors for an improved design point about which a new response surface is established. In order to be computationally efficient, very simple response surfaces must be used. An elaborate algorithm for low-dimensional problems using a quadratic response surface has been proposed and analyzed in [118]. Since the optimization approach based on the response surface methodology is very similar to the one that has been described under Option B in this section, it will not be considered further.
(IV) In [119] a *tabu search* algorithm has been proposed for finding robust solutions to a discretized one-dimensional multi-modal test function and to knapsack problems with type B uncertainties. While this algorithm shares certain similarities with *evolutionary algorithms*, it currently represents a singular application and will not be considered further in this survey.

[13] This obvious deficiency does not imply that meta-modeling techniques cannot be practically useful in improving robust system design.

[14] By "direct optimization algorithms" we refer to methods which only use values of the function $\tilde{f}$ to be optimized as inputs, but no gradient or higher-order derivative information. However, this does not exclude that the algorithm calculates internally an estimate of these derivatives.

The techniques I–III are usually aiming at the solution of optimization problems of type

$$\left.\begin{array}{ll} \text{optimize:} & \mathrm{E}[U(f)|\mathbf{x}], \quad \text{(a)} \\ \text{subject to:} & \mathbf{x} \in \mathscr{X}, \quad \text{(b)} \end{array}\right\} \tag{65}$$

where $\mathscr{X} \subseteq \mathbb{R}^N$ is the space of feasible solutions. That is, these techniques can be mainly applied to optimize expectancy robustness measures as defined in Section 3.2.2.

The idea of *stochastic approximation* dates back to the work of Robbins and Monro [120]. Combined with the usual gradient search strategy, it yields an iterative update formula for the design parameters $\mathbf{x}$

$$\mathbf{x}^{(t+1)} := \Pi_{\mathscr{X}}[\mathbf{x}^{(t)} - a^{(t)}\tilde{\boldsymbol{\gamma}}(\mathbf{x}^{(t)})]. \tag{66}$$

Here the function $\Pi_{\mathscr{X}}[\mathbf{y}]$ projects a vector $\mathbf{y} \in \mathbb{R}^N$ to the nearest point in the set of feasible points $\mathscr{X}$ and $\tilde{\boldsymbol{\gamma}}(\mathbf{x}^{(t)})$ is an estimate of the gradient $\boldsymbol{\gamma}(\mathbf{x}^{(t)}) = \nabla_{\mathbf{x}}\mathrm{E}[U(f)|\mathbf{x}]$. Under the condition that $U(f(\mathbf{x}))$ does not grow faster than a quadratic function and the sequence of step-sizes $a^{(t)}$ fulfills

$$\sum_{t=1}^{\infty} a^{(t)} = \infty \quad \text{and} \quad \sum_{t=1}^{\infty} (a^{(t)})^2 < \infty, \tag{67}$$

it can be shown that (66) converges to the *minimizer* of $\mathrm{E}[U(f)|\mathbf{x}]$. More elaborated versions of algorithm (66) are referred to as *stochastic quasi-gradient methods* [121]. Even though these algorithms guarantee local convergence on a quadratic model, the convergence rate of $1/\sqrt{t}$ is slow. This is due to the choice of $a^{(t)} = a^{(0)}/t$ ensuring the validity of (67).

In order to derive convergence proofs for stochastic approximation algorithms one usually has to assume the existence of the (first) moments of the function to be optimized. Therefore, e.g., cases with Cauchy-noise are not well covered. Interestingly, the $(\mu/\mu_I, \lambda)$-Evolution Strategy[15] does not suffer from such limitations. As has been shown in [122], this kind of biologically inspired direct search technique can easily cope with Cauchy noise.

Several improvements of the conceptual algorithm (66) have been proposed in order to increase its computational efficiency and convergence rate at the price of sacrificing convergence guarantee. There are notably two algorithms: 1. Spall's *simultaneous perturbation stochastic approximation* (*SPSA*) [123], that estimates the gradient by just two function evaluation per iteration. 2. Kelley's *implicit filtering* [124] which uses the central difference formula to estimate the gradient and employs a specially tailored step-size adaptation technique.

Pattern search methods build a class of direct optimization methods which intentionally *do not* approximate the local gradient or Hessian of the function f to be optimized. They rather generate search points according to a pattern (where the distance between these points can be large as opposed to the difference schemes of derivative approximators) and accept those points which appear as improvements (w.r.t. its f-values) over the prior search points. A short review of these methods can be found in [125] (aspects of noisy optimization are not considered there).

Even though numerically not very stable, Nelder and Mead's *simplex strategy* [126][16] is according to [128] the most popular direct search method. For noisy optimization problems improved simplex strategies have been proposed in [129,130]. The latter paper seems to be the only one presenting a convergence proof for a pattern search method under noise. The proof is based on the assumption that the standard deviation of the noise reduces faster than the step size when approaching the optimum. Theoretical investigation as to the *solution quality* in the case that the noise does *not* decrease (this is the interesting case for robust optimization) has not been done.

Pattern search methods have a certain appeal because they directly search the design space by inspecting pattern points on the basis of "necessity". That is, pattern search steps are only performed up to the next improvement. This is in contrast to the response surface methodology where a constant number of test designs must be generated to build the response surface used to determine the next design point. While this argument seems to be obvious in the *non*-noisy case, things might be different if noise deceives the f information leading to wrong acceptance decisions.

Interestingly, application of direct pattern search methods in robust design optimization are hard to find. This raises the question whether these methods have simply not been considered up to now or whether they are not well suited for robust design optimization. When considering the case studies on robust designs published, it appears that most of these examples are dealing with a surprisingly small number N of design variables. Therefore, when considering high-dimensional design spaces, then things might change. Obviously, there is a need to reevaluate the different robust design optimization methodologies on a test bed consisting of *scalable* problems. We will come back to this issue in Section 5.

4.2.3. Evolutionary algorithms in the context of robustness

Evolutionary algorithms (EA) are population based probabilistic direct search and optimization algorithms gleaned from principles of Darwinian evolution. Starting with an initial population of μ candidate designs $\mathbf{x}_m$ (also referred to as "parents") and the corresponding observed (calculated, simulated, or measured) objective function values $f_m = f(\mathbf{x}_m)$ (also referred to as "fitness" values), an *offspring* population of λ designs $\tilde{\mathbf{x}}_l$ is created from the parents using variation operators. These variation operators change each single parent $\mathbf{x}_m$ randomly according to a specific probability distribution. This process is referred to as *mutation*. If more than one parent is involved in the

[15] This kind of direct optimization technique belongs to the class of *Evolutionary Algorithms* (*EA*), to be discussed in Section 4.2.3.

[16] For a convergence analysis of the simplex strategy in the deterministic case, see [127].

procreation of an offspring individual $\tilde{\mathbf{x}}$, one speaks of *recombination*.

While mutation and recombination produce diversity in the design space, an antagonist is needed giving the evolution a direction toward the optimum. This is done by so-called *selection operators*, which are designed to choose those offspring individuals as parents for the next generation which exhibit high fitnesses (i.e., small f values for minimization, large f values for maximization). Depending on the manner in which the variation and selection operators are designed and the spaces in which they act, different classes of evolutionary algorithms (EA) have been proposed, known as *Evolution Strategies (ES)* [34,28], *Evolutionary Programming* (*EP*) [131], *Genetic Algorithms* (*GA*) [30], and Genetic Programming [132]. For a comprehensive overview, the reader is referred to [133]. Note, in this review we do not discuss *simulated annealing* [134–136] explicitly. With respect to our terminology, simulated annealing appears as a special case of an evolutionary algorithm comprising one parent, one offspring, a mutation operator, and a time dependent probabilistic selection operator.

From viewpoint of optimization, EA may be regarded as "yet another direct optimization technique" applicable to general optimization problems defined by (1). While EA can be successfully applied to *deterministic* robust counterpart functions F of f, we will especially consider EA for *noisy* optimization. From this point of view, they may be regarded as an alternative to the other direct optimization algorithms reviewed in Section 4.2.2. Such applications of EA usually take as f-inputs the results of Monte-Carlo-like simulations. These inputs can be regarded as estimates of the theoretical robustness measures such as expectancy measures defined in Section 3.2.2 or (rough) estimates of robust regularizers according to Section 3.2.1. EA applications of these types are to be found in (chronological order, not exhaustive):

- [137], where EP has been used to evolve neural networks, robust w.r.t. random deletion of weights and biases of the neurons.
- [138], where an ES algorithm has been applied to the evolution of robust optical filter designs.
- [139], where a specially tailored GA for robust optimization for financial time series prediction has been suggested. While this paper does not explicitly take uncertainties into account, a robustness measure based on mean value and variance has been used and a fitness sharing model has been proposed aiming at avoiding a high concentration of individuals in the vicinity of sharp (i.e., instable) peaks.
- [35], where a "re-evaluating" distributed GA" for wing-box optimization has been introduced. The authors have used the robust regularization idea (9) to evolve robust designs.
- [140] (in German) where the problem of robust design of multilayer optical coatings [138] has been analyzed using ES. An excerpt of this work can be found in:
- [51], where the thickness and the refraction indexes are treated as uncertain design variables, i.e., as type B uncertainties.
- [36], where the robust regularization (9) of Section 3.2.1 has been realized approximately by sampling the $f(\mathbf{x}, \boldsymbol{\alpha})$ function in the neighborhood of the design point $\mathbf{x}$ (or of the nominal $\boldsymbol{\alpha}$, respectively) and taking the maximum (or minimum). This minimax approach, termed "two-space GA" (because in general, $\mathbf{x}$- and $\boldsymbol{\alpha}$-space are sampled independently), has been applied to parallel machine scheduling.
- [141], where type D feasibility uncertainties have been analyzed. A chance-constrained GA has been presented based on Monte-Carlo sampling and Latin hypercube sampling, respectively.[17]
- [109], where an automotive inner structure panel is optimized using a hybrid genetic algorithm optimizing the frame topology while a nonlinear programming algorithm optimizes the size parameters.
- [142], where a *multi-objective* GA has been employed to find an optimal design of an automobile valve-train with the two criteria manufacturing costs and performance variance and several constraints.
- [143], where the robust regularization and expectancy measures for the robust optimization of machine scheduling has been used.
- [144], where the expectancy measure has been applied to evolve robust maintenance schedules for a railway company.
- [67], where besides the application of neural networks to MC-driven reliability-based design optimization, a multi-criteria robust optimization problem for the design of a 39-bar truss has been solved using a multi-objective $(\mu + \lambda)$-Evolution Strategy. The same authors developed also a so-called *cascade evolutionary algorithm* for multi-objective optimization in [62].
- [145], where the multi-objective NSGA-II Evolutionary Algorithm has been used to find robust design of compressor fan blades against erosion.
- [146], where evolutionary algorithms are combined with surrogate models for robust optimization, [147], where trust region methods are combined with evolutionary algorithms and [148] where multi-objective optimization is combined with surrogate models for robust design optimization.
- [149], where evolutionary algorithms have been employed to generate maximally different alternative solutions to an optimization problem that is only incompletely represented by a mathematical model.
- [150], where a priori information on the desired robustness of the final design has been used with a multi-objective evolutionary algorithm that converges to a solution with good nominal performance and maximal robust-

[17] Note an EA combining feasibility robustness and type A – type C uncertainties has not been found in literature.

ness. Design of experiment methods were used to reduce the computational cost which is high for inverse methods.

Most of the applications presented rely – when using expectancy measures – on resampling strategies in order to reduce the noise. That is, given a design point, a number of κ samples is generated and the average of the observed $f(\mathbf{x})$ values is used as (still noisy) estimate of the robustness measure. The question arises how to choose κ and whether it makes sense to devote the same number κ of re-evaluations to each of the λ design points in the population. Branke [52] empirically investigated various reevaluation strategies, e.g., evaluating seemingly good individuals more often. However, the results obtained w.r.t. the computational efficiency were not very conclusive and sampling each population member just once appears not a bad choice (see also [53, Chapter 8] for further considerations).

The latter observation is in accordance with theoretical findings concerning the steady state solution quality of $(\mu/\mu_I,\lambda)$-ES[18] on simple test functions. In [49], the performance of the $(\mu/\mu_I,\lambda)$-ES on a quadratic N-dimensional sphere model with normally distributed actuator noise $\boldsymbol{\delta}\sim\mathcal{N}(\mathbf{0},\varepsilon^2\mathbf{I})$, i.e., type B uncertainties, has been analyzed. In [151], the analysis has been extended to the general quadratic fitness model (maximization considered)

$$f(\mathbf{x}) := \mathbf{b}^{\mathrm{T}}\mathbf{x} - \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} \tag{68}$$

($\mathbf{Q}$, positive definite symmetric matrix) having the robust counterpart (24)

$$F_1(\mathbf{x}) = \mathrm{E}[f|\mathbf{x}] = f(\mathbf{x}) - \varepsilon^2\mathrm{Tr}[\mathbf{Q}]. \tag{69}$$

The resulting expected steady state fitness error $\mathrm{E}[\Delta\tilde{f}]$ of the ES with μ parents and λ offspring is given by [151]

$$\mathrm{E}[\Delta\tilde{f}] \geqslant \frac{N\varepsilon^2\mathrm{Tr}[\mathbf{Q}]}{8\mu^2 c^2_{\mu/\mu,\lambda}}\left(1+\sqrt{1+\frac{8\mu^2 c^2_{\mu/\mu,\lambda}\mathrm{Tr}[\mathbf{Q}^2]}{\mathrm{Tr}[\mathbf{Q}]^2}}\right), \tag{70}$$

where $c_{\mu/\mu,\lambda}$ is a strategy-specific constant (see Appendix B) which depends asymptotically only on the μ/λ-ratio. Making the substitution $s := 8\mu^2 c^2_{\mu/\mu,\lambda}$, one can easily see that (70) is a monotonously decreasing function of s. Therefore, the error $\mathrm{E}[\Delta\tilde{f}]$ can be reduced by increasing the population size (keeping $\mu/\lambda = \text{const.}$). Furthermore, a closer analysis shows that resampling each individual κ times results in a change from $s\propto\mu^2$ to $s\propto\kappa\mu^2$. That is, resampling helps decreasing the error at a price of $\kappa\lambda$ function evaluations per generation. Assuming the same number of function evaluations, one can alternatively increase the population by the factor κ with the result $s\propto(\kappa\mu)^2$. In this case, we can clearly see that increasing the population size is the more efficient strategy compared to resampling. Recently, the same conclusion has been drawn on a more complex test function to be discussed in Section 5.2.3.

Using only one sample per design point is also standard in *GAs with Robust Solution Searching Scheme (GAs/RS)* proposed by Tsutsui et al. [152,153]. In [154], Tsutsui showed empirically that resampling is less efficient than using just one sample per design point. The GAs/RS addresses type B uncertainties. Since the standard GA [30,33] operates on a *genotype space*, usually the space of bit-strings $\{0,1\}^\ell$ of length ℓ, a so-called *genotype-phenotype mapping* is needed to transform the bit-strings $\mathbf{b}$ to the elements $\mathbf{x}$ of the design space $\mathscr{X}\subseteq\mathbb{R}^N$. The elements of the design space are referred to as the *phenotypes*.

Tsutsui's approach to robust design applies actuator noise to the *phenotypic level*. This is in accordance with (7). However, the GA itself operates on the genotypes. Theoretically, it is not clear which impact this division among the two spaces has e.g., with respect to convergence. The theoretical considerations presented in [153] are based on Holland's [155] schema theorem and are valid for an *infinite* population size λ only. It is claimed in [153] that the GAs/RS with *fitness-proportionate* selection operator calculates implicitly $\mathrm{E}[f|\mathbf{x}]$, i.e., the robust counterpart (24) for $\lambda\to\infty$. At the same time, it is well known that a GA with fitness-proportionate selection operator does *not* converge to the optimum [156,157]. Tsutsui's empirical investigations show that GAs/RS are able to locate robust solutions approximately for low-dimensional problems and large population sizes. Unfortunately, a more rigorous analysis both empirical (problem dimensions etc.) as well as theoretical (expected steady state fitness error, etc.) has not been carried out.

One must add that almost all direct optimization algorithms with noise lack this rigorous analysis. Nevertheless, there is reason to believe that EA are especially good at *direct* robust optimization (i.e., without calculating deterministic robust counterparts of f explicitly). Looking at nature, it seems that species are well adapted to their environments and that the structures evolved through natural evolution appear to be *optimal on average*. That is, the *species as a whole* exhibits a certain "optimality" while the individuals itself are *not* concentrated at a singular point in the "design space". This is just what one is looking for in robust design. Taking this seriously, the application of EA in robust design might lead to a *new quality in* and a *new approach to* robust optimization. That is, the EA is not only used as a well working optimization strategy under noise, but it serves as a *robustness optimizing strategy per se*.

When looking at the EA-literature on robust optimization, this point of view seems to be not so well established: Uncertainties are almost always added explicitly. However, one may regard the mutation operator itself as a *robustness tester*:[19] If mutations $\mathbf{z}$ act directly on the design variables, as is customary in ES/EP[20], an offspring individual $\tilde{\mathbf{x}}$ is

[18] For a definition of this algorithm, see Appendix A.

[19] Indeed in biology, mutations are noise for the inheritance (and cell division) processes against which the system has to be sufficiently robust to succeed.

[20] See e.g., the example of the $(\mu/\mu_I,\lambda)$-ES in Appendix A.

formed by adding $\mathbf{z}$ to the recombinant $\langle \mathbf{x} \rangle$ and robustness is tested by adding actuator noise $\boldsymbol{\delta}$ to the offspring design $\tilde{\mathbf{x}}$. As a result, the function for evaluating the robustness of an offspring design becomes

$$\tilde{f} = f\left(\langle \mathbf{x} \rangle + \underbrace{\mathbf{z} + \boldsymbol{\delta}}_{:=\tilde{\mathbf{z}}}\right). \tag{71}$$

That is, the effect of mutation and type B uncertainties can be emulated by just another mutation $\tilde{\mathbf{z}}$. In order to have this work, *mutations and parameter uncertainties must enter the system at the same level*. That is why, this approach *cannot* be used in GAs with genotype-phenotype mapping such as Tsutsui's GAs/RS considered above. However, for ES algorithms this idea might open up an approach for simple and efficient robust optimization strategies. An ES taking advantage of this idea has been proposed in [158].

A similar idea is due to Thompson [159] who stated that "Population dynamics can give rise to some level of fault tolerance 'for free'." Applied to an GA operating on a binary design space (without genotype-phenotype mapping), using a mutation rate $p_m > 0$, this has the effect of checking type B robustness to a certain extend. An application for finding robust electronic designs on field-programmable gate arrays (FPGA) using an $(1+1)$-ES has been presented in [160].

5. Remarks on performance aspects and test scenarios

In the former sections, we have discussed the different (algorithmic) techniques that can be used for robust and noisy optimization. While the methods from mathematical programming provide guarantees for solution quality and efficiency at least partially, their applicability is rather restricted. At the same time, the deterministic numerical and randomized approaches to robust optimization and especially direct search methods such as evolutionary algorithms allow the flexible incorporation of direct robustness tests. However, for all of those techniques reviewed in the last section usually no theoretical (finite time) results with regard to quality and convergence are available. Therefore, there is a pressing need to evaluate and compare the different techniques – a research topic of increasing importance. In the following sections, we will review and discuss some of these aspects.

5.1. Performance of direct search algorithms

In this survey, we have reviewed the different approaches to finding robust optimal designs. When facing a new design task the natural question arises, however, which of the approaches should be used and which methods are especially well suited to a specific design optimization task. It is quite clear that we cannot provide a definitive answer here. Actually, this would not even be possible for ordinary design optimization. Here we will only discuss some implications for robust optimization where the quality of a design must be evaluated by simulation programs. That is, we are acting in a *black-box* scenario, similar to Fig. 1. Under such conditions, direct optimization techniques should be considered at the first place. Whereas, the deterministic approaches, presented in Section 4.2.1, can only be applied to a restricted class of problems (mostly problems which can be quadratically approximated using Taylor expansion and with normally distributed uncertainties).

However, even when sticking to the direct search techniques, one does not find any useful guideline in literature as to the question which of the methods should be used in robust optimization. Actually, leaving aside any computational performance aspects, for most of the direct search algorithms proposed in literature, we even do not know what the algorithms' output will be given a sufficient long running time. While there is a certain guarantee for *stochastic approximation* algorithms to converge to a local minimizer of $\mathrm{E}[f|\mathbf{x}]$ (under strong restrictions on the class of functions $f(\mathbf{x})$ and $t \to \infty$), predictions of the solution quality of optimization strategies on noisy functions are hard to find. Apart from the performance considerations for $(\mu/\mu_I, \lambda)$-ES, mentioned in Section 4.2.3, there seems to be no further theoretical investigations. This makes performance comparison of different algorithms a serious problem.

However, sound empirical performance comparisons using a test bed of robust design problems are also spare. There are some empirical investigations concerning ordinary optimization problems superimposed with noise, i.e., for type C uncertainties. These investigations usually aim at showing that a respective algorithm performs better (one might add with a wink: usually the one designed by the respective authors) than some others on a set of test problems, e.g., [118,128–130]. In [161] a performance comparison of the $(\mu/\mu_I, \lambda)$-ES with other direct search strategies[21] on the quadratic sphere model, Eq. (61) where $\mathbf{Q} = \mathbf{I}$, with additive normally distributed proportionate noise has been given showing the superiority of the ES on this special test function. While one can argue that the sphere model is overly simple, it does allow for a deeper understanding of the working principles of the different algorithms. Furthermore, the scaling behavior w.r.t. the problem dimension and the noise strength can be evaluated. Therefore, we argue that performance comparisons should start with such simple test functions. Considering more complex test functions – as usually done in EA-literature on robust optimization, e.g., [153,53,60] – always bears the risk that the results obtained are very specific and cannot be generalized. This empirical analysis should be the second step and particular care must be taken to aim at analyzing a certain class of functions, e.g., a number of different design problems instead of just one.

[21] Strategies that have been considered: *direct pattern search* [162], *multi-directional search* [163], *implicit filtering* [164].

There is clearly a need for a set of *scalable* test functions and corresponding robust optimization problems that could serve as a library for testing the different optimization algorithms. Such a test library should contain a collection of problems of different "complexities". A possible taxonomy of test scenarios could be:

(a) Scalable robust optimization problems with known analytical solutions. The analysis should provide estimates on expected solution quality, or even better, estimates on the expected runtime for reaching a solution of a certain quality. Results of the former type have already been obtained for ES with actuator noise (discussed above). In the next section, some candidates for this problem class will be presented.
(b) Scalable robust optimization problems amenable to constrained linear or quadratic programming as discussed in Section 4.1. Since the (nearly) exact robust solutions are known, the solution quality of the direct optimization strategies could be assessed easily.
(c) Test problems gleaned from real-world robust design tasks of different complexity, easy to program and without any ambiguities. These problems should be "typical" in the sense that they also represent major characteristics of a problem class. Clearly, agreeing on the members in this class is most difficult. Also, the real robust optimum will often not be known such that one can only rely on relative evaluations.

In the following, we will consider class (a) test functions only. The other two classes are empty up until now. No publications on these topics have been found.

5.2. Class (a) test scenarios

5.2.1. The quadratic sphere and general quadratic models

The quadratic N-dimensional sphere

$$f_{\mathrm{s}}(\mathbf{x}) := \|\mathbf{x}\|^2 = \sum_{i=1}^{N} x_i^2 \tag{72}$$

is probably the most simple *scalable* test function firstly considered in context of type B uncertainties in [49]. Optimization goal is the minimization. One can easily determine the robust counterparts F. For the case of robust regularization (9) using the neighborhood $\mathscr{X} = \{\boldsymbol{\xi} : \|\boldsymbol{\xi} - \mathbf{x}\| \leqslant \epsilon\}$, i.e., an Euclidean ball around the design point $\mathbf{x}$, one gets

$$\begin{aligned} F_{\mathrm{s}B}(\mathbf{x},\epsilon) &= \sup_{\|\boldsymbol{\delta}\|\leqslant\epsilon} (\mathbf{x}+\boldsymbol{\delta})^2 \\ &= \sup_{\|\boldsymbol{\delta}\|\leqslant\epsilon} \left(\|\mathbf{x}\|^2 + 2\|\mathbf{x}\|\|\boldsymbol{\delta}\| \cos(\measuredangle(\mathbf{x},\boldsymbol{\delta})) + \|\boldsymbol{\delta}\|^2 \right). \end{aligned} \tag{73}$$

Taking into account that the cosine assumes its maximum for $\mathbf{x} \| \boldsymbol{\delta}$ and $\|\boldsymbol{\delta}\| \leqslant \epsilon$, one immediately obtains

$$F_{\mathrm{s}B}(\mathbf{x},\epsilon) = (\|\mathbf{x}\|^2 + 2\|\mathbf{x}\|\epsilon + \epsilon^2) = (\|\mathbf{x}\| + \epsilon)^2 \tag{74}$$

and the robust minimizer is at $\hat{\mathbf{x}} = \mathbf{0}$.

Considering the first momentum robustness measure (25), one finds

$$\begin{aligned} F_{\mathrm{s}1}(\mathbf{x}) &= \mathrm{E}[f_{\mathrm{s}}(\mathbf{x}+\boldsymbol{\delta})|\mathbf{x}] = \mathrm{E}[\mathbf{x}^2 + 2\mathbf{x}^{\mathrm{T}}\boldsymbol{\delta} + \boldsymbol{\delta}^2|\mathbf{x}] \\ &= \mathbf{x}^2 + 2\mathbf{x}^{\mathrm{T}}\mathrm{E}[\boldsymbol{\delta}] + \mathrm{E}[\boldsymbol{\delta}^2]. \end{aligned} \tag{75}$$

Assuming for the $\boldsymbol{\delta}$ uncertainties $\mathrm{E}[\boldsymbol{\delta}] = \mathbf{0}$, the robust minimizer is at $\hat{\mathbf{x}} = \mathbf{0}$. The special case $\boldsymbol{\delta} = \mathscr{N}(\mathbf{0}, \epsilon^2\mathbf{I})$ leads to

$$F_{\mathrm{s}1}(\mathbf{x}) = \|\mathbf{x}\|^2 + N\epsilon^2. \tag{76}$$

Similarly, one finds for the variance in the $\boldsymbol{\delta} = \mathscr{N}(\mathbf{0}, \epsilon^2\mathbf{I})$ case [49]

$$\mathrm{Var}[f_{\mathrm{s}}|\mathbf{x}] = 4\epsilon^2\|\mathbf{x}\|^2 + 2N\epsilon^4. \tag{77}$$

In this special case, minimizing the first moment and the variance are not conflicting goals and the minimizer is $\hat{\mathbf{x}} = \mathbf{0}$.

The probabilistic threshold measure (31) can also be applied. As has been shown in [49],

$$\begin{aligned} F_{sq}(\mathbf{x}) &= \Pr[f_{\mathrm{s}} \leqslant q|\mathbf{x}] \\ &= \Pr\left[\epsilon^2 \sum_{i=1}^{N} \left(\frac{x_i}{\epsilon} + \mathscr{N}(0,1) \right)^2 \leqslant q \middle| \mathbf{x} \right] \to \max \\ &= \Pr[\chi'^2_N(\zeta) \leqslant t|\mathbf{x}] \to \max. \end{aligned} \tag{78}$$

The random variate $\chi'^2_N(\zeta)$ in (78) obeys a non-central χ^2 distribution with N degrees of freedom, non-centrality parameter $\zeta = \mathbf{x}^2/\epsilon^2$, and threshold $t = q/\epsilon^2$. One can prove for arbitrary thresholds $q > 0$ that the global minimizer is at $\hat{\mathbf{x}} = \mathbf{0}$.

The performance of the $(\mu/\mu_I, \lambda)$-ES on the quadratic sphere with normally distributed actuator noise has been investigated in [49,165]. The steady state performance (final fitness error) appears now as a special case $\mathbf{Q} = \mathbf{I}$ in (70) valid for the general quadratic model (68). Since we have already considered (68), we will not explicitly discuss it here, but only note that the *general quadratic model* (68) should definitely belong to this class of test functions.

5.2.2. The sphere model with environmental uncertainties and the variance problem

This model presents an example for type A uncertainties

$$f_{\mathrm{e}}(\mathbf{x},\alpha) := a - (\alpha+1)\|\mathbf{x}\|^{\beta} + b\alpha, \quad \beta > 0,\ \alpha \in \mathbb{R}. \tag{79}$$

It is the task to robustly *maximize* (79). Note, this test function is similar to (28) where minimization has been considered.

The robust regularization approach of Section 3.2.1, can be easily transferred to robust *maximization*. In this case the robust counterpart must be determined as the *infimum* of $f_{\mathrm{e}}(\mathbf{x},\alpha)$. Using an $|\alpha| \leqslant \epsilon$ neighborhood, we get

$$\begin{aligned} F_{\mathrm{e}B}(\mathbf{x},\epsilon) &= \inf_{|\alpha|\leqslant\epsilon} (f_{\mathrm{e}}(\mathbf{x},\alpha)) = \inf_{\alpha\in[-\epsilon,\epsilon]} (a - \|\mathbf{x}\|^{\beta} + (b - \|\mathbf{x}\|^{\beta})\alpha) \\ &= \begin{cases} a - \|\mathbf{x}\|^{\beta} - (b - \|\mathbf{x}\|^{\beta})\epsilon, & \text{if } \|\mathbf{x}\|^{\beta} \leqslant b, \\ a - \|\mathbf{x}\|^{\beta} + (b - \|\mathbf{x}\|^{\beta})\epsilon, & \text{if } \|\mathbf{x}\|^{\beta} > b. \end{cases} \end{aligned} \tag{80}$$

Rearranging terms, this reads

$$F_{eB}(\mathbf{x},\epsilon)=\begin{cases} a-\epsilon b-(1-\epsilon)\|\mathbf{x}\|^{\beta}, & \text{if } \|\mathbf{x}\|^{\beta}\leqslant b,\\ a+\epsilon b-(1+\epsilon)\|\mathbf{x}\|^{\beta}, & \text{if } \|\mathbf{x}\|^{\beta}>b. \end{cases} \tag{81}$$

Visual inspection of the robust counterpart (81) reveals that the robust maximum of $f_e(\mathbf{x},\alpha)$ is given by

$$\max_{\mathbf{x}} F_{eB}(\mathbf{x},\epsilon)=\begin{cases} a-\epsilon b, & \text{if } 0\leqslant\epsilon\leqslant 1,\\ a-b, & \text{if } \epsilon>1 \end{cases} \tag{82}$$

and the optimizer obeys

$$\|\hat{\mathbf{x}}\|^{\beta}=\arg\max_{\|\mathbf{x}\|^{\beta}} F_{eB}(\mathbf{x},\epsilon)=\begin{cases} 0, & \text{if } 0\leqslant\epsilon\leqslant 1,\\ b, & \text{if } \epsilon>1. \end{cases} \tag{83}$$

That is, for $0\leqslant\epsilon\leqslant 1$ it holds $\hat{\mathbf{x}}=\mathbf{0}$; for $\epsilon>1$ all designs $\hat{\mathbf{x}}$ which fulfill $\|\hat{\mathbf{x}}\|=b^{1/\beta}$ are optimal; and for $\epsilon=1$ all $\hat{\mathbf{x}}$ which fulfill $\|\hat{\mathbf{x}}\|<b^{1/\beta}$ are optimal.[22] For $\epsilon>1$ we encounter an example where the optimizer of the robust counterpart does *not* coincide with that of the original test function (79) with vanishing uncertainties.

Assuming stochastic α uncertainties, one can alternatively consider the first moment (23) of f_e and the variance measure (26) of Section 3.2.2. Rewriting (79) as $f_e(\mathbf{x},\alpha)=a-\|\mathbf{x}\|^{\beta}+(b-\|\mathbf{x}\|^{\beta})\alpha$, one immediately sees that (provided that $\mathrm{E}[\alpha]=0$)

$$F_{e1}(\mathbf{x})=\mathrm{E}[f_e|\mathbf{x}]=a-\|\mathbf{x}\|^{\beta} \tag{84}$$

and

$$\mathrm{Var}[f_e|\mathbf{x}]=(b-\|\mathbf{x}\|^{\beta})^2\mathrm{Var}[\alpha]. \tag{85}$$

In contrast to the robust regularization (81), there is an optimum of F_{e1} at $\hat{\mathbf{x}}=\mathbf{0}$ *independent* of the choice of α. For example, assuming $\alpha\sim\mathcal{N}(0,\epsilon^2)$, the standard deviation ϵ of the uncertainty α has no influence on the location of the optimum. However, considering the location of the minimum of the variance (85), one sees that this minimum is at those designs $\hat{\mathbf{x}}$ for which $\|\hat{\mathbf{x}}\|=b^{1/\beta}\neq 0$. Therefore, maximizing performance and minimizing the variance are two conflicting goals on this test function. Interestingly, the states with minimal variance agree with those of the robust regularization with $\epsilon>1$ in (83). This appears reasonable – to a certain extend – since the worst case scenario considers large deviations from the nominal design, conversely the variance may be regarded as quantity measuring these deviations.

As the third robustness measure, we consider the probabilistic threshold measure (33), (34) assuming $\alpha\sim\mathcal{N}(0,\epsilon^2)$. After a short calculation one obtains (see, [49])

$$F_{eq}(\mathbf{x})=\Pr[f_e\leqslant q|\mathbf{x}]=\Phi\left(\frac{q-a+\|\mathbf{x}\|^{\beta}}{\epsilon|b-\|\mathbf{x}\|^{\beta}|}\right)\to\min, \tag{86}$$

where $\Phi(y)$ is the cdf (cumulative distribution function) of the standard Gaussian variate

$$\Phi(y)=\frac{1}{\sqrt{2\pi}}\int_{t=-\infty}^{t=y}\mathrm{e}^{-\frac{1}{2}t^2}\,\mathrm{d}t=\frac{1}{2}\left(1+\mathrm{erf}\left(\frac{y}{\sqrt{2}}\right)\right). \tag{87}$$

The robustness measure (86) is displayed in Fig. 6. As one can see, this robustness measure does not provide a conclusive and simple decision rule how to choose $\hat{\mathbf{x}}$. While there is a certain bias to $\hat{\mathbf{x}}=\mathbf{0}$ for large q thresholds, there is also a region where the local minimum of Pr is at $\|\hat{\mathbf{x}}\|^{\beta}=b$. Considering (86), this is the case for $q<a-\|\hat{\mathbf{x}}\|^{\beta}$. One also sees that for such q values there is a plateau-like region of nearly optimal $\mathbf{x}$ values. Since it is difficult to draw general conclusions as to the choice of $\hat{\mathbf{x}}$ on this test function independent of q, we stop considering this measure for now.

Even though test function (79) is a simple sphere, it possesses a rich repertoire of properties. This also holds for the behavior of the $(\mu/\mu_I,\lambda)$-ES on this function. Provided that $\alpha\sim\mathcal{N}(0,\epsilon^2)$ one can show [49] that the expected steady state fitness error $\mathrm{E}[\Delta\tilde{f}]$ obeys

$$\mathrm{E}[\Delta\tilde{f}]\geqslant\frac{\epsilon Nb}{\epsilon N+2\beta\mu c_{\mu/\mu,\lambda}}. \tag{88}$$

That is, assuming a constant truncation ratio μ/λ (recall: μ – number of parents, λ – number of offspring), the robust optimum a of F_{e1}, Eq. (84), can be approached arbitrarily close by increasing the population size λ. The investigations in [49] also showed that there are parameter combinations of N, b, β, ϵ, μ, and λ for which the convergence/divergence of the ES depends on the initial distance $\mathbf{x}$ to the optimizer $\hat{\mathbf{x}}=\mathbf{0}$.

As Eq. (88) shows, the ES optimizes the first momentum measure. If one is interested in minimizing the variance (regardless of the expected value of f), the ES does *not* provide a direct approach. This holds for all test functions investigated so far (except those cases where the optimizer of the expected value and the variance are the same). This property is similar to that of the *stochastic approximation*

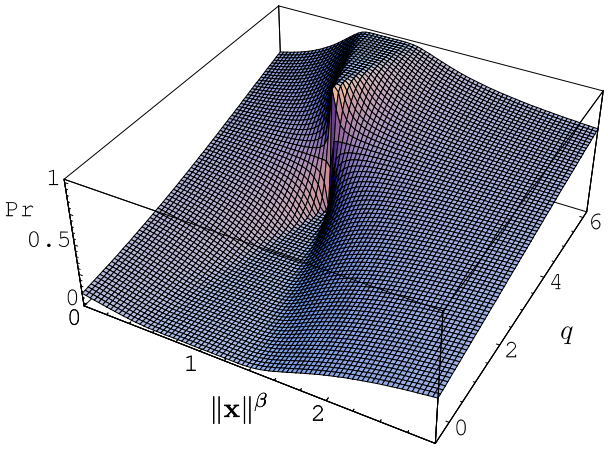

Fig. 6. Probabilistic robustness measure of test function (79) with $a=5$, $b=1$, $\beta=1$, and $\epsilon=4$. Given a threshold q, those $\|\mathbf{x}\|^{\beta}$ are desirable for which Pr is small.

[22] *Remark*: It would be interesting to analyze the behavior of an EA where the fitness function performs a minimum search over κ samples. Such a type of EA has been proposed in [36]. Using the test function (79) one might have a chance to obtain theoretical results.

algorithms. It seems that there is no direct way of evolving variance-minimal solutions using a population of λ *single* $f(\mathbf{x}_l, \alpha)$ samples. A variance estimate can be obtained by sampling κ times each single design $\mathbf{x}_l$. That is, one needs at least $\kappa = 2$ samples from each design $\mathbf{x}_l$.

5.2.3. Functions with noise-induced multi-modality

In [166] a new class of test functions has been proposed, motivated from design optimization of gas-turbine blades. Unlike the test functions considered so far, the newly proposed function class exhibits a (smooth) topology change of the mean value landscape depending on the regularization (or uncertainty) parameter ϵ.

5.2.3.1. Function "f_2". As a first example we consider the robust maximization of

$$f_2(\mathbf{x}, \alpha) := a - \frac{(x_{N-1} + \alpha)^2 + \sum_{i=1}^{N-2} x_i^2}{x_N^2 + b} - x_N^2, \quad b > 0, \ \mathbf{x} \in \mathbb{R}^N \tag{89}$$

with type A uncertainties. The derivation of the robust counterparts will only be sketched here, for an in-depth treatment, the reader is referred to [167].

Let us first calculate the robust regularization function $F_{2\mathrm{B}}$. In the worst case scenario assuming $|\alpha| \leqslant \epsilon$ one has

$$\begin{aligned} F_{2\mathrm{B}}(\mathbf{x}, \epsilon) &= \inf_{\alpha \in [-\epsilon, \epsilon]} (f_2(\mathbf{x}, \alpha)) \\ &= a - x_N^2 - \frac{\sum_{i=1}^{N-2} x_i^2}{x_N^2 + b} - \frac{1}{x_N^2 + b} \max_{\alpha \in [-\epsilon, \epsilon]} ((x_{N-1} + \alpha)^2). \end{aligned} \tag{90}$$

Since

$$\max_{\alpha \in [-\epsilon, \epsilon]} ((x_{N-1} + \alpha)^2) = \begin{cases} (x_{N-1} + \epsilon)^2, & \text{if } x_{N-1} \geqslant 0, \\ (x_{N-1} - \epsilon)^2, & \text{if } x_{N-1} < 0, \end{cases} = (|x_{N-1}| + \epsilon)^2, \tag{91}$$

one gets the robust counterpart

$$F_{2\mathrm{B}}(\mathbf{x}, \epsilon) = a - \frac{(|x_{N-1}| + \epsilon)^2 + \sum_{i=1}^{N-2} x_i^2}{x_N^2 + b} - x_N^2. \tag{92}$$

In order to maximize (92) one sees that $\forall i = 1, \ldots, N-1$: $\hat{x}_i = 0$ must hold. The determination of the *local* optimizer $\check{x}_N$ of $F_{2\mathrm{B}}(\mathbf{x}, \epsilon)$ for arbitrary x_i $(i < N)$ can be done using calculus

$$\check{x}_N = \begin{cases} 0, & \text{if } \sqrt{(|x_{N-1}| + \epsilon)^2 + \sum_{i=1}^{N-2} x_i^2} \leqslant b, \\ \pm \sqrt{\sqrt{(|x_{N-1}| + \epsilon)^2 + \sum_{i=1}^{N-2} x_i^2} - b}, & \text{otherwise.} \end{cases} \tag{93}$$

Inserting $\hat{x}_i = 0$ (for $i < N$) one gets from (93) the global optimizer

$$\hat{x}_N = \begin{cases} 0, & \text{if } \epsilon \leqslant b, \\ \pm\sqrt{\epsilon - b}, & \text{if } \epsilon > b \end{cases} \tag{94}$$

and therefore

$$\hat{\mathbf{x}} = \begin{cases} \mathbf{0}, & \text{if } \epsilon \leqslant b, \\ (0, \ldots, 0, \pm\sqrt{\epsilon - b})^{\mathrm{T}}, & \text{if } \epsilon > b. \end{cases} \tag{95}$$

While for $\epsilon \leqslant b$ there is a maximum at $\mathbf{x} = \mathbf{0}$, there emerge *two maxima* for $\epsilon > b$. That is, depending on the strength ϵ of the uncertainty, the topology of the robust counterpart changes from a unimodal to a bi-modal landscape. The maximum value is obtained by inserting (95) into (92)

$$\hat{F}_{2\mathrm{B}} = \begin{cases} a - \frac{\epsilon^2}{b}, & \text{if } \epsilon \leqslant b, \\ a + b - 2\epsilon, & \text{if } \epsilon > b. \end{cases} \tag{96}$$

The bifurcation behavior observed in (94) is not a peculiarity of the regularization approach used, it also appears when considering the momentum robustness measure. Assuming $\alpha \sim \mathcal{N}(0, \epsilon^2)$, the expected value measure (23) can be easily calculated

$$F_{21}(\mathbf{x}, \epsilon) = \mathrm{E}[f_2|\mathbf{x}] = a - \frac{r^2 + \epsilon^2}{x_N^2 + b} - x_N^2, \tag{97}$$

where

$$r := \sqrt{\sum_{i=1}^{N-1} x_i^2}. \tag{98}$$

Using the aggregated quantity r, the robust counterpart of f_2 can be displayed in Fig. 7. For small levels ϵ of uncertainties the robust counterpart landscape possesses a unique maximum. From the graph on the left-hand side one gets the optimal design parameter setting $\hat{r} = 0$, $\hat{y}_N = 0$, i.e., $\hat{\mathbf{y}} = \mathbf{0}$. This solution is *robust* with respect to the change of the noise strength ϵ. However, when exceeding a certain level, the mean fitness landscape changes dramatically as can be seen in the right-hand graph of Fig. 7. While $r = 0$ still remains the optimal setting, $y_N = 0$ is no longer optimal. Remarkably, the maximum at $y_N = 0$ has changed to a minimum. The analysis can be found in [166]. It yields *exactly* the same result as in the case of the robust worst case regularization, interpreting ϵ in (95) and (96) as standard deviation of the α variate.[23]

The robust optimization of f_2, Eq. (89), using the probabilistic threshold measure (33), (34) assuming $\alpha \sim \mathcal{N}(0, \epsilon^2)$ has been considered in [167]. It has been shown that in general the global robust optimizer $\hat{\mathbf{x}}$

$$\hat{\mathbf{x}} = \begin{cases} \mathbf{0}, & \text{if } a - q \leqslant b, \\ (0, \ldots, 0, \pm\sqrt{(a - b - q)/2})^{\mathrm{T}}, & \text{if } a - q > b \end{cases} \tag{99}$$

depends on the threshold q, but not on the strength of the uncertainty ϵ. The critical point here is that the bifurcation behavior of the x_N optimizer depends on the threshold value q chosen. This first comes as a surprise, however, considering the freedom in choosing q one can choose $q := a + b - 2\epsilon$. If inserted in (99) one recovers (95).

[23] Note, α need not be a Gaussian normal variate. $\mathrm{E}[\alpha] = 0$ is the only condition on α that must be fulfilled in order to yield the results.

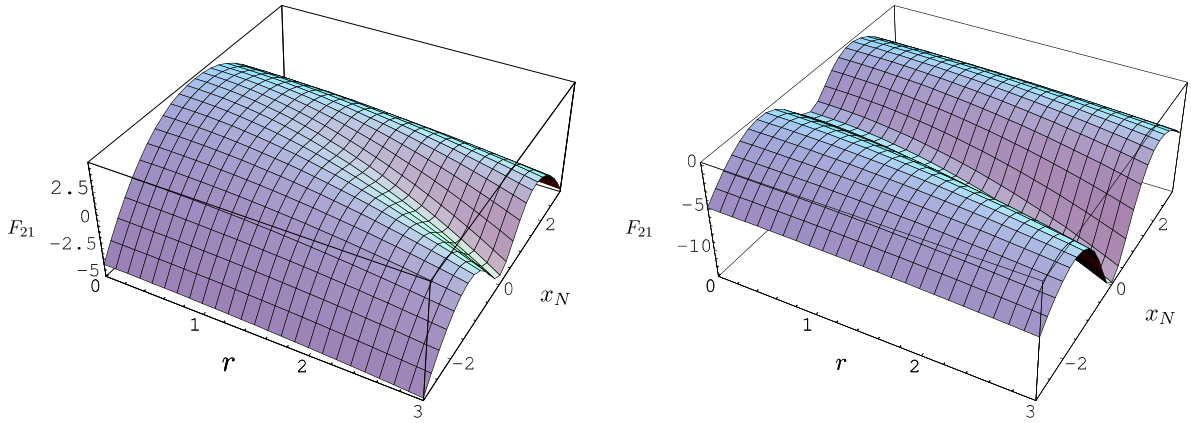

Fig. 7. Expected value landscapes of f_2 given by Eqs. (97) and (98) for $\epsilon = 0.25$ (left) and $\epsilon = 3.0$ (right) ($a = 5$, $b = 1$).

5.2.3.2. Variations on f_2. A generalization of f_2 has been presented in [166]. Using our notations, the test function f_3 reads

$$f_3(\mathbf{x},\boldsymbol{\alpha}) := a - \frac{\sum_{i=1}^{N_1-1} x_i^2 + \sum_{i=N_1}^{N_2-1}(x_i+\alpha_i)^2}{b + \sum_{i=N_2}^{N} x_i^2} - \sum_{i=N_2}^{N} x_i^2, \quad b > 0, \tag{100}$$

where

$$1 \leqslant N_1 < N_2 \leqslant N. \tag{101}$$

Function f_2 is contained in (100) for $N_1 = N-1$ and $N_2 = N$. Considering the expectancy robustness measure, one can easily calculate the robust counterpart (23). Assuming $\mathrm{E}[\alpha_i] = 0$ and $\mathrm{Var}[\alpha_i] = \epsilon^2$ one obtains for the robust optimizer (see [167])

$$\hat{\mathbf{x}} = \begin{cases} \mathbf{0}, & \text{if } \epsilon \leqslant b/\sqrt{N_2 - N_1}, \\ (0,\ldots,0,\hat{x}_{N_2},\ldots,\hat{x}_N)^{\mathrm{T}}, & \text{if } \epsilon > b/\sqrt{N_2 - N_1}, \end{cases} \tag{102}$$

where the $\hat{x}_{N_2},\ldots,\hat{x}_N$ are located on a sphere

$$\sum_{i=N_2}^{N} \hat{x}_i^2 = \epsilon\sqrt{N_2 - N_1} - b. \tag{103}$$

That is, if the uncertainty strength ϵ exceeds the limit $b/\sqrt{N_2 - N_1}$, the robust counterpart function $F_{31}(\mathbf{x},\epsilon)$ changes from a unimodal function to a multi-modal function of $\mathbf{x}$. This is referred to as *noise-induced multi-modality*. If $N_2 < N$ then there is an $(N - N_2)$-dimensional manifold of optimal robust designs $\hat{\mathbf{x}}$. The maximum of the robust counterpart $F_{31}(\mathbf{x},\epsilon)$ is given by

$$\hat{F}_{31} = \begin{cases} a - \epsilon^2(N_2 - N_1)/b, & \text{if } \epsilon \leqslant b/\sqrt{N_2 - N_1} \\ a + b - 2\epsilon\sqrt{N_2 - N_1}, & \text{if } \epsilon > b/\sqrt{N_2 - N_1}. \end{cases} \tag{104}$$

5.2.3.3. Function f_4. Predicting the behavior of the $(\mu/\mu_I,\lambda)$-ES on test functions f_2 and f_3 still remains a challenge. Using a very rough model it has been shown in [166] that one can explain the *qualitative* behavior of the ES on f_2. However, this does not suffice for theoretical performance evaluations which are expected to be exact in the asymptotical limit of infinite problem size N. In order to fill this gap, f_4 has been proposed in [167]. It is defined as a special case of f_3, Eq. (100), using $N_1 = 1$ and $N_2 = N$, i.e.,

$$f_4(\mathbf{x},\boldsymbol{\alpha}) := a - \frac{\sum_{i=1}^{N-1}(x_i+\alpha_i)^2}{b + x_N^2} - x_N^2, \quad b > 0. \tag{105}$$

Assuming $\alpha_i \sim \mathcal{N}(0,\epsilon^2)$, the steady state behavior of the $(\mu/\mu_I,\lambda)$-ES has been analyzed in [167]. The expected quadratic deviation of the first $N-1$ components is obtained as

$$\mathrm{E}\left[\sum_{i=1}^{N-1} x_i^2\right] \simeq \frac{(N-1)^2\epsilon^2}{8\mu^2 c^2_{\mu/\mu,\lambda}}\left(1 + \sqrt{1 + \frac{8\mu^2 c^2_{\mu/\mu,\lambda}}{N-1}}\right) \tag{106}$$

and the expected steady state value of x_N

$$\mathrm{E}[x_N] \simeq \pm\sqrt{\sqrt{(N-1)\epsilon^2\left[1 + \frac{N-1}{8\mu^2 c^2_{\mu/\mu,\lambda}}\left(1 + \sqrt{1 + \frac{8\mu^2 c^2_{\mu/\mu,\lambda}}{N-1}}\right)\right]} - b}. \tag{107}$$

A closer examination of (106) and (107) shows for $\mu/\lambda = \text{const.}$ that increasing the population size λ allows for an arbitrarily exact approximation of the true robust optimizer. Furthermore, it also shows that upgrading the population of λ designs by a factor of κ is more efficient than resampling λ designs κ times.

Again the question remains, how other direct search algorithms perform on these test functions.

We have argued at the beginning of this section that due to the limited versatility of the mathematical programming approaches and the limited theoretical results for direct optimization approaches, the empirical investigation of robust optimization techniques becomes increasingly important. Besides making sure that the technical – statis-

tical – means are chosen correctly [168], the choice of the "test" problem is of utmost importance. It should represent either the characteristics of a certain practically relevant problem class or of a theoretical problem property, or be of considerable practical interest in itself. In this section, we made a first step to introduce test functions that exhibit some characteristics which seem to be shared by some practical aerodynamic design optimization problems, see [166].

6. Outlook – trends and research perspectives

In this survey, we have given a comprehensive overview over the field of robust optimization. Starting from Taguchi's robust design methodology, we first considered the robust regularization approach based on worst case scenarios usually used to find robust solutions to linear and quadratic constrained optimization problems. While this approach has a certain appeal from the viewpoint of mathematics, its usability for real-world applications is, however, very restricted. Often the evaluation of designs in practice cannot be approximated by linear or quadratic models. Furthermore, in general, one cannot expect the availability of an analytical model. In such situations one has to resort to simulation tools such as computational fluid dynamics or even directly to experiments. Emulating the design uncertainties, the output of such programs is inherently and explicitly noisy. In order to optimize a design based on noisy quality information, suitable direct optimization techniques are required. Empirically as well as principally evolutionary algorithms are direct optimization techniques that work well on noisy problems. Here "principally" refers to the ability of biological evolution – the paradigm behind evolutionary computation – to generate robust and flexible systems that are well adapted to noisy and dynamic environments.

Most design problems reported in the literature are low-dimensional, i.e., the design is represented by only a few parameters mostly of the order of ten. Under such conditions, most approaches such as response surface methodology, direct pattern search, Kriging method, etc. work (nearly) equally well. However, there is a growing number of optimization problems that require a detailed design representation and where the number of parameters increases by nearly an order of magnitude, see e.g., [169,170].

Robust design optimization using direct search algorithms is still in its beginning. Developing and analyzing optimization algorithms for noisy functions is likely to be an research area of growing interest driven by the increasing engineering demand to produce solutions whose performance is not just increased at the design point but for a wide interval of operating conditions. Research in robust design optimization is currently carried out in the fields of operations research, engineering optimization and computational intelligence (among others). Looking for synergies between the different philosophies in these fields is one promising approach towards a qualitative step in robust optimization. Furthermore, mathematical precision must be more wholeheartedly coupled to engineering pragmatism, e.g., specifying upper and lower bounds on the expected search performance is very valuable even for the practitioner.

Finally, there are some more short term research questions that have to be tackled in the near future to make robust optimization the rule and not the exception in engineering design:

- What kind of direct search technique should be used for which problem class? In order to be able to provide some answer to this question, we have to carefully set up useful test cases and test problems for robust design and analyze them in detail.
- We must extend our knowledge on expected performance indices by direct search techniques. It is likely that this is most practically achieved by theoretical studies of simplified version of the test problems that we have defined above possibly resulting in upper and lower bounds for the true test cases.
- How is performance in general related to robustness and are there additional means to achieve robustness through adaptivity during operation. It is likely that in many cases maximal performance at the design point is in contradiction to robustness over a wide interval of operating conditions. Thus, robust design is inherently a multi-criteria optimization problem. Research in this direction is just beginning [60].
- How can constraint handling be efficiently coupled to robustness and how do both quality measures interact both theoretically as well as empirically on a set of characteristic benchmark problems.

Of course this list is by no means exhaustive. There is a large variety of open problems giving rise to exciting research topics which should be considered next.

Appendix A. The $(\mu/\mu_I, \lambda)$ evolution strategy

The $(\mu/\mu_I, \lambda)$-ES [28,34] is a simple but very effective EA for optimization in real-valued search spaces (i.e., the space of the design variables $\mathbf{x}$). It uses a population of μ parental individuals $\mathbf{x}_m$ and generates a population of λ offspring $\tilde{\mathbf{x}}_l$ $(l = 1, \ldots, \lambda)$ by *intermediate multirecombination* and *mutation*. Intermediate multirecombination is performed by centroid calculation. That is, given a set of μ vectors $\mathbf{a}_m$ (or scalars), the centroid $\langle \mathbf{a} \rangle$ is defined as

$$\langle \mathbf{a} \rangle := \frac{1}{\mu} \sum_{m=1}^{\mu} \mathbf{a}_m. \tag{A.1}$$

On top of the centroid $\langle \mathbf{x} \rangle$ a mutation is applied by adding a(n) (isotropic[24]) normally distributed random vector

[24] Using isotropic mutations represents the simplest form expressing unbiasedness in the search space. There are also more elaborate ES versions using correlated Gaussian mutations [171,172], however, these are not considered in this introductory appendix.

$\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. σ is referred to as the *mutation strength*. The mutation process is performed λ times producing a set of λ offspring individuals with design (or objective) parameters $\tilde{\mathbf{x}}_l$ and fitness values (i.e., objective function values) $f(\tilde{\mathbf{x}}_l)$. In order to obtain a new parent population selection must be applied. This is done by (μ, λ)-truncation selection taking the top μ best individuals (w.r.t. the measured fitness) as parents for the next generation. In order to refer to the mth best individual, we use the "$m;\lambda$" notation.

The efficient working of the ES under various conditions strongly depends on a sophisticated control of the mutation strength σ during the evolution. There are two standard approaches to σ-control: the σ self-adaptation [173,34] and alternatively the cumulative step size adaptation [174,171].

The σ self-adaptation technique is based on the coupled inheritance of design and σ parameters. Using the notation

$$\langle \mathbf{a} \rangle^{(g)} := \frac{1}{\mu} \sum_{m=1}^{\mu} \mathbf{a}_{m;\lambda}^{(g)} \tag{A.2}$$

for intermediate recombination, where (g) is the generation counter, the $(\mu/\mu_I, \lambda)$-σSA-ES can be expressed in "offspring notation"

$$\forall l = 1, \ldots, \lambda : \begin{cases} \tilde{\sigma}_l^{(g+1)} := \langle \tilde{\sigma} \rangle^{(g)} \mathrm{e}^{\tau \mathcal{N}_l(0,1)}, \\ \tilde{\mathbf{x}}_l^{(g+1)} := \langle \tilde{\mathbf{x}} \rangle^{(g)} + \tilde{\sigma}_l^{(g+1)} \mathcal{N}_l(\mathbf{0}, \mathbf{I}). \end{cases} \tag{A.3}$$

That is, each offspring individual (indexed by l) gets its own mutation strength $\tilde{\sigma}$. And this mutation strength is used as mutation parameter for producing the offspring's design parameter(s). In (A.3) we used the log-normal update rule for mutating the mutation strength. The learning parameter τ is usually chosen as $\tau = 1/\sqrt{N}$, where N is the search space (design space) dimensionality.

While in evolutionary self-adaptive ES each individual get its own set of endogenous strategy parameters, *cumulative step size adaptation* uses a single mutation strength parameter σ per generation to produce all the offspring. This σ is updated by a deterministic rule which is controlled by certain statistics gathered over the course of generations. The statistics used is the so-called (normalized) cumulative path-length $\mathbf{s}$. If $\|\mathbf{s}\|$ is greater than the expected length of a random path, σ is increased. In the opposite situation, σ is decreased. The update rule reads

$$\left.\begin{aligned} &\forall l = 1, \ldots, \lambda : \tilde{\mathbf{x}}_l^{(g+1)} := \langle \tilde{\mathbf{x}} \rangle^{(g)} + \sigma^{(g)} \mathcal{N}_l(\mathbf{0}, \mathbf{I}), \\ &\mathbf{s}^{(g+1)} := (1-c)\mathbf{s}^{(g)} + \sqrt{(2-c)c}\,\frac{\sqrt{\mu}}{\sigma^{(g)}} \left(\langle \tilde{\mathbf{x}} \rangle^{(g+1)} - \langle \tilde{\mathbf{x}} \rangle^{(g)} \right), \\ &\sigma^{(g+1)} := \sigma^{(g)} \exp\left(\frac{\|\mathbf{s}^{(g+1)}\| - \overline{\chi}_N}{D \overline{\chi}_N} \right), \end{aligned}\right\} \tag{A.4}$$

where $\mathbf{s}^{(0)} = \mathbf{0}$ is chosen initially. The recommended standard settings for the cumulation parameter c and the damping constant D are $c = 1/\sqrt{N}$ and $D = \sqrt{N}$. For the expected length of a random vector comprising N standard normal components, the approximation $\overline{\chi}_N = \sqrt{N}(1 - 1/4N + 1/21N^2)$ can be used.

Appendix B. The progress coefficient $c_{\mu/\mu,\lambda}$

The progress coefficient $c_{\mu/\mu,\lambda}$ is defined as the expected value of the average over the μ largest samples out of a population of λ random samples from the standard normal distribution. According to [102, p. 247], $c_{\mu/\mu,\lambda}$ can be expressed by a single integral

$$c_{\mu/\mu,\lambda} = \frac{\lambda - \mu}{2\pi} \binom{\lambda}{\mu} \int_{-\infty}^{\infty} \mathrm{e}^{-t^2} (\Phi(t))^{\lambda-\mu-1} (1 - \Phi(t))^{\mu-1} \, \mathrm{d}t, \tag{B.1}$$

where $\Phi(t)$ is the cumulative distribution function (cdf) of the standard normal variate.

In [102, p. 249] an asymptotically exact $c_{\mu/\mu,\lambda}$ expression has been derived. For $\mu, \lambda \to \infty$ and truncation ratios $0 < \mu/\lambda \leqslant 1$, one finds

$$c_{\mu/\mu,\lambda} \simeq \frac{\lambda}{\mu} \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2} \left(\Phi^{-1}\left(1 - \frac{\mu}{\lambda}\right) \right)^2 \right], \tag{B.2}$$

where Φ^{-1} is the inverse to the cumulative distribution function Φ.

References

[1] M. Walker, R. Hamilton, A technique for optimally designing fibre-reinforced laminated plates with manufacturing uncertainties for maximum buckling strength, Engrg. Optim. 37 (2) (2005) 135–144.

[2] P. Yan, M. Zhou, A life cycle engineering approach to development of flexible manufacturing systems, IEEE Trans. Robot. Automat. 19 (3) (2003) 465–473.

[3] W. Wanyana, A. Ertas, H.-C. Zhang, S. Ekwaro-Osire, Life cycle engineering: issues, tools and research, Int. J. Comput. Integrat. Manuf. 16 (4–5) (2003) 307–316.

[4] J. Marczyk, Stochastic multidisciplinary improvement: beyond optimization, American Institute of Aeronautics and Astronautics, AIAA-2000-4929.

[5] R. Suri, K. Otto, Manufacturing system robustness through integrated modeling, J. Mech. Des. 123 (4) (2001) 630–636.

[6] M. Kalsi, K. Hacker, K. Lewis, A comprehensive robust design approach for decision trade-offs in complex systems design, J. Mech. Des. 123 (1) (2001) 1–10.

[7] X. Du, Y. Wang, W. Chen, Methods for robust multidisciplinary design, Tech. Rep. 2000-1785, American Institute of Aeronautics and Astronautics, AIAA, 2000.

[8] J. Allen, C. Seepersad, H. Choi, F. Mistree, Robust design for multiscale and multidisciplinary applications, J. Mech. Des. 128 (4) (2006) 832–843.

[9] J. Birge, F. Louveaux, Introduction to Stochastic Programming, Springer-Verlag, New York, 1997.

[10] J. Mulvey, R. Vanderbei, S. Zenios, Robust optimization of large-scale systems, Oper. Res. 43 (2) (1995) 264–281.

[11] D. Bai, T. Carpenter, J. Mulvey, Making a case for robust optimization models, Manage. Sci. 43 (7) (1997) 895–907.

[12] G. Taguchi, Quality Engineering through Design Optimization, Kraus International Publications, New York, 1984.

[13] G.-J. Park, T.-H. Lee, K. Lee, K.-H. Hwang, Robust design: an overview, AIAA J. 44 (1) (2006) 181–191.

[14] B. Huang, X. Du, A robust design method using variable transformation and Gauss–Hermite integration, Int. J. Numer. Methods Engrg. 66 (12) (2006) 1841–1858.

[15] G. Taguchi, Introduction to Quality Engineering, American Supplier Institute, 1989.

[16] J. Koehler, A. Owen, Computer experiments, in: S. Ghosh, C. Rao (Eds.), Handbook of Statistics, vol. 13, Elsevier Science B.V., 1996, pp. 261–308.
[17] J. Sacks, W. Welch, J. Mitchell, H. Wynn, Design and analysis of computer experiments, Statistical Science 4 (4) (1989) 409–435.
[18] K.-H. Hwang, K.-W. Lee, G.-J. Park, Robust optimization of an automobile rearview mirror for vibration reduction, Struct. Multidisc. Optim. 21 (2001) 300–308.
[19] M. Trosset, Taguchi and robust optimization, Tech. Rep. 96-31, Department of Computational & Applied Mathematics, Rice University, 1996.
[20] V.N. Nair, B. Abraham, J. MacKay, G. Box, R.N. Kacker, T.J. Lorenzen, J.M. Lucas, R.H. Myers, G.G. Vining, J.A. Nelder, M.S. Phadke, J. Sacks, W.J. Welch, A.C. Shoemaker, K.L. Tsui, S. Taguchi, C.F.J. Wu, Taguchi's parameter design: a panel discussion, Technometrics 34 (2) (1992) 127–161.
[21] W. Chen, J. Allen, K.-L. Tsui, F. Mistree, A procedure for robust design: Minimizing variations caused by noise factors and control factors, ASME J. Mech. Des. 118 (4) (1996) 478–493.
[22] D. Jung, B. Lee, Development of a simple and efficient method for robust optimization, Int. J. Numer. Methods Engrg. 53 (9) (2002) 2201–2215.
[23] G. Klir, T. Filger, Fuzzy Sets, Uncertainty, and Information, Prentice-Hall, Englewood Cliffs, NJ, 1998.
[24] J. Helton, Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainties, J. Statist. Comput Simul. 57 (1997) 3–76.
[25] C. Seepersad, J. Allen, D. McDowell, F. Mistree, Robust design of cellular materials with topological and dimensional imperfections, J. Mech. Des. 128 (6) (2006) 1285–1297.
[26] S. Mahadevan, R. Rebba, Inclusion of model errors in reliability-based optimization, J. Mech. Des. 128 (4) (2006) 936–944.
[27] A. Lewis, Robust regularization, Technical Report, Simon Fraser University, Vancouver, Canada, September 2002.
[28] H.-G. Beyer, H.-P. Schwefel, Evolution strategies: a comprehensive introduction, Natural Comput. 1 (1) (2002) 3–52.
[29] A. Eiben, J. Smith, Introduction to Evolutionary Computing, Natural Computing Series, Springer, Berlin, 2003.
[30] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, Reading, MA, 1989.
[31] D. Goldberg, The Design of Innovation: Lessons from and for Competent Genetic Algorithms, Kluwer Academic, Boston, 2002.
[32] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, second ed., Springer, Berlin, 1994.
[33] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1996.
[34] I. Rechenberg, Evolutionsstrategie '94, Frommann-Holzboog Verlag, Stuttgart, 1994.
[35] M. McIlhagga, P. Husbands, R. Ives, A comparison of search techniques on a wing-box optimisation problem, in: H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature, vol. 4, Springer, Berlin, 1996, pp. 614–623.
[36] J. Herrmann, A genetic algorithm for minimax optimization problems, in: Proceedings of the Congress on Evolutionary Computation, vol. 2, IEEE Press, 1999, pp. 1099–1103.
[37] L. El Ghaoui, H. Lebret, Robust solutions to least-squares problems with uncertain data, SIAM J. Matrix Anal. Appl. 18 (4) (1997) 1035–1064.
[38] A. Ben-Tal, A. Nemirovski, Robust convex optimization, Math. Oper. Res. 23 (4) (1998) 769–805.
[39] A. Ben-Tal, A. Nemirovski, Robust optimization – methodology and applications, Math. Programm. 92 (3) (2002) 453–480.
[40] J. Nocedal, W. Wright, Numerical Optimization, Springer-Verlag, New York, 1999.
[41] D. Bertsimas, D. Pachamanova, M. Sim, Robust linear optimization under general norms, Oper. Res. Lett. 32 (6) (2004) 510–516.
[42] D. Bertsimas, M. Sim, The price of robustness, Oper. Res. 52 (1) (2004) 35–53.
[43] D. Bertsimas, M. Sim, Robust discrete optimization and network flows, Math. Programm. 98 (2003) 49–71.
[44] D. Bertsimas, M. Sim, Robust discrete optimization under ellipsoidal uncertainty sets, Tech. Rep. Working Paper, MIT, 2004.
[45] D. Bertsimas, M. Sim, Robust conic optimization, Tech. Rep. Working Paper, MIT 2004.
[46] D. Bertsimas, A. Thiele, A robust optimization approach to supply chain management, Tech. Rep. Working Paper, MIT, 2003.
[47] K. Otto, E. Antonsson, Design parameter selection in the presence of noise, Res. Engrg. Des. 6 (4) (1994) 234–246.
[48] R.-B. Wets, Modeling and solution strategies for unconstrained stochastic optimization problems, Ann. Oper. Res. 1 (1984) 3–22.
[49] H.-G. Beyer, M. Olhofer, B. Sendhoff, On the behavior of $(\mu/\mu_I, \lambda)$-ES optimizing functions disturbed by generalized noise, in: K. De Jong, R. Poli, J. Rowe (Eds.), Foundations of Genetic Algorithms, vol. 7, Morgan Kaufman, San Francisco, CA, 2003, pp. 307–328.
[50] I. Das, Robustness optimization for constrained, nonlinear programming problems, Tech. Rep. TR97-01, Technical Reports, Department of Computational & Applied Mathematics, Rice University, Houston, TX (1997).
[51] D. Wiesmann, U. Hammel, T. Bäck, Robust design of multilayer optical coatings by means of evolutionary algorithms, IEEE Trans. Evolution. Comput. 2 (4) (1998) 162–167.
[52] J. Branke, Creating robust solutions by means of evolutionary algorithms, in: A. Eiben, T. Bäck, M. Schoenauer, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature, vol. 5, Springer-Verlag, Heidelberg, 1998, pp. 119–128.
[53] J. Branke, Evolutionary Optimization in Dynamic Environments, Kluwer Academic Publishers, Dordrecht, 2002.
[54] H. Leung, Y. Zhu, I. Revilla, J. Fan, H. Chen, I. Pangga, C. Vera Cruz, T. Mew, Using genetic diversity to achieve sustainable rice disease management, Plant Disease 87 (10) (2003) 1156–1169.
[55] J. Darlington, C.C. Pantelides, B. Rustem, B. Tanyi, Algorithms for optimal decisions under uncertainty, in: 4th International Parallel Computing Workshop, PCW'95, Imperial College/Fujitsu Parallel Computing Research Centre, London, 1995, pp. 137–157.
[56] K.-H. Lee, G.-J. Park, Robust optimization considering tolerances of design variables, Comput. Struct. 79 (2001) 77–86.
[57] S. Takriti, S. Ahmed, On robust optimization of two-stage systems, Tech. Rep., Mathematical Sciences Department, T.J. Watson Research Center, IBM, Yorktown Heights, NY, 2001.
[58] W. Chen, M. Wiecek, J. Zhang, Quality utility – a compromise programming approach to robust design, ASME J. Mech. Des. 121 (2) (1999) 179–187.
[59] N. Rolander, J. Rambo, Y. Joshi, J. Allen, F. Mistree, An approach to robust design of turbulent convective systems, J. Mech. Des. 128 (4) (2006) 844–855.
[60] Y. Jin, B. Sendhoff, Trade-off between performance and robustness: an evolutionary multiobjective approach, in: C. Fonseca, P. Fleming, E. Zitzler, K. Deb (Eds.), Evolutionary Multi-Criterion Optimization: Second International Conference, EMO 2003, Springer-Verlag, Heidelberg, 2003, pp. 237–251.
[61] T. Ray, Constrained robust optimal design using a multiobjective evolutionary algorithm, in: Proceedings of the CEC'02 Conference, IEEE, Piscataway, NJ, 2002, pp. 419–424.
[62] N. Lagaros, V. Plevris, M. Papadrakakis, Multi-objective design optimization using cascade evolutionary computation, Comput. Methods Appl. Mech. Engrg. 194 (2005) 3496–3515.
[63] J. Zhang, M. Wiecek, W. Chen, Local approximation of the efficient frontier in robust design, J. Mech. Des. 122 (2) (2000) 232–236.
[64] J.-S. Kang, M.-H. Suh, T.-Y. Lee, Robust economic optimization of process design under uncertainty, Engrg. Optim. 36 (1) (2004) 51–75.
[65] H. Agarwal, Reliability based design optimization: Formulations and methodologies, Ph.D. Thesis, University of Notre Dame, South Bend, IN, 2004.
[66] N. Tatel, H. Agarwal, A. Tovar, J. Renaud, Reliability based topology optimization using a hybrid cellular automaton algorithm, in: J. Herskovits, S. Mazorche, A. Canelas (Eds.), Proceedings of the

6th World Congress on Structural and Multidisciplinary Optimization (WCSMO6), ISSMO, Rio de Janeiro, 2005.
[67] M. Papadrakakis, N. Lagaros, V. Plevris, Design optimization of steel structures considering uncertainties, Engrg. Struct. 27 (9) (2005) 1408–1418.
[68] S. Gunawan, P. Papalambros, Bayesian approach to reliability-based optimization with incomplete information, J. Mech. Des. 128 (4) (2006) 909–918.
[69] K.-Y. Chan, S. Skerlos, P. Papalambros, Monotonicity and active set strategies in probabilistic design optimization, J. Mech. Des. 128 (4) (2006) 893–900.
[70] M. Li, S. Azarm, A. Boyars, A new deterministic approach using sensitivity region measures for multi-objective robust and feasibility robust design optimization, J. Mech. Des. 128 (4) (2006) 874–883.
[71] H. Agarwal, J. Renaud, Reliability based design optimization using response surfaces in application to multidisciplinary systems, Engrg. Optim. 36 (3) (2004) 291–311.
[72] X. Du, W. Chen, Towards a better understanding of modeling feasibility robustness in engineering design, ASME J. Mech. Des. 122 (2) (2000) 385–394.
[73] B. Youn, K. Choi, The performance moment integration method for reliability-based robust design optimization, in: Proceedings of Design Engineering Technical Conference DETC'04, ASME, 2004, DETC2004–57471.
[74] M. Mourelatos, J. Liang, A methodology for trading-off performance and robustness under uncertainty, J. Mech. Des. 128 (4) (2006) 856–863.
[75] L. Simões, Fuzzy optimization of structures by the two-phase method, Computers & Structures 79 (26–28) (2001) 2481–2490.
[76] F. Campos, J. Villar, M. Jiménez, Robust solutions using fuzzy chance constraints, Engrg. Optim. 38 (6) (2006) 627–645.
[77] T. Ross, Fuzzy Logic with Engineering Applications, McGraw-Hill, New York, 1995.
[78] J. Buckley, E. Eslami, An Introduction to Fuzzy Logic and Fuzzy Sets, Physica-Verlag, A Springer-Verlag Company, Heidelberg, 2002.
[79] M. Savoia, Structural reliability analysis through fuzzy number approach, with application to stability, Comput. Struct. 80 (12) (2002) 1087–1102.
[80] K. Sentz, S. Ferson, Combination of evidences in Dempster-Shafer theory, Tech. Rep. Report No. SAND2002-0835, Sandia National Laboratories, 2002.
[81] Z. Mourelatos, J. Zhou, A design optimization method using evidence theory, J. Mech. Des. 128 (4) (2006) 901–908.
[82] L. Choi, K.K. amd Du, B. Youn, D. Gorsich, Possibility-based design optimization method for design problems with both statistical and fuzzy input data, in: 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil, 2005.
[83] B. Youn, K. Choi, L. Du, D. Gorsich, Integration of possibility-based optimization to robust design for epistemic uncertainty, in: 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil, (2005).
[84] S. Chen, Comparing probabilistic and fuzzy set approaches for design in the presence of uncertainty, Ph.D. thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2000.
[85] M. Arakawa, H. Yamakawa, H. Ishikawa, Robust design using fuzzy numbers with intermediate variables, in: 3rd World Congress of Structural and Multidisciplinary Optimization, 1999.
[86] S. Andradóttir, A review of simulation optimization techniques, in: D. Medeiros, E. Watson, J. Carson, M. Manivannan (Eds.), Proceedings of the 1998 Winter Simulation Conference, IEEE, Piscataway, NJ, 1998, pp. 151–158.
[87] A. Ben-Tal, A. Nemirovski, C. Roos, Robust solutions of uncertain quadratic and conic-quadratic problems, SIAM J. Optim. 13 (2) (2002) 535–560.
[88] D. Bertsimas, M. Sim, Robust discrete optimization, Tech. Rep., Sloan School of Management and Operations Research Center, MIT, Cambridge, MA, 2002.
[89] M. Lobo, Robust and convex optimization with applications in finance, Ph.D. thesis, Department of Electrical Engineering, Stanford University, Stanford, 2000.
[90] F. Lutgens, J. Sturm, Robust one period option modelling, Tech. Rep. CentER DP 2002-114, Universiteit van Tilburg, 2002.
[91] Y. Kanno, I. Takewaki, Evaluation and maximization of robustness of trusses by using semidefinite programming, in: J. Herskovits, S. Mazorche, A. Canelas (Eds.), Proceedings of the 6th World Congress on Structural and Multidisciplinary Optimization (WCSMO6), ISSMO, Rio de Janeiro, 2005.
[92] S. Sundaresan, K. Ishii, D. Houser, A robust optimization procedure with variations on design variables and constraints, in: ASME Design Automation Conference, ASME, 1993, pp. 387–394.
[93] Y. Zhang, A general robust-optimization formulation for nonlinear programming, Technical Report TR-0413, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 77005, 2005.
[94] A. Parkinson, C. Sorensen, N. Pourhassan, A general approach for robust optimal design, Trans. ASME 115 (1993) 74–80.
[95] X. Du, W. Chen, A methodology for managing the effect of uncertainty in simulation-based design, AIAA J. 38 (8) (2000) 1471–1478.
[96] A. Haldar, S. Mahadevan, Probability, Reliability, and Statistical Methods in Engineering Design, Wiley, New York, 2000.
[97] R. Rackwitz, Reliability analysis – a review and some perspectives, Structural Safety 23 (2001) 365–395.
[98] S.-B. Yoon, I.-S. Jung, D.-S. Hyun, J.-P. Hong, Y.-J. Kim, Robust shape optimization of electromechanical devices, IEEE Trans. Magn. 35 (3) (1999) 1710–1713.
[99] C. Gumber, P. Newman, G.-W. Hou, Effect of random geometry uncertainty on the computational design of a 3-d flexible wing, American Institute of Aeronautics and Astronautics AIAA-2002-2806.
[100] I. Doltsinis, Z. Kang, Robust design of structures using optimization methods, Comput. Methods Appl. Mech. Engrg. 193 (2004) 2221–2237.
[101] K.S. Miller, Multidimensional Gaussian Distributions, J. Wiley, New York, 1964.
[102] H.-G. Beyer, The Theory of Evolution Strategies, Natural Computing Series, Springer, Heidelberg, 2001.
[103] J.-C. Yu, Design optimization for robustness using quadrature factorial models, Engrg. Optim. 30 (1998) 203–225.
[104] L. Green, H.-Z. Lin, M. Khalessi, Probabilistic methods for uncertainty propagation applied to aircraft design, in: 20th AIAA Applied Aerodynamics Conference, American Institute of Aeronautics and Astronautics, paper: AIAA 2002-3140, 2002.
[105] S. Padula, W. Li, Options for robust airfoil optimization under uncertainty, in: 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia, AIAA, 2002.
[106] M. Fu, Optimization for simulation: theory vs. practice, INFORMS J. Comput. 14 (2002) 192–215.
[107] J. Liu, Monte Carlo Strategies in Scientific Computing, Springer, New York, 2001.
[108] M. Papadrakakis, Y. Tsompanakis, N. Lagaros, M. Fragiadakis, Reliability based optimization of steel frames under seismic loading conditions using evolutionary computation, J. Theor. Appl. Mech. 42 (3) (2004) 585–608.
[109] E. Sandgren, T. Cameron, Robust design optimization of structures through consideration of variation, Comput. Struct. 80 (20–21) (2002) 1605–1613.
[110] K.-H. Lee, G.-J. Park, W.-S. Joo, A general robust optimization using the kriging based approximation method, in: J. Herskovits, S. Mazorche, A. Canelas (Eds.), Proceedings of the 6th World Congress on Structural and Multidisciplinary Optimization (WCSMO6), ISSMO, Rio de Janeiro, 2005.
[111] J. Martin, T. Simpson, A Monte Carlo method for reliability-based design optimization, Tech. Rep. 2006-2146, American Institute of Aeronautics and Astronautics AIAA, 2006.

[112] H. Wang, N. Kim, Y.-J. Kim, Safety envelope for load tolerance and its application to fatigue reliability design, J. Mech. Des. 128 (4) (2006) 919–927.
[113] T. Simpson, J. Peplinski, P. Koch, J. Allen, On the use of statistics in design and the implications for deterministic computer experiments, in: J. Shah (Ed.), ASME Design Theory and Methodology '97, ASME, New York, 1997, ASME97–DETC97/DTM–3881.
[114] Y. Jin, A comprehensive survey of fitness approximation in evolutionary computation, Soft Comput. J. 9 (1) (2005) 3–12.
[115] R. Jin, X. Du, W. Chen, The use of metamodeling techniques for optimization under uncertainty, J. Struct. Multidisciplinary Optim. 25 (2) (2003) 99–116.
[116] P. Chang, B. Williams, T. Santner, W. Notz, D. Bartel, Robust optimization of total joint replacements incorporating environmental variables, J. Biomech. Eng. 121 (1999) 304–310.
[117] D. Xu, S. Albin, Robust optimization of experimentally derived objective functions, IIE Trans. 35 (2003) 793–802.
[118] E. Elster, A. Neumaier, A grid algorithm for bound constrained optimization of noisy functions, IMA J. Numer. Anal. 15 (4) (1995) 585–608.
[119] K. Sörensen, Tabu searching for robust solutions, in: Proceedings of the 4th Metaheuristics International Conference, MIC'2001, Porto, Portugal, 2001, pp. 707–712.
[120] H. Robbins, S. Monro, A stochastic approximation method, Ann. Math. Statist. 29 (1951) 400–407.
[121] Y. Ermoliev, Stochastic quasigradient methods, in: Y. Ermoliev, R.-B. Wets (Eds.), Numerical Techniques for Stochastic Optimization, Springer-Verlag, Berlin, 1988, pp. 141–185.
[122] D.V. Arnold, H.-G. Beyer, On the effects of outliers on evolutionary optimization, in: J. Liu, Y.-M. Cheung, H. Yin (Eds.), Fourth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2003), Springer, Heidelberg, Germany, 2003, pp. 151–160.
[123] J. Spall, An overview of the simultaneous perturbation method for efficient optimization, Johns Hopkins APL Technical Dig. 19 (1998) 482–492.
[124] T. Choi, C. Kelley, Superlinear convergence and implicit filtering, SIAM J. Optim. 10 (4) (2000) 1149–1162.
[125] M. Wright, Direct search methods: once scorned, new respectable, in: D. Griffiths, G. Watson (Eds.), Numerical Analysis, Addison Wesley Longman Limited, Redwood, 1995, pp. 191–208.
[126] J. Nelder, R. Mead, A simplex method for function minimization, Computer Journal 7 (1965) 308–313.
[127] J. Lagarias, J. Reeds, M. Wright, P. Wright, Convergence properties of the Nelder–Mead simplex method in low dimensions, SIAM J. Optim. 9 (1) (1998) 112–147.
[128] R. Barton, J. Ivey, Nelder–Mead simplex modifications for simulation optimization, Manage. Sci. 42 (7) (1996) 954–973.
[129] D. Humphrey, J. Wilson, A revised simplex search procedure for stochastic simulation response surface optimization, INFORMS J. Comput. 12 (4) (2002) 272–283.
[130] E. Anderson, M. Ferris, A direct search algorithm for optimization with noisy function evaluations, SIAM J. Optim. 11 (3) (2001) 837–857.
[131] D.B. Fogel, Evolving Artificial Intelligence, Ph.D. thesis, University of California, San Diego, 1992.
[132] J. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, MA, 1992.
[133] T. Bäck, U. Hammel, H.-P. Schwefel, Evolutionary computation: comments on the history and current state, IEEE Trans. Evolution. Comput. 1 (1) (1997) 3–17.
[134] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[135] E. Aarts, P. van Laarhoven, Simulated Annealing: Theory and Applications, Reidel, Dordrecht, 1987.
[136] J. Spall, Stochastic optimization: stochastic approximation and simulated annealing, in: J. Webster (Ed.), Encyclopedia of Electrical and Electronics Engineering, vol. 20, Wiley, New York, 1999, pp. 529–542.
[137] A. Sebald, D. Fogel, Design of fault-tolerant neural networks for pattern classification, in: D. Fogel, W. Atmar (Eds.), Proc. of the First Annual Conf. on Evolutionary Programming, Evolutionary Programming Society, La Jolla, CA, 1992, pp. 90–99.
[138] H. Greiner, Robust filter design by stochastic optimization, in: F. Abeles (Ed.), Proc. SPIE, vol. 2253, The International Society for Optical Engineering, 1994, pp. 150–160, paper: AIAA 2002-3140.
[139] O. Pictet, M. Dacorogna, B. Chopard, M. Oussaidene, R. Schirru, M. Tomassini, Using genetic algorithms for robust optimization in financial applications, Tech. Rep. OVP.1995-02-06, Olsen & Associates, Research Institute for Applied Economics, Zürich, Switzerland, 1996.
[140] D. Wiesmann, Robust design mit evolutionsstrategien, Diploma thesis, University of Dortmund, Department of Computer Science, 1997.
[141] H. Loughlin, S. Ranjithan, Chance-constrained genetic algorithms, in: W. Banzhaf, J. Daida, A. Eiben, M. Garzon, V. Honavar, M. Jakiela, R. Smith (Eds.), GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann, San Francisco, CA, 1999, pp. 369–376.
[142] E. Kazancioglu, G. Wu, J. Ko, S. Bohac, Z. Filipi, S. Hu, D. Assanis, K. Saitou, Robust optimization of an automobile valvetrain using a multiobjective genetic algorithm, in: Proceedings of DETC'03 ASME 2003 Design Engineering Technical Conferences, Chicago, September 2–6, 2003, DETC03/DAC-48714.
[143] M. Sevaux, K. Sörensen, A genetic algorithm for robust schedules in a just-in-time environment, Research Report LAMIH SP-2003-1, University of Valenciennes, CNRS, UMR 8530, France, 2003.
[144] M. Sevaux, Y. Le Quéré, Solving a robust maintenance scheduling problem at the French railways company, Research Report LAMIH SP-2003-3, University of Valenciennes, CNRS, UMR 8530, France, March 2003.
[145] A. Kumar, A. Keane, P. Nair, S. Shahpar, Robust design of compressor fan blades against erosion, J. Mech. Des. 128 (4) (2006) 864–873.
[146] Y. Ong, P. Nair, K. Lum, Max–min surrogate assisted evolutionary algorithm for robust design, IEEE Trans. Evolution. Comput. 10 (4) (2006) 392–404.
[147] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, Trusted evolutionary algorithm, in: Congress on Evolutionary Computation (CEC), IEEE Press, 2006, pp. 456–463.
[148] T. Ray, W. Smith, A surrogate assisted parallel multiobjective evolutionary algorithm for robust engineering design, Engrg. Optim. 38 (8) (2006) 997–1011.
[149] E. Zechmann, S. Ranjithan, An evolutionary algorithm to generate alternatives (EAGA) for engineering optimization problems, Engrg. Optim. 36 (5) (2004) 539–553.
[150] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, B. Lee, Inverse multiobjective robust evolutionary design, Genetic Programm. Evolvable Mach. 7 (4) (2006) 383–404.
[151] H.-G. Beyer, Actuator noise in recombinant evolution strategies on general quadratic fitness models, in: K. Deb et al. (Eds.), GECCO-2004: Proceedings of the Genetic and Evolutionary Computation Conference, LNCS, vol. 3102, Springer-Verlag, Heidelberg, 2004, pp. 654–665.
[152] S. Tsutsui, A. Ghosh, Y. Fujimoto, A robust solution searching scheme in genetic search, in: H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature, vol. 4, Springer, Heidelberg, 1996, pp. 543–552.
[153] S. Tsutsui, A. Ghosh, Genetic algorithms with a robust solution searching scheme, IEEE Trans. Evolution. Comput. 1 (3) (1997) 201–208.
[154] S. Tsutsui, A comparative study on the effects of adding perturbations to phenotypic parameters in genetic algorithms with a robust solution searching scheme, Proceedings of the 1999 IEEE System,

Man, and Cybernetics Conference – SMC'99, vol. 3, IEEE, 1996, pp. 585–591.
[155] J.H. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, 1975.
[156] K. De Jong, Genetic algorithms are not function optimizers, in: L.D. Whitley (Ed.), Foundations of Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 1993, pp. 5–17.
[157] G. Rudolph, Convergence Properties of Canonical Genetic Algorithms, IEEE Transaction on Neural Networks 5 (1) (1994) 96–101.
[158] H.-G. Beyer, B. Sendhoff, Evolution Strategies for Robust Optimization, in: Proceedings of the WCCI'06 Conference, IEEE Press, Piscataway, NJ, 2006, pp. 4489–4496.
[159] A. Thompson, Evolutionary techniques for fault tolerance, in: Proc. UKACC Intl. Conf. on Control, IEE Conference Publication, 1996, pp. 693–698.
[160] A. Thompson, P. Layzell, Evolution of robustness in an electronics design, in: J. Miller, A. Thompson, P. Thomson, T. Fogarty (Eds.), Evolvable Systems: From Biology to Hardware, Proceedings of the Third International Conference (ICES-2000), LNCS, vol. 1801, Springer-Verlag, Berlin, 2000, pp. 218–228.
[161] D.V. Arnold, H.-G. Beyer, A comparison of evolution strategies with other direct search methods in the presence of noise, Comput. Optim. Appl. 24 (2003) 135–159.
[162] M. Bell, M. Pike, Remark on Algorithm 178, Commun. ACM 9 (1966) 684–685.
[163] V. Torczon, Multi-directional search: a direct search algorithm for parallel machines, Ph.D. thesis, Department of Computational & Applied Mathematics, Rice University, Houston, TX, USA, 1989.
[164] P. Gilmore, C. Kelley, An implicit filtering algorithm for optimization of functions with many local minima, SIAM J. Optim. 5 (1995) 269–285.
[165] H.-G. Beyer, M. Olhofer, B. Sendhoff, On the impact of systematic noise on the evolutionary optimization performance – a sphere model analysis, Genetic Programm. Evolvable Mach. 5 (4) (2004) 327–360.
[166] B. Sendhoff, H.-G. Beyer, M. Olhofer, On noise induced multimodality in evolutionary algorithms, in: L. Wang, K. Tan, T. Furuhashi, J.-H. Kim, F. Sattar (Eds.), Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning – SEAL, vol. 1, 2002, pp. 219–224.
[167] H.-G. Beyer, B. Sendhoff, Functions with noise-induced multimodality: A test for evolutionary robust optimization – properties and performance analysis, IEEE Trans. Evolution. Comput. 10 (5) (2006) 507–526.
[168] T. Bartz-Beielstein, Experimental Research in Evolutionary Computation – The New Experimentalism, Natural Computing Series, Springer, Berlin, 2006.
[169] M. Hasenjäger, B. Sendhoff, T. Sonoda, T. Arima, Three dimensional aerodynamic optimization for an ultra-low aspect ratio transonic turbine stator blade, in: Proceedings of the ASME Turbo Expo, 2005.
[170] M. Hasenjäger, B. Sendhoff, T. Sonoda, T. Arima, Three dimensional evolutionary aerodynamic design optimization with CMA-ES, in: H.-G. Beyer et al. (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference – GECCO, 2005, pp. 2173–2180.
[171] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evolutionary Computation 9 (2) (2001) 159–195.
[172] N. Hansen, S. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evolution. Comput. 11 (1) (2003) 1–18.
[173] H.-P. Schwefel, Numerical Optimization of Computer Models, Wiley, Chichester, 1981.
[174] N. Hansen, A. Ostermeier, Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation, in: Proceedings of 1996 IEEE Int'l. Conf. on Evolutionary Computation (ICEC '96), IEEE Press, NY, 1996, pp. 312–317.