Evolution Strategies for Robust Optimization

Hans-Georg Beyer, Bernhard Sendhoff

2006

Evolution Strategies for Robust Optimization

Hans-Georg Beyer and Bernhard Sendhoff, *Senior Member, IEEE*

***Abstract*— In this paper, we propose two evolutionary strategies for the optimization of problems with *actuator noise* as encountered in *robust optimization*, where the design or objective parameters are subject to noise: the ROSAES and the ROCSAES. Both algorithms use a control rule for increasing the population size when the residual error to the optimizer state has been reached. Theoretical analysis has previously shown that the residual error depends among other factors on the population size and on the variance of the noise. Furthermore, ROSAES exploits the similarity of the mutation term in evolutionary strategies and the additive noise term in the case of actuator noise. The population variance is controlled to guarantee that the realized noise level is adjusted correctly. Simulations are carried out on test functions and the results are analyzed with respect to the performance and the dependence of ROSAES and ROCSAES on newly introduced exogenous strategy parameters.**

I. Introduction

Optimization in the presence of noise (robust optimization) has received increasing attention in recent years not least due to the factual necessity to deal with this problem for many (if not most) practical optimization cases. That is, given a design $\mathbf{x}$, evaluating its quality $\tilde{f}(\mathbf{x})$ yields stochastic quantity values. As a result, an optimization algorithm applied to $\tilde{f}(\mathbf{x})$ must deal with these uncertain quality information and it must use this information to calculate a robust optimum based on an appropriate robustness measure.

Probably the most widely used measure is the expected value of $\tilde{f}(\mathbf{x})$, i.e. $\mathrm{E}[\tilde{f}|\mathbf{x}]$ [1]. If one were able to calculate $\mathrm{E}[\tilde{f}|\mathbf{x}]$ analytically, the resulting optimization problem would be an ordinary one, and standard (numerical) optimization techniques could be applied. However, real-world applications will usually not allow for an analytical treatment. Therefore, one has to rely on numerical estimates of $\mathrm{E}[\tilde{f}|\mathbf{x}]$ using Monte Carlo simulations. Alternatively, one can use direct search strategies capable of dealing with the noisy information directly. This is clearly the domain of evolutionary algorithms (EAs) [2]. A short review of evolutionary algorithms applied to different classes of noisy optimization problems can be found in [3].

In this paper, we will consider the design of Evolution Strategies (ESs) especially tailored for the treatment of design uncertainties. In this class, noise is added to the object or design variables that are subject to optimization [4]. This kind of uncertainties concerns e.g. production tolerances and limitations, or actuator imprecision $\boldsymbol{\delta}$ acting directly on the design variables $\mathbf{x}$. That is, the performance f of a design becomes a stochastic quantity $\tilde{f}$ via *internal* design perturbations $\tilde{f}(\mathbf{x}) = f(\mathbf{x}+\boldsymbol{\delta})$, where the random vector $\boldsymbol{\delta}$ obeys a user specified distribution (often Gaussian distribution) and $\mathrm{E}[\boldsymbol{\delta}] = \mathbf{0}$.

Assuming a continuous design space, the expected value robustness measure is given by the integral

$$\mathrm{E}[\tilde{f}|\mathbf{x}] = \int_{\mathbb{R}^N} f(\mathbf{x}+\boldsymbol{\delta})p(\boldsymbol{\delta})\,\mathrm{d}^N\boldsymbol{\delta} \tag{1}$$

and the optimal design $\hat{\mathbf{x}}$ is formally obtained by

$$\hat{\mathbf{x}} = \arg\mathrm{opt}_{\mathbf{x}} \int_{\mathbb{R}^N} f(\mathbf{x}+\boldsymbol{\delta})p(\boldsymbol{\delta})\,\mathrm{d}^N\boldsymbol{\delta}. \tag{2}$$

The usual way how one tries to find approximate solutions to (2) using EAs is to use the design uncertainties $\boldsymbol{\delta}$ explicitly. That is, given an individual design $\mathbf{x}$, the perturbation $\boldsymbol{\delta}$ is added explicitly to the design $\mathbf{x}$. While the EA works on the evolution of $\mathbf{x}$, the goal function in the black-box is evaluated w.r.t. $\tilde{\mathbf{x}} := \mathbf{x} + \boldsymbol{\delta}$. Since in $(\mu/\mu_I, \lambda)$-ESs considered here, an individual offspring design is the result of a mutation $\mathbf{z}$ applied to the parental centroid $\langle\mathbf{x}\rangle$, the actually design tested is $\tilde{\mathbf{x}} = \langle\mathbf{x}\rangle + \mathbf{z} + \boldsymbol{\delta}$. Taking now another perspective, one might interpret $\mathbf{z}+\boldsymbol{\delta}$ as a *mutation* in its own right. This raises the question whether it is really necessary to artificially add the perturbation $\boldsymbol{\delta}$ in a black-box to the design $\mathbf{x}$. As an alternative one might simply use a mutation $\tilde{\mathbf{z}} = \mathbf{z} + \boldsymbol{\delta}$ with a *larger* mutation strength instead of $\mathbf{z}$. In other words, the *mutations itself may serve as robustness tester*. In this paper, we will introduce an algorithm that does exactly this.

Even though ESs are regarded as well suited for noisy optimization, its application to robust optimization bears some subtleties not well known: Due to selection, the robustness of a design $\mathbf{x}$ is *not* tested w.r.t. samples of the density function $p(\boldsymbol{\delta})$. Selection prefers those designs which are by chance well adopted to the *individual* realizations of the perturbation $\boldsymbol{\delta}$. For example, when considering actuator noise of standard deviation ε on a sphere model $\|\mathbf{x}\|^2$ (to be minimized), the actually measured standard deviation D_i of a specific component i of the parent population will usually be smaller, i.e. $D_i < \varepsilon$. This is so because selection singles out all those $\mathbf{x}+\boldsymbol{\delta}$ states with large length $\|\mathbf{x}+\boldsymbol{\delta}\|$. That is, shorter $\boldsymbol{\delta}$ vectors are preferred resulting in a smaller measured standard deviation. Therefore, an ES algorithm for robust optimization must take into account this effect and take appropriate counter measures. It is the aim of this work to present a rule that allows for controlling the observed parental variance in such a way that robustness is tested correctly.

From theoretical results it is known that evolution strategies can only get arbitrarily close to the optimizer state of an optimization problem with actuator noise, in the limit

H.-G. Beyer is with the Vorarlberg University of Applied Sciences, Dornbirn, Austria (email: Hans-Georg.Beyer@fhv.at).

B. Sendhoff is with the Honda Research Institute Europe, Carl-Legien-Str. 30, 63073 Offenbach, Germany (email: bs@honda-ri.de).

of infinite population. Thus, practically, a residual error cannot be circumvented. This residual error decreases with the population size (and increases with the noise strength). Therefore, as a second concept for the robust evolutionary strategies we introduce a rule for population control that adaptively increases the population size when it is necessary to get closer to the optimizer state. The higher fidelity of the optimization results of course has to be paid for by a larger number of function evaluations. However, this increase is controlled. Alternatively, one might use κ times resampling, i.e., averaging over κ repeated f-measurements, where κ is increased over the run if need be. However, there is empirical [5] and theoretical evidence [6], [7], [13] that increasing the population size is usually more efficient.

After an introduction of the nomenclature in the next section, we will outline two robust evolution strategies ROSAES and ROCSAES using a pseudo-code description of the algorithms. In Section V, we present and discuss experiments on two test functions: the sphere function with actuator noise and a function with noise induced multimodality. We will conclude in the last section.

II. Notational Conventions

In order to unify and simplify the notations for the pseudo code description of the algorithms the following conventions will be used:

1) g is the generation (time) counter, it appears as parenthesized superscript on the respective quantities;
2) $\mathbf{x} \in \mathbb{R}^N$ is the N-dimensional object parameter vector. $x_i \equiv (\mathbf{x})_i$ its ith component.
3) N denotes the object parameter space dimension;
4) σ is the standard deviation of the normally distributed mutations in ES;
5) μ is the parental population size. Quantities related to parental individuals are indexed by subscript m.
6) λ is the number of offspring generated in a single generation. Quantities related to offspring individuals are indexed by subscript l and are denoted with a tilde.
7) ϑ is the truncation ratio, $\vartheta := \frac{\mu}{\lambda}$.
8) Normally distributed random variables/numbers y are denoted by $\mathcal{N}(\overline{y}, \sigma^2)$ where $\overline{y}$ is the expected value of y and σ its standard deviation.
9) A vector $\mathbf{y}$ of normally distributed random variates is symbolized by a boldface $\boldsymbol{\mathcal{N}}(\overline{\mathbf{y}}, \boldsymbol{V})$, where $\overline{\mathbf{y}}$ is the expected value vector of $\mathbf{y}$ and $\boldsymbol{V}$ stands either for the covariance matric $\mathbf{C}$ or – somewhat unusual – the vector of *standard deviations*.
10) The subscript notation $m;\lambda$ denotes quantities of the mth-best individual, i.e., that individual being the mth-largest (in the case of maximization) or smallest (in the case of minimization) w.r.t. its *observed* (measured) fitness $f(\mathbf{x})$.
11) $\langle \mathbf{y} \rangle$ denotes the parental population average, i.e.
$$\langle \mathbf{y} \rangle := \tfrac{1}{\mu} \textstyle\sum_{m=1}^{\mu} \mathbf{y}_m. \tag{3}$$
This is basically a centroid calculation.
12) Overlined symbols, e.g. $\overline{x_i}$, when used in an algorithm, are used to denote averaging over time, i.e. usually this is a weighted average over the generations g.
13) R is the length of the centroid state
$$R := \|\langle \mathbf{x} \rangle\|. \tag{4}$$
14) r is the length of the first $N-1$ components of the vector $\mathbf{x}$
$$r := \sqrt{\textstyle\sum_{i=1}^{N-1} x_i^2}. \tag{5}$$

III. ROSAES – Robust-Optimization-$(\mu/\mu_I, \lambda)$-σSA-ES

To meet the required properties an EA should fulfill in order to qualify as a strategy for robust optimization (optimization under actuator noise), the EA must contain:

1) a strategy for controlling the *actually observed* fluctuation strength of the actually tested parental object parameter sets,
2) a strategy for successively increasing the population size in order to prevent the EA from reaching a steady state fitness error.

The algorithm shown in Fig. 1 realizes the first item through the Lines 18–24 and the second item through the Lines 25–30. We will now discuss the proposed ROSAES in detail explaining the different parts separately.

1) The $(\mu/\mu_I, \lambda)$-Core: ROSAES is built upon the standard $(\mu/\mu_I, \lambda)$-σSA-ES using log-normal mutations for the mutation strength σ. The procreation of the λ offspring is done in Lines 8–14, with Line 9 performing the log-normal mutation of the recombined strategy parameter σ.[1] In Line 11, the mutation of the object parameter is performed. As usual this is done on top of the recombinant $\langle \mathbf{x} \rangle$ (again the intermediate recombination needs to be done only once in Line 17), however the *actual* strength by which the mutation is performed differs from "standard" ES: In order to account for the actuator noise, the strength consists of the ES-specific contribution σ and an actuator noise contribution ε. Since normality of the actuator noise is assumed, the sum of the strategy-specific mutation contribution and the actuator noise contribution is still a normally distributed random vector, however, with variance $\tilde{\sigma}_l^2 + \varepsilon_i^2$ for the ith component. Performing the mutations in this way allows for taking advantage of the ES-immanent mutation (of strength σ_l) as an additional robustness tester. That is, there is no need to decrease the ES's mutation strength during evolution to very small values since the mutation can take over a part of the robustness testing itself. Whether this idea results in a performance advantage compared to the usual robust optimization approach[2] is still an open issue and needs further thorough (simulative) exploration.

[1] The intermediate recombination itself is done in Line 16, since it needs to be calculated only once per generation.

[2] By "usual approach" we mean the standard evolutionary robustness test by which the actuator noise is generated in the black-box invisible to the EA. This approach will be taken in Section IV for the CSA-ES.

```
RO-$(\mu/\mu_I, \lambda)$-$\sigma$SA-ES (Maximization)
$g := 0$;   $\langle\sigma\rangle := \sigma^{(\text{init})}$;   $\langle\mathbf{x}\rangle := \mathbf{x}^{(\text{init})}$;   1
$\mu := \mu^{(\text{init})}$;   $\lambda := \lceil\mu/\vartheta\rceil$;   $\overline{\mathbf{x}} := \langle\mathbf{x}\rangle$;   2
For $i := 1$ To $N$; $\overline{x_i^2} := \langle x_i^2\rangle$; End For   3
$\boldsymbol{\varepsilon} := \boldsymbol{\varepsilon}^\star$;   4
$\langle F\rangle^{(g)} := \frac{1}{\mu}\sum_{m=1}^{\mu} f\left(\mathbf{x}^{(\text{init})} + \boldsymbol{\mathcal{N}}_m(\mathbf{0}, \boldsymbol{\varepsilon}^\star)\right)$;   5
$\overline{\Delta F} := 0$;   6
Repeat   7
    For $l := 1$ To $\lambda$   8
        $\tilde{\sigma}_l := \langle\sigma\rangle \exp[\tau_\sigma \mathcal{N}_l(0,1)]$;   9
        For $i := 1$ To $N$;   10
            $(\tilde{\mathbf{x}}_l)_i := \langle x_i\rangle + \sqrt{\tilde{\sigma}_l^2 + \varepsilon_i^2}\,\mathcal{N}_{l,i}(0,1)$;   11
        End For   12
        $\tilde{F}_l := f(\tilde{\mathbf{x}}_l)$;   13
    End For   14
    $g := g + 1$;   15
    $\langle\sigma\rangle := \frac{1}{\mu}\sum_{m=1}^{\mu} \tilde{\sigma}_{m;\lambda}$;   16
    $\langle\mathbf{x}\rangle := \frac{1}{\mu}\sum_{m=1}^{\mu} \tilde{\mathbf{x}}_{m;\lambda}$;   17
    For $i := 1$ To $N$   18
        $\langle x_i^2\rangle := \frac{1}{\mu}\sum_{m=1}^{\mu} (\tilde{\mathbf{x}}_{m;\lambda})_i^2$;   19
        $\overline{x_i} := (1 - c_x)\overline{x_i} + c_x\langle x_i\rangle$;   20
        $\overline{x_i^2} := (1 - c_x)\overline{x_i^2} + c_x\langle x_i^2\rangle$;   21
        $D_i := \sqrt{\overline{x_i^2} - \overline{x_i}^2}$;   22
        $\varepsilon_i := \varepsilon_i \exp[\tau_\varepsilon \text{Sign}(\varepsilon_i^\star - D_i)]$;   23
    End For   24
    $\langle F\rangle^{(g)} := \frac{1}{\mu}\sum_{m=1}^{\mu} \tilde{F}_{m;\lambda}$;   25
    $\overline{\Delta F} := (1 - c_f)\overline{\Delta F} + c_f\left(\langle F\rangle^{(g)} - \langle F\rangle^{(g-1)}\right)$;   26
    If $(\text{Mod}(g, \Delta g) = 0) \wedge (\overline{\Delta F} \leq 0)$ Then   27
        $\mu := \lceil\mu c_\mu\rceil$;   28
        $\lambda := \lceil\mu/\vartheta\rceil$;   29
    End If   30
Until Termination_Condition   31
```

Fig. 1. Pseudocode of the RO-$(\mu/\mu_I, \lambda)$-σSA-ES, abbreviated: ROSAES.

The core part of ROSAES needs the fixing of an exogenous strategy parameter, the learning parameter τ_σ. It is well known that the progress rate of the $(\mu/\mu_I, \lambda)$-ES depends sensitively on τ_σ [8]. Therefore, we cannot expect to find a general optimal choice. However, in order to ensure linear convergence order on the sphere, it suffices to ensure $\tau_\sigma \propto 1/\sqrt{N}$ and

$$\tau_\sigma = N^{-1/2} \tag{6}$$

seems to be a reasonable choice.

2) The Robustness Variance Control: The actual mutation strength acting in Line 11 depends on σ and ε_i. It is important to realize that ε_i is *not* equivalent to the *desired* actuator noise strength $\varepsilon_i^\star$. The latter is the *desired* strength by which the *actually realized* design instances should be tested. As we have already argued, due to the (μ, λ)-selection, the actual variances of the selected (i.e. parental) $\tilde{\mathbf{x}}_{m;\lambda}$ states are usually smaller than the desired $\varepsilon_i^\star$. Therefore, ε_i must be *controlled* in such a way that the observed (i.e. measured) standard deviation

$$D_i := \sqrt{\text{Var}\left[\{(\tilde{\mathbf{x}}_{1;\lambda})_i, \ldots, (\tilde{\mathbf{x}}_{\mu;\lambda})_i\}\right]} \tag{7}$$

gets close to $\varepsilon_i^\star$.

In order to build a control algorithm we need basically two ingredients:

1) a method for measuring the parental population variance or its standard deviation D_i,
2) a control rule for changing the ε_i.

a) 1. Determining the population variance: Since robustness testing is highly noisy, calculating the parental population variance from just one generation results in highly fluctuating D_i estimates not well suited for ε_i control. Therefore, a smooth D_i estimate is needed. One way of smoothing the data is by weighted cumulation, also known as exponential averaging. The standard deviation of x_i

$$D_i = \sqrt{\overline{x_i^2} - \overline{x_i}^2}, \tag{8}$$

can be obtained from the smoothed time averages of x_i and x_i^2. The exponential smoothing is done in Lines 20 and 21 (Fig. 1), respectively. The population average of the squared x_i coordinates needed in Line 21 is calculated in Line 19. The exponential averaging in Lines 20 and 21 is designed in such a way that the $\overline{x_i}$ and $\overline{x_i^2}$ information fades away exponentially fast if $\langle x_i\rangle$ and $\langle x_i^2\rangle$, respectively, are zero. The time constant of this process is controlled by the cumulation time constant $c_x \in [0, 1]$. Since the changing rates of the ES (e.g. the progress rate on the sphere) are often of the order $1/N$, it is reasonable to use

$$c_x = N^{-1} \tag{9}$$

as a first choice.

b) 2. How to control ε_i.: Given a stable estimate of the real parental x_i population standard deviation, one can compare it with the desired actuator noise strength $\varepsilon_i^\star$. That is, the aim is to control the (observed) D_i in such a way that

$$D_i \approx \varepsilon_i^\star. \tag{10}$$

If the ES is able to get close to the robust optimizer, then condition (10) ensures that robustness is guaranteed for the correct target actuator noise strength. While, in general, we cannot be sure that the ES locates the robust optimizer (as we will see later on), fulfilling condition (10) can be ensured asymptotically by the control rule

$$\varepsilon_i := \varepsilon_i \exp[\tau_\varepsilon \text{Sign}(\varepsilon_i^\star - D_i)]. \tag{11}$$

If $D_i = \varepsilon_i^\star$, (11) does not change ε_i. In the case $D_i < \varepsilon_i^\star$, ε_i is increased and if $D_i > \varepsilon_i^\star$, ε_i is decreased. Due to the choice of the *sign* function, the ε change rate is independent of the actual value of the $D_i - \varepsilon_i^\star$ difference, i.e. $\mathrm{e}^{\pm\tau_\varepsilon}$. This ensures that large differences do not result in extreme ε_i changes. As an alternative one might replace the *sign* function by a sigmoid function, e.g. the hyperbolic tangent.

The choice of the parameter τ_ε, which may be interpreted as a damping constant, must be taken with care. The dynamics of the D_i and ε_i interfere with each other. As a

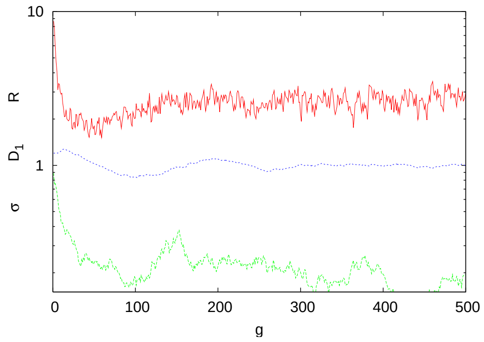

Fig. 2. The effect of ε control for a $(10/10_I, 100)$-ES. The desired actuator noise strength is $\varepsilon_i^\star = 1.0$. The top (red solid) curve represents the parental centroid distance $R = \|\langle \mathbf{x} \rangle\|$ to the optimizer, the middle (blue dotted) curve displays the measured parental standard deviation D_1, and the bottom (green) curve displays the mutation strength $\sigma = \langle \sigma \rangle$.

result, such a system can exhibit instabilities, e.g. oscillatory behavior. In order to prevent such instabilities, c_x and τ_ε must be chosen appropriately. While there is clearly a need for a thorough analysis, in the investigations done so far, the choice

$$\tau_\varepsilon = c_x/3 \tag{12}$$

worked flawlessly. Figure 2 shows the impact of ε-control on the actually realized actuator noise strength. As test function a sphere model $-\|\mathbf{x}\|^2$ (maximization!) with $N = 40$ and $\varepsilon_i^\star = 1.0$ has been chosen. One clearly sees that the control mechanism works well, D_i stabilizes around the target value 1.0 and no strong fluctuations occur.

3) How to Control the Population Size:

a) On the behavior of EAs under noise: It is well known that noise deteriorates the performance of the EAs. If the function to be optimized is noisy at its global or local optimizer, the EA cannot reach the optimizer in expectation. That is, the parental individuals are located in the long run (steady state behavior) in a certain (expected) distance to the optimizer, both in the object parameter space and usually also in the fitness space. For simple fitness models the expected final localization error can be estimated.

The effect of noise on the behavior of an ES optimizing a sphere model is that the parental population and its centroid are located in a certain expected distance R_∞ from the optimum. This can be verified in Fig. 2 (red curves) where the centroid distance R to the optimizer is displayed for a $(10/10_I, 100)$-ES evolving on a sphere model with actuator noise. In [7] the asymptotically exact formula for R_∞ has been calculated. Here we only show the approximation of R_∞ for $\sigma \ll \varepsilon$:

$$R_\infty \geq \frac{N\varepsilon}{\sqrt{8}\mu c_{\mu/\mu,\lambda}} \sqrt{1 + \sqrt{1 + \frac{8\mu^2 c^2_{\mu/\mu,\lambda}}{N}}}. \tag{13}$$

Equation (13) clearly shows that increasing the population size (assuming a constant truncation ratio $\vartheta = \mu/\lambda$) reduces the steady state residual distance to the optimizer. Therefore, increasing the population size is a means for locating the optimizer of the sphere with actuator noise arbitrarily precise. While it is an open issue whether population upgrading does always improve the final solution quality, one sees that – at least for the sphere model – a large population size may be desirable. However, if there is only weak noise, using a large population might be a waste of computer resources. Therefore, controlling the population size during the evolution can be a means to improve the efficiency of the EA.

b) A rule for controlling the population size.: In order to control the population size λ, a measure is needed which allows for a decision whether to decrease/increase λ. Assuming a stationary actuator noise distribution, the dynamics of the ES will (usually) approach a steady state behavior in a certain vicinity of the optimizer. That is, for a certain time period one observes on average a measurable improvement in the *observed* parental fitness values. If, however, one reaches the vicinity of the steady state, parental fitness will start to fluctuate around an average value. Therefore, if one observes on average no improvements of the fitness values from generation g to $g+1$, it is time to increase the population size.[3] The average parental fitness change ΔF is given by

$$\Delta F = \langle F \rangle^{(g)} - \langle F \rangle^{(g-1)}, \tag{14}$$

where $\langle F \rangle := \frac{1}{\mu} \sum_{m=1}^{\mu} \tilde{F}_{m;\lambda}$. Since ΔF itself is a strongly fluctuating quantity, an exponential smoothing should be used again to avoid unnecessary population increase due to random fluctuations. Algorithmically, this is done in Line 26, Fig 1, where c_f determines the time constant by which old $\overline{\Delta F}$-information vanishes exponentially fast. As a natural choice,

$$c_f = N^{-1} \tag{15}$$

can be used.

The population size control is realized in the algorithm in Fig. 1 through the Lines 25 to 30. Up to now we have considered the Lines 25 and 26. The λ update rule is realized in Lines 27 to 30. When considering maximization, desired fitness changes are of the kind $\overline{\Delta F} > 0$. Therefore, if $\overline{\Delta F} \leq 0$ the population size λ should be increased. Since the increased population does not necessarily change the sign of $\overline{\Delta F}$ in the next generation (random fluctuations!), the test of the update rule in Line 27 is performed every Δg-th generation. The λ-update itself, Line 29, is done via the μ-increase in Line 28. The new μ is obtained from the old μ using the change rate c_μ. Considering the sphere model, one can easily infer from (13) – assuming the equal sign in (13) – that the μ-asymptotic becomes

$$R_\infty \simeq \frac{\varepsilon N^{3/4}}{\sqrt{\sqrt{8} c_{\mu/\mu,\lambda}}} \frac{1}{\sqrt{\mu}}. \tag{16}$$

Therefore, increasing the population size multiplicatively results in an exponential decrease of R_∞ w.r.t. generation

[3] Note, we only consider the case of λ increase. A rule for λ-decrease has not been developed so far.

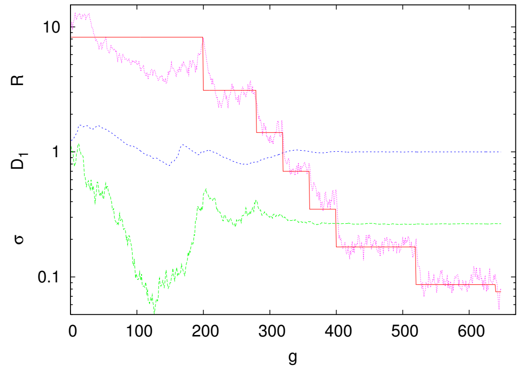

Fig. 3. On the effect of population size control used in an ES with $\vartheta = 0.4$ acting on a $N = 40$ sphere $-\|\mathbf{x}\|^2$ with actuator noise $\varepsilon_i^\star = 1$ and ε-control. The top (pink) dashed curve represents the parental centroid distance $R = \|\langle\mathbf{x}\rangle\|$ to the optimizer, the solid (red) staircase like curve represents the lower R_∞ expected value bound given by Eq. (13), the (blue) dotted curve displays the measured parental standard deviation D_1, and the (green) curve displays the mutation strength $\sigma = \langle\sigma\rangle$ (compare also Fig. 2). A total of approximately $5{,}000{,}000$ function evaluations has been used.

time interval Δg.[4]

There are three new exogenous strategy parameters to be fixed: The truncation ratio $\vartheta = \mu/\lambda$, the update time interval Δg, and the μ change rate c_μ. Up to now there is no theory for choosing these strategy parameters, however, simulations suggest that $\Delta g = N, c_\mu = 4$ represents a reasonable choice (see also Figs. 6 and 7). The truncation ratio for this setting should be in the interval $\vartheta = 0.4, \ldots, 0.6$.

Figure 3 provides a proof of concept for the proposed control rule used in an ES optimizing the sphere with actuator noise. As one can clearly see, the observed R dynamics follows closely the R_∞ curve derived from (13) using $\mu^{(g)}$ and $\lambda^{(g)}$ from the actual ES run. This also holds for the ES version *without* ε-control (the corresponding figure is not shown).

IV. ROCSAES – ROBUST-OPTIMIZATION-$(\mu/\mu_I, \lambda)$-CSA-ES

The cumulative step size adaptation (CSA) as well as its non-isotropic covariance matrix adaptation (CMA) [9] counterpart have recently been widely used in ES practice. Therefore, it seems reasonable to search for a version of this ES class that allows for robust optimization.

As a first step, we will consider the CSA-ES, i.e. the case of isotropic mutations only. Due to the special way how the mutation strength is determined in CSA-ES and how the offspring are generated using the *same* mutation strength σ for all offspring individuals, there seems to be no direct way to transfer the idea of direct robustness testing through mutations to the CSA-ES.[5] Therefore, we fall back to the black-box approach: The CSA-ES is applied without modifications to the function $f(\mathbf{x})$ which is *internally* disturbed by actuator noise of strength $\boldsymbol{\varepsilon}$. Therefore, we have to differentiate between the ES' individual vectors $\tilde{\mathbf{x}}_l$, generated in Line 10, Fig. 4, and the real actuator state $\tilde{\mathbf{x}}_l^{\text{a}}$, Line 11, entering the f function in Line 12. The $\tilde{\mathbf{x}}_l^{\text{a}}$ is invisible to the standard CSA-ES, however, it is needed for calculating the actually realized parental actuator fluctuations measured by the standard deviation D_i.

```
RO-$(\mu/\mu_I, \lambda)$-CSA-ES (Maximization)

$g := 0$;     $\sigma := \sigma^{(\text{init})}$;     $\langle\mathbf{x}\rangle^{(g)} := \mathbf{x}^{(\text{init})}$;          1
$\mu := \mu^{(\text{init})}$;     $\lambda := \lceil\mu/\vartheta\rceil$;     $\overline{\mathbf{x}^{\text{a}}} := \langle\mathbf{x}\rangle$;          2
For $i := 1$ To $N$; $\overline{x_i^{\text{a}2}} := \langle x_i^2\rangle$; End For          3
$\boldsymbol{\varepsilon} := \boldsymbol{\varepsilon}^\star$;          4
$\langle F\rangle^{(g)} := \frac{1}{\mu}\sum_{m=1}^{\mu} f\left(\mathbf{x}^{(\text{init})} + \boldsymbol{\mathcal{N}}_m(\mathbf{0}, \boldsymbol{\varepsilon}^\star)\right)$;          5
$\overline{\Delta F} := 0$;          6
$\mathbf{s} := \mathbf{0}$;          7
Repeat          8
    For $l := 1$ To $\lambda$          9
        $\tilde{\mathbf{x}}_l := \langle\mathbf{x}\rangle^{(g)} + \sigma\boldsymbol{\mathcal{N}}_l(\mathbf{0}, \mathbf{1})$;          10
        $\tilde{\mathbf{x}}_l^{\text{a}} := \tilde{\mathbf{x}}_l + \boldsymbol{\mathcal{N}}_l(\mathbf{0}, \boldsymbol{\varepsilon})$;          11
        $\tilde{F}_l := f(\tilde{\mathbf{x}}_l^{\text{a}})$;          12
    End For          13
    $g := g + 1$;          14
    $\langle\mathbf{x}\rangle^{(g)} := \frac{1}{\mu}\sum_{m=1}^{\mu}\tilde{\mathbf{x}}_{m;\lambda}$;          15
    $\mathbf{s} := (1 - c_\sigma)\mathbf{s} + \sqrt{(2 - c_\sigma)c_\sigma}\,\frac{\sqrt{\mu}}{\sigma}$
                $\left(\langle\mathbf{x}\rangle^{(g)} - \langle\mathbf{x}\rangle^{(g-1)}\right)$;          16
    $\sigma := \sigma\exp\left[\frac{\|\mathbf{s}\|^2 - N}{2Nd_\sigma}\right]$;          17
    For $i := 1$ To $N$          18
        $\langle x_i^{\text{a}}\rangle := \frac{1}{\mu}\sum_{m=1}^{\mu}(\tilde{\mathbf{x}}_{m;\lambda}^{\text{a}})_i$ ;          19
        $\langle x_i^{\text{a}2}\rangle := \frac{1}{\mu}\sum_{m=1}^{\mu}(\tilde{\mathbf{x}}_{m;\lambda}^{\text{a}})_i^2$ ;          20
        $\overline{x_i^{\text{a}}} := (1 - c_x)\overline{x_i^{\text{a}}} + c_x\langle x_i^{\text{a}}\rangle$;          21
        $\overline{x_i^{\text{a}2}} := (1 - c_x)\overline{x_i^{\text{a}2}} + c_x\langle x_i^{\text{a}2}\rangle$;          22
        $D_i := \sqrt{\overline{x_i^{\text{a}2}} - \overline{x_i^{\text{a}}}^2}$ ;          23
        $\varepsilon_i := \varepsilon_i\exp[\tau_\varepsilon\mathrm{Sign}(\varepsilon_i^\star - D_i)]$;          24
    End For          25
    $\langle F\rangle^{(g)} := \frac{1}{\mu}\sum_{m=1}^{\mu}\tilde{F}_{m;\lambda}$;          26
    $\overline{\Delta F} := (1 - c_f)\overline{\Delta F} + c_f\left(\langle F\rangle^{(g)} - \langle F\rangle^{(g-1)}\right)$;          27
    If $(\mathrm{Mod}(g, \Delta g) = 0) \wedge (\overline{\Delta F} \le 0)$ Then          28
        $\mu := \lceil\mu c_\mu\rceil$;          29
        $\lambda := \lceil\mu/\vartheta\rceil$;          30
    End If          31
Until Termination_Condition          32
```

Fig. 4. Pseudo code of the RO-$(\mu/\mu_I, \lambda)$-CSA-ES (ROCSAES).

The CSA specific part of the algorithm, Fig. 4, is located in Lines 16 and 17: the path cumulation and the σ update. Note, in contrast to the original CSA-ES [9], we have used a slightly different σ update rule proposed by Arnold in [6]. This simplifies the algorithm since there is no need to approximate the expected value of the χ-distribution. In order to perform the path cumulation, the cumulation time constant c_σ must be fixed. Two different recommendations concerning c_σ can be found in literature: $\propto 1/\sqrt{N}$ and $\propto 1/N$ (see e.g.

[4] Of course, this is bought at the price of exponentially increasing fitness evaluations per time period Δg.

[5] Note, in general we have to ensure different noise levels ε_i for the different coordinate directions. There is a certain hope that this could be accomplished by a modified CMA-ES, however, this is beyond the scope of this work.

[10], [9], [11]). From viewpoint of stability

$$c_\sigma = N^{-1} \tag{17}$$

should be chosen. According to experimental evidences [10] and theoretical analysis [11], the damping constant d_σ must be chosen depending on c_σ

$$d_\sigma = c_\sigma^{-1} \tag{18}$$

The rest of the algorithm in Fig. 4 is directly taken from the ROSAES, Fig. 1. The same holds for the recommended choice of the endogenous strategy parameters c_x, c_f, c_μ, and ϑ.

V. Experiments

The algorithms have been tested on various noisy test functions. Due to space limitations we will consider only a subset here.

A. Test functions

In the following experiments, we will use two test functions. Firstly, the standard sphere function with actuator noise, which we will denote as f_{asp} and which is defined as

$$f_{\text{asp}}(\mathbf{x}) := -\|\mathbf{x}+\boldsymbol{\delta}\|^2, \text{ where } \boldsymbol{\delta} \sim \varepsilon\boldsymbol{\mathcal{N}}(\mathbf{0},\mathbf{1}). \tag{19}$$

As has been shown in [12], the robust optimizer is at $\hat{\mathbf{x}} = \mathbf{0}$.

Secondly, we will run simulations on a function which is more complex and which belongs to the class of "functions with noise induced multi-modality - FNIMs". These functions exhibit an interesting bifurcation like behavior of their modalities under noise and are closer related to practical optimization problems than the sphere function, see [3], [13] for a comprehensive discussion. The function f_{fnim} is defined as

$$\begin{aligned} f_{\text{fnim}}(\mathbf{x}) \quad &:= \quad a - \frac{\sum_{i=1}^{N-1}(x_i+\delta_i)^2}{b+x_N^2} - x_N^2, \\ &\qquad b>0, \text{ where } \delta_i \sim \varepsilon\mathcal{N}_i(0,1). \end{aligned} \tag{20}$$

It has been introduced in [13] as an example function amenable to a theoretical analysis of the ES behavior. The conditional expectation $\mathrm{E}_{\boldsymbol{\delta}}[f(\mathbf{x},\boldsymbol{\delta})|\mathbf{x}]$ can be easily obtained from (20)

$$\begin{aligned} \mathrm{E}[f_{\text{fnim}}|\mathbf{x}] \quad &= \quad a - \frac{r^2+(N-1)\varepsilon^2}{x_N^2+b} - x_N^2, \\ &\qquad \text{where } r := \sqrt{\textstyle\sum_{i=1}^{N-1} x_i^2}. \end{aligned} \tag{21}$$

As has been shown in [13], given $r>0$, the local optimal x_N is at $(r_{\text{t}}^2 = b^2-(N-1)\varepsilon^2)$

$$\left.\begin{array}{ll} \tilde{x}_N = 0, & \text{for } r^2 \le r_{\text{t}}^2, \\ \tilde{x}_N = \pm\sqrt{\sqrt{r^2+(N-1)\varepsilon^2}-b}, & \text{for } r^2 > r_{\text{t}}^2. \end{array}\right\} \tag{22}$$

and the global optimizer becomes $(\varepsilon_{\text{t}} = b/\sqrt{N-1})$

$$\left.\begin{array}{ll} \hat{\mathbf{x}} = \mathbf{0}, & \text{for } \varepsilon \le \varepsilon_{\text{t}}, \\ \hat{\mathbf{x}} = \left(0,\ldots,0,\pm\sqrt{\sqrt{N-1}\,\varepsilon-b}\right)^{\mathrm{T}}, & \text{for } \varepsilon > \varepsilon_{\text{t}}. \end{array}\right\} \tag{23}$$

B. Results on f_{asp}

As we mentioned in the last section, EAs with actuator noise suffer from a residual error, which increases with the noise variance and decreases with the population size. This clearly indicates that the adaptive population sizing of the ROSAES and the ROCSAES should be superior to a standard σSA-ES or to a CSA-ES. As an example we have plotted in Fig. 5 the dynamics of the residual error of the four strategies versus the number of function evaluations for ESs optimizing the sphere model with actuator noise (19), $N = 40$, in Fig. 5. The standard ES and CSA-ES used the maximal parental population size $\mu = 320$ ($\vartheta = 0.4$) finally reached by ROSAES as basis for the comparison, i.e., they performed standard $(320/320_I, 800)$ strategies. As one can clearly see, increasing gradually the population size offers a performance gain. Furthermore, as to CSA-ES we see, however, that the standard strategy exhibits typical instabilities (the increase of the residual distance in the final phase of the run) when working with large population sizes.

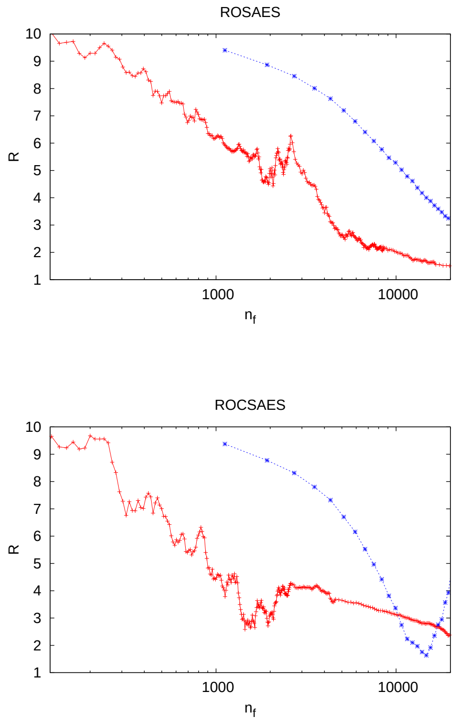

Fig. 5. Typical single run dynamics of the residual distance R to the optimizer state versus number of function evaluations n_f. Results are shown a) for the ROSAES and b) the ROCSAES on a sphere with actuator noise. The (blue, top curve)) $*$-data points are obtained from standard $(320/320_I, 800)$-ES and CSA-ES, respectively.

The rule for controlling the population size in both algorithms ROSAES and ROCSAES depends on two exogenous strategy parameters Δg and c_μ and on the truncation ratio ϑ. The performance (i.e. the residual distance to the optimizer state) of both algorithms is plotted against these parameters in Figs. 6 and 7. We observe that the residual distance

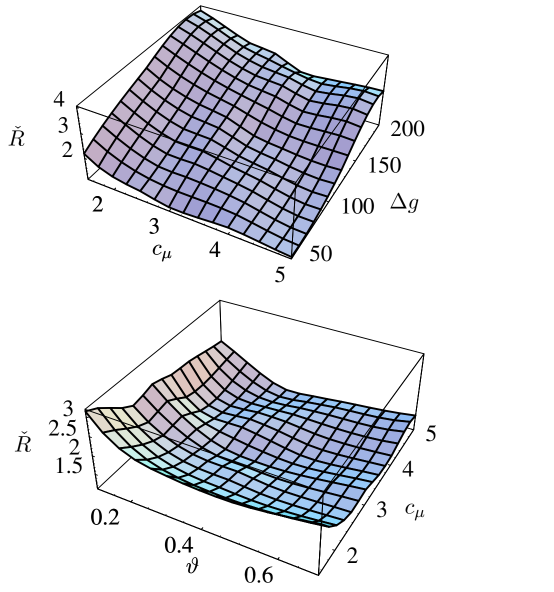

Fig. 6. Performance of ROSAES with $n_\mathrm{f} = 10,000$ on sphere, $N = 40$, with *actuator noise* of strength $\varepsilon = 1.0$. Truncation ratio ϑ of upper figure is $\vartheta = 0.4$, Δg of bottom figure is $\Delta g = N = 40$.

changes relatively smoothly with the strategy parameters and that no instabilities occur for both algorithms. At the same time, the ROSAES performs better than the ROCSAES. Extensive simulation studies have been carried out on various noisy test functions indicating that the tendencies shown in Fig. 6 seems to be generally valid for the ROSAES. As a rule of thumb, $\vartheta = 0.4$, $c_\mu = 4$, and $\Delta g = N$ is a reasonable choice for the strategy parameters. However, it should be also mentioned that ROCSAES can exhibit instabilities for very large population sizes ($\mu \overset{>}{\propto} N^2$ not shown in this paper).

C. Results on f_fnim

Figure 8 shows the optimization results depending on the actuator noise strength ε for $n_\mathrm{f} = 10,000$ function evaluations using ROSAES. As one can see in Figure 8, the ES is able to approximate the global optimizer curve (solid curve), given by Eq. (23). There is only a small influence of the truncation ratio ϑ. However, there remains a small approximation error w.r.t. x_N. This seems to be a finite N-dimensionality effect which has been observed in the analysis of the standard $(\mu/\mu_I, \lambda)$-σSA-ES, too. On the other hand, it has been shown in [13] that the asymptotically correct formula for the expected value of x_N using the standard $(\mu/\mu_I, \lambda)$-ES is given by

$$\mathrm{E}[x_N] = \pm\left(\left((N-1)\varepsilon^2\left[1+\frac{(N-1)\beta}{8\mu^2 c^2_{\mu/\mu,\lambda}}\right]\right)^{1/2} - b\right)^{1/2},$$

$$\beta = 1+\sqrt{1+\frac{8\mu^2 c^2_{\mu/\mu,\lambda}}{N-1}}. \qquad (24)$$

As one can easily see, increasing the population size in (24) such that $0 < \vartheta = \text{const.} < 1$ and $\lambda = \frac{\mu}{\vartheta} \to \infty$ yields

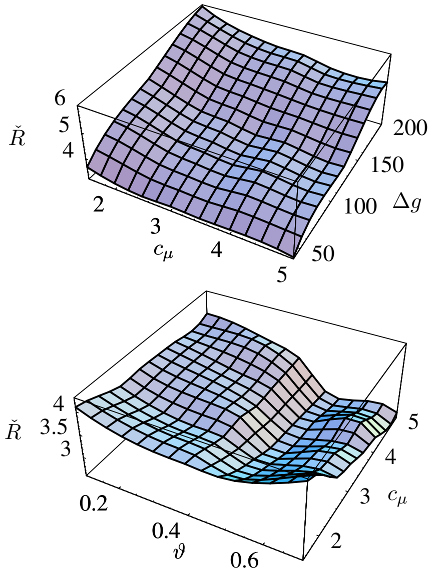

Fig. 7. Performance of ROCSAES with $n_\mathrm{f} = 10,000$ on sphere, $N = 40$, with *actuator noise* of strength $\varepsilon = 1.0$. Truncation ratio ϑ of upper figure is $\vartheta = 0.4$, Δg of bottom figure is $\Delta g = N = 40$.

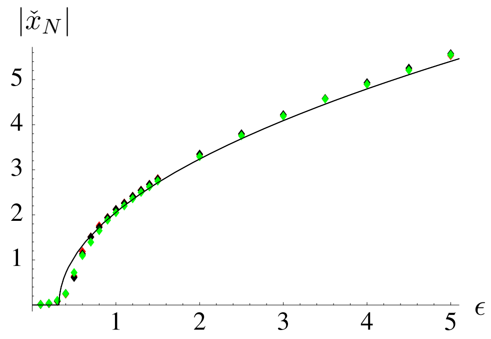

Fig. 8. Optimization results of ROSAES on the x_N-coordinate of function f_fnim, $N = 40$, $a = 5.0$, $b = 2.0$, depending on the actuator noise strength ε. Population update time $\Delta g = N = 40$ and $n_\mathrm{f} = 10,000$. As truncation ratios $\vartheta = 0.3$ (red data points), $\vartheta = 0.4$ (black data points), and $\vartheta = 0.5$ (green data points) have been chosen (data points are partially concealed). The curve presents the x_N value of the global optimizer (23).

the global optimizer of x_N, Eq. (23). This is in accordance with simulations (not shown here). Furthermore, considering the length r, Eq. (5), of the first $N-1$ components of the solution vector, the simulations show that it decreases monotonously with the number n_f of function evaluations.[6] Therefore, we can conclude that using ROSAES on f_fnim yields asymptotically the global optimizer.

Considering the performance of ROCSAES in Fig. 9 we see larger deviations compared to ROSAES, Fig. 8. This deviation can be slightly decreased by increasing the number of function evaluations to $n_\mathrm{f} = 50,000$. However, increasing n_f further results in instable behavior. The reason for this disappointing observation can be clearly addressed to the

[6] Recall, the global optimizer is at $r = 0$, Eq. (23). See also the R-dynamics on the sphere with actuator noise in Fig. 3.

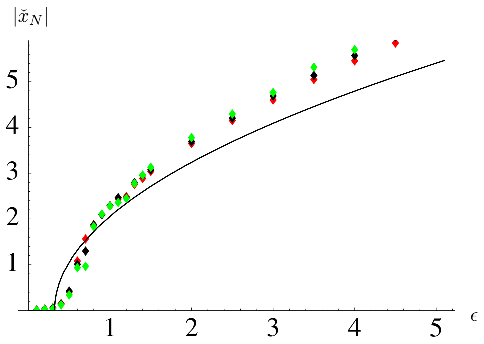

Fig. 9. Optimization results of ROCSAES on the x_N-coordinate of function f_4, $N = 40$, $a = 5.0$, $b = 2.0$, depending on the actuator noise strength ε. Population update time $\Delta g = N = 40$ and $n_\mathrm{f} = 10,000$. As truncation ratios $\vartheta = 0.3$ (red data points), $\vartheta = 0.4$ (black data points), and $\vartheta = 0.5$ (green data points) have been chosen (data points are partially concealed). The curve presents the x_N value of the global optimizer (23).

large population sizes. It is an interesting side result of the investigations performed: CSA-ES becomes increasingly instable (divergence or chaotic behavior of the mutation strength σ) for large population sizes.

VI. Conclusion

In this paper, we have proposed two ESs for robust (noisy) optimization that solve two problems standard ESs (as well as other EAs) are suffering from: a) Due to selection, the actually observed actuator noise variance does usually deviate from the target variance and b) a fixed population size results in a final expected deviation from the real robust optimizer no matter how long the EA runs. The solutions are actuator noise control and population size control. Furthermore, it might come as a surprise, but standard self-adaptive ESs (but not CSA-ES) utilize the mutations itself as robustness tester. From this point of view, standard σ-self-adaptive ESs are strategies for robust optimization *per se*. In [14] a similar idea has been offered in context of evolvable hardware. In our work, we have shown that such a concept does really work.

From viewpoint of performance, we can conclude that the robust evolution strategies with the newly proposed control rules outperform their standard counterparts, i.e. mutative evolution strategy σSA-ES and cumulative step-size adaptation CSA-ES. Furthermore, the ROSAES is superior and more stable compared to the ROCSAES. Although we have not yet analyzed in depth why this might be the case, we expect that the reason is the difficulty of the CSA-ES with increasing population sizes.

The robust evolution strategies depend on a number of exogenous strategy parameters that must be set manually. Although the optimal choice of these parameters will depend on the problem (i.e. no globally optimal set can be given), we have shown that the performance changes smoothly with the new strategy parameters and that no instabilities occur.

The ROSAES performs well also on more complex test functions, like the function f_fnim. The performance of the ROCSAES is usually inferior. Therefore, from the current results we rather recommend the use of ROSAES for problems with actuator noise. It is quite clear that the proposed strategies will also have their drawbacks. Experiments that have not been included in this paper show that there are also FNIM functions where the approach to the robust optimizer depends on the truncation ratio. Therefore, our future research will aim at a deeper understanding of the behavior of ROSAES and ROCSAES in robust optimization scenarios.

References

[1] J. Branke, *Evolutionary Optimization in Dynamic Environments*. Dordrecht: Kluwer Academic Publishers, 2002.

[2] Y. Jin and J. Branke, “Evolutionary Optimization in Uncertain Environments – A Survey,” *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.

[3] B. Sendhoff, H.-G. Beyer, and M. Olhofer, “The influence of stochastic quality functions on evolutionary search,” in *Recent Advances in Simulated Evolution and Learning*, ser. Advances in Natural Computation, K. Tan, M. Lim, X. Yao, and L. Wang, Eds. World Scientific, 2004, pp. 152–172.

[4] S. Tsutsui, “A comparative study on the effects of adding perturbations to phenotypic parameters in genetic algorithms with a robust solution searching scheme,” in *Proceedings of the 1999 IEEE System, Man, and Cybernetics Conference – SMC’99*, vol. 3. IEEE, 1999, pp. 585–591.

[5] J. Fitzpatrick and J. Grefenstette, “Genetic Algorithms in Noisy Environments,” in *Machine Learning: Special Issue on Genetic Algorithms*, P. Langley, Ed. Dordrecht: Kluwer Academic Publishers, 1988, vol. 3, pp. 101–120.

[6] D. Arnold, *Noisy Optimization with Evolution Strategies*. Kluwer Academic Publishers, 2002.

[7] H.-G. Beyer, M. Olhofer, and B. Sendhoff, “On the impact of systematic noise on the evolutionary optimization performance - a sphere model analysis,” *Genetic Programming and Evolvable Machines*, vol. 5, no. 4, pp. 327–360, 2004.

[8] L. Grünz and H.-G. Beyer, “Some observations on the interaction of recombination and self-adaptation in evolution strategies,” in *Congress on Evolutionary Computation CEC*. IEEE Press, 1999, pp. 639–645.

[9] N. Hansen and A. Ostermeier, “Completely Derandomized Self-Adaptation in Evolution Strategies,” *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[10] N. Hansen, “Verallgemeinerte individuelle Schrittweitenregelung in der Evolutionsstrategie,” Doctoral thesis, Technical University of Berlin, Berlin, 1998.

[11] D. Arnold and H.-G. Beyer, “Performance analysis of evolutionary optimization with cumulative step length adaptation,” *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 617–622, 2004.

[12] H.-G. Beyer, M. Olhofer, and B. Sendhoff, “On the behavior of $(\mu/\mu_I, \lambda)$-ES optimizing functions disturbed by generalized noise,” in *Foundations of Genetic Algorithms VII*, K. de Jong, R. Poli, and J. Rowe, Eds. Morgan Kaufmann, 2002, pp. 307–328.

[13] H.-G. Beyer and B. Sendhoff, “Functions with Noise-Induced Multi-Modality: A Test for Evolutionary Robust Optimization – Properties and Performance Analysis,” *IEEE Transactions on Evolutionary Computation*, 2005, accepted.

[14] A. Thompson and P. Layzell, “Evolution of Robustness in an Electronics Design,” in *Evolvable Systems: From Biology to Hardware, Proceedings of the Third International Conference (ICES-2000)*, ser. LNCS, J. Miller, A. Thompson, P. Thomson, and T. Fogarty, Eds., vol. 1801. Berlin: Springer Verlag, 2000, pp. 218–228.