Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)

Evolutionary Algorithms in the Presence of Noise: To Sample or Not to Sample

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

***Abstract*— In this paper, we empirically analyze the convergence behavior of evolutionary algorithms (evolution strategies – ES and genetic algorithms – GA) for two noisy optimization problems which belong to the class of functions with noise induced multi-modality (FNIMs). Although, both functions are qualitatively very similar, the ES is only able to converge to the global optimizer state for one of them. Additionally, we observe that canonical GA exhibits similar problems. We present a theoretical analysis which explains the different behaviors for the two functions and which suggests to resort to resampling strategies to solve the problem. Although, resampling is an inefficient way to cope with noisy optimization problems, it turns out that depending on the properties of the problem, (moderate) resampling might be necessary to guarantee convergence to the robust optimizer.**

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 $\mathbf{y}$, evaluating its quality $f(\mathbf{y})$ yields stochastic quantity values. As a result, an optimization algorithm applied to $f(\mathbf{y})$ must deal with these uncertain quality information and it must use this information to calculate a robust optimum based on an appropriate robustness measure.

In this paper, we will use a robustness measure which is based on the expectation value of the objective function. The robust optimum is given by

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \left[\mathrm{E}[f|\mathbf{y}]\right]. \tag{1}$$

We will apply the fitness measure, Eq. (1), to the optimization of two functions which at first glance seem very similar. They both belong to the class of functions with noise induced multi-modality and qualitatively their corresponding fitness landscapes can hardly be distinguished. These functions will be introduced and discussed in the next section. In section III, we will see that the convergence behavior of the evolution strategy is, however, very different for both functions. Furthermore, we also present some empirical results for the canonical genetic algorithm for one of the two functions. In section IV, we present theoretical reasons for the behavior of the evolution strategy and finally, we resort to the resampling technique in section V in order to improve the search behavior. In the last section, we will conclude the paper.

II. Test Functions with Noise Induced Multi-Modality

Functions with Noise Induced Multi-Modality (FNIMs) have first been introduced in [1] and analyzed in [2]. They belong to the class of topology changing functions under the influence of noise [3]. This class of functions was motivated qualitatively from observations of practical design optimization problems, as e.g. reported in [4]. These functions are typically unimodal without the influence of noise and undergo a process that has been termed bifurcation if the noise level reaches a certain threshold the value of which is determined by other parameters of the function. The following two functions are typical examples of this class. Although they are functionally very similar, we will see in the next section, that they are quite different with respect to the convergence behavior of the evolution strategy. The first is

$$f_2(\mathbf{y}) := -\frac{(y_{N-1}+\delta)^2 + \sum_{i=1}^{N-2} y_i^2}{y_N^2+b} - y_N^2, \qquad \text{where} \quad b>0 \quad \text{and} \quad \delta \sim \varepsilon\mathcal{N}(0,1). \tag{2}$$

The conditional expectation needed in (1) becomes

$$\mathrm{E}[f_2|\mathbf{y}] = -\frac{r^2+\varepsilon^2}{y_N^2+b} - y_N^2, \;\; \text{where} \;\; r := \sqrt{\sum_{i=1}^{N-1} y_i^2}. \tag{3}$$

In [3] it has been shown that, given $r>0$, the local optimal y_N is at

$$\left.\begin{array}{ll} \tilde{y}_N = 0, & \text{for} \quad r^2 \le b^2-\varepsilon^2 \\ \tilde{y}_N = \pm\sqrt{\sqrt{r^2+\varepsilon^2}-b}, & \text{for} \quad r^2 > b^2-\varepsilon^2 \end{array}\right\} \tag{4}$$

and the robust global optimum is at

$$\left.\begin{array}{ll} \hat{\mathbf{y}} = \mathbf{0}, & \text{for} \;\; \varepsilon \le b, \\ \hat{\mathbf{y}} = \left(0,\ldots,0,\pm\sqrt{\varepsilon-b}\right)^{\mathrm{T}}, & \text{for} \;\; \varepsilon > b \end{array}\right\}. \tag{5}$$

For $N=2$ the conditional expectation of f_2 is shown in Figure 1 for two different ε corresponding to the two cases unimodal and bimodal in Eq. (5).
Function f_4 is defined as

$$f_4(\mathbf{y}) := -\frac{\sum_{i=1}^{N-1}(y_i+\delta_i)^2}{y_N^2+b} - y_N^2, \qquad \text{where} \quad b>0, \quad \text{and} \quad \delta_i \sim \varepsilon\mathcal{N}_i(0,1), \tag{6}$$

with the conditional expectation

$$\mathrm{E}[f_4|\mathbf{y}] = -\frac{r^2+(N-1)\varepsilon^2}{y_N^2+b} - y_N^2. \tag{7}$$

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).

1-4244-0703-6/07/$20.00 ©2007 IEEE

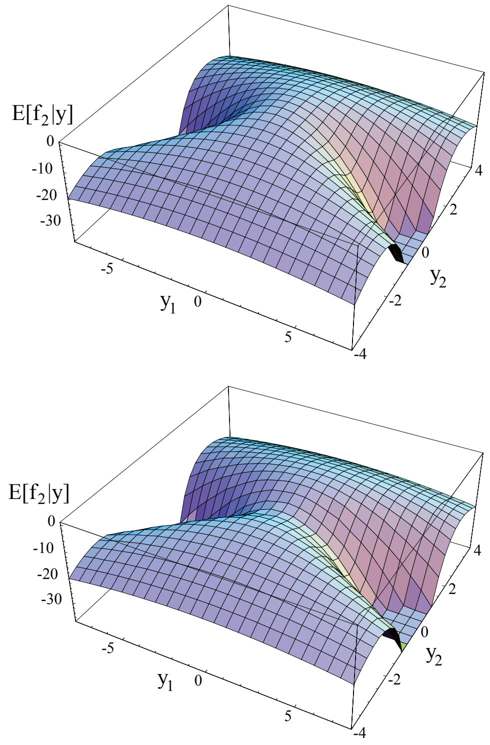

Fig. 1. Conditional expectation $\mathrm{E}[f_2|\mathbf{y}]$ of function f_2 for $N = 2$; $b = 1$, $\varepsilon = 0.1$ (upper figure) and $\varepsilon = 3$ (lower figure). For $\varepsilon \le b$ function f_2 is unimodal (upper figure) and for $\varepsilon > b$ bimodal (lower figure).

As has been shown in [2], given $r > 0$, the locally optimal y_N is ($r_{th} = b^2 - (N-1)\varepsilon^2$)

$$\left.\begin{array}{ll}\tilde{y}_N = 0, & \text{for } r^2 \le r_{th}\\ \tilde{y}_N = \pm\sqrt{\sqrt{r^2+(N-1)\varepsilon^2}-b}, & \text{for } r^2 > r_{th}\end{array}\right\} \tag{8}$$

and the global optimizer is given by ($\varepsilon_{th} = b/\sqrt{N-1}$)

$$\left.\begin{array}{ll}\hat{\mathbf{y}} = \mathbf{0}, & \text{for } \varepsilon \le \varepsilon_{th}\\ \hat{\mathbf{y}} = \left(0,\ldots,0,\pm\sqrt{\sqrt{N-1}\,\varepsilon-b}\right)^{\mathrm{T}}, & \text{for } \varepsilon > \varepsilon_{th}\end{array}\right\} \tag{9}$$

Figures for function f_4 are not included because they are qualitatively identical to Figure 1.

III. Empirical Analysis

A. Outline of the Evolution Strategy

In this work, we only consider the evolutionary self-adaptation strategy for the control of the mutation strength σ of the isotropic Gaussian mutation operator. Empirical investigations [5] as well as theoretical considerations [6] have shown, that the alternative – the cumulative step size adaptation (CSA) proposed by Ostermeier et al. [7], [8] – does not work well in highly noisy environments: Either the CSA exhibits premature convergence (for small population sizes), or it exhibits instable and divergent behavior (especially for population sizes much larger than the search space dimensionality). In contrast to CSA, the σ-self-adaptive $(\mu/\mu,\lambda)$-ES (σSA-ES) works stable and without premature convergence, provided that one uses moderate truncation ratios (one should avoid using too small truncation ratios μ/λ).

The σSA-ES used is based on the coupled inheritance of object and strategy parameters. Using the notation

$$\langle\mathbf{v}\rangle^{(g)} := \frac{1}{\mu}\sum_{m=1}^{\mu}\mathbf{v}_{m;\lambda}^{(g)} \tag{10}$$

for intermediate recombination, which is basically an averaging over the $\mathbf{v}$ parameters of the best μ offspring individuals (the "$m;\lambda$"-notation means "index of the $m$best individual" taken from the pool of λ offspring individuals of generation g, "$^{(g)}$" is the generation counter), the $(\mu/\mu_I,\lambda)$-σSA-ES iterates an evolution loop

$$\forall l = 1,\ldots,\lambda:\ \left\{\begin{array}{ll}\sigma_l^{(g+1)} & := \langle\sigma\rangle^{(g)}\mathrm{e}^{\tau\mathcal{N}_l(0,1)}\\ \mathbf{y}_l^{(g+1)} & := \langle\mathbf{y}\rangle^{(g)} + \sigma_l^{(g+1)}\boldsymbol{\mathcal{N}}_l(\mathbf{0},\mathbf{1}).\end{array}\right. \tag{11}$$

Each offspring individual (indexed by l) gets its individual mutation strength σ. This σ is used as individual mutation parameter for isotropically producing the offspring's object parameter using a (Gaussian) normally distributed random vector $\boldsymbol{\mathcal{N}}$ with zero mean.

The mutation of the mutation strength is done by multiplication with a log-normally distributed random variate in (11) using the distribution parameter τ also referred to as learning parameter. We use the standard choice $\tau = 1/\sqrt{N}$ throughout the simulations.

Optimization of noisy objective functions with ES requires the use of large populations. However, in order to use computer resources efficiently as possible, it is best to start with small populations sizes λ and increase λ successively during the evolution (keeping the truncation ratio μ/λ constant). There are different strategies to implement such a population growth. The strategy employed in the experiments used moving averages of the observed fitnesses to decide when to increase the population size by a constant factor, see [9].

B. Empirical Evaluation of $(\mu/\mu,\lambda)$-ES on f_2

Due to the population growth mechanism [9] incorporated in the standard σSA-ES, the ES should be able to approximate the robust optimizer of the test function f_2 arbitrarily exact provided that the ES is allowed to increase the population size sufficiently. From the theory developed in [2] this should hold for f_4 in any case since this theory is exact for $N \to \infty$: The theory predicts the expected steady state r^2

$$\text{defining}\qquad \xi_1 = 8\mu^2 c_{\mu/\mu,\lambda}^2$$

$$\xi_2 = 1 + \sqrt{1 + \frac{\xi_1}{N-1}}$$

$$\mathrm{E}[r^2] = \frac{(N-1)^2\varepsilon^2}{\xi_1}\xi_2 \tag{12}$$

and the expected steady state y_N

$$\mathrm{E}[y_N] = \begin{cases} 0, & \text{for} \quad \frac{b^2}{(N-1)\varepsilon^2} \geq 1 + \frac{N-1}{\xi_1}\xi_2, \\ \pm\sqrt{\sqrt{(N-1)\varepsilon^2\left[1 + \frac{N-1}{\xi_1}\xi_2\right]} - b}, & \text{else.} \end{cases} \tag{13}$$

Assuming a fixed truncation ratio

$$\vartheta := \frac{\mu}{\lambda} = \text{const.}, \qquad 0 < \vartheta < 1, \tag{14}$$

one can see that

$$\mathrm{E}[r^2] \xrightarrow{\lambda\to\infty} 0 \qquad \text{and} \qquad \mathrm{E}[y_N] \xrightarrow{\lambda\to\infty} \hat{y}_N. \tag{15}$$

This can be confirmed by experiments using $N < \infty$ sufficiently large. In Figure 2 one sees that the ES is able to approximate the global optimizer (9) well. Considering the dynamics of the ES on f_2, one observes a similar behavior for the aggregated r, it evolves toward 0. However, as to the steady state of y_N, one observes a significant deviation from $\hat{y}_N$ in the experiment presented in Fig. 3. Considering only the final state of y_N, one might speculate whether this is due to premature convergence of the ES. However, a closer look at the recombined mutation strength $\langle\sigma\rangle^{(g)}$ reveals that the mutation strength is considerably above zero. Also, the y_N dynamics crosses the global optimizer state $\hat{y}_N$. That is, even if the strategy were initialized in the vicinity of $\hat{y}_N$, this state appears not to be an attractor state. In the experiment presented in Fig. 3, the final (expected) steady state y_N appears to be less than the global optimizer $\hat{y}_N$. This is, however, not a general tendency. Actually, one can "control" the steady state by tuning the truncation ratio ϑ. This is in surprising contrast to the behavior of the ES on f_4. Figure 4 shows the influence of the truncation ratio ϑ on the final y_N steady state for f_2 (such a dependence on the truncation ratio is not observed for f_4). As one can see, by choosing the "correct" ϑ one can get close to the global optimizer states. However, the correct ϑ is not known a priori. Therefore, it cannot be used to build up a reliably working ES for robust optimization.

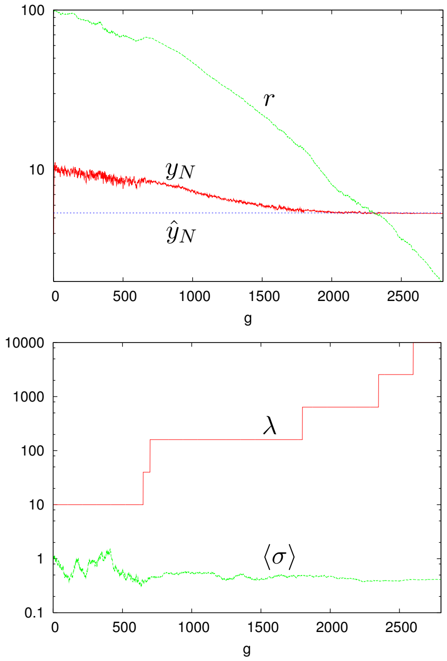

Fig. 2. Dynamics of the evolution of the σSA-ES on f_4. The function specific parameters are $b = 1$, $\varepsilon = 3$, and $N = 100$. Truncation ratio used is: $\vartheta = 0.4$. Top picture: the horizontal (dashed blue) line represents the global optimizer state $\hat{y}_N$ of the object parameter y_N given by Eq. (9). The bottom picture displays the mutation strength σ and the population size λ dynamics.

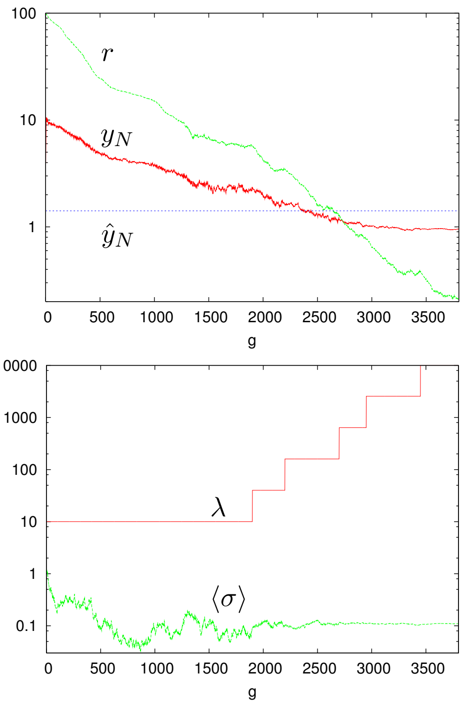

Fig. 3. Dynamics of the evolution of the σSA-ES on f_2. The function specific parameters are $b = 1$, $\varepsilon = 3$, and $N = 100$. Truncation ratio used is: $\vartheta = 0.4$. The horizontal (dashed blue) line in the upper figure represents the global optimizer state $\hat{y}_N$ of the object parameter y_N given by Eq. (5), respectively. We can observe that the steady state y_N departs from the global optimizer state $\hat{y}_N$. The bottom picture shows that the mutation strength σ is considerably above zero (i.e., no premature convergence).

C. Outline of the GA

Although this paper clearly concentrates on the performance and analysis of evolution strategies, we want to present some empirical results for genetic algorithms as well. The main reason being that Tsutsui and Gosh [10], [11] motivated for their *genetic algorithms with a robust solution searching scheme (GAs/RS)* the use of larger population sizes instead of resampling. Therefore, we were curious to see whether GAs/RS would also exhibit convergence problems on function f_2 or not.

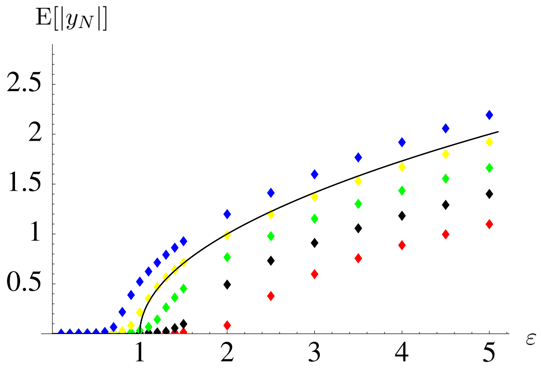

Fig. 4. Mean steady state y_N values of ES runs on test function f_2 with $N = 40$ and $b = 1.0$ depending on noise strength ε for ES with *different* truncation ratios ϑ. The data points presented are averages obtained from 20 independent runs each. The truncation ratios displayed are (from bottom to top) $\vartheta = 0.3$ (red data points), $\vartheta = 0.4$ (black), $\vartheta = 0.5$ (green), $\vartheta = 0.6$ (yellow), and $\vartheta = 0.7$ (blue), data points are partially overlapping. The curve presents the y_N values of the global optimizer $\hat{y}_N$ as given by Eq. (5).

In [10] the authors analyze the population average $\bar{f}$ under the influence of (actuator) noise because in the schema theorem the expected number of schemata depends on the relation between average schema fitness and average population fitness (this corresponds to fitness-proportional selection)

$$\bar{f} = \frac{1}{\lambda}\sum_{i=1}^{\lambda} f(\mathbf{y}_i, \delta_i) \tag{16}$$

Under the assumption of a continuous parameter space and infinite population size $\lambda \to \infty$, we can write with $q(\delta)$ and $p(\mathbf{y}, t)$ being the probability distributions of the noise (in our example $\delta \sim \varepsilon\mathcal{N}(0, 1)$) and of the parameter $\mathbf{y}$ in the population at time t, respectively,

$$\begin{aligned}\bar{f} &= \int_{\mathbf{y}}\int_{\delta} f(\mathbf{y}, \delta)\, p(\mathbf{y}, t)\, q(\delta)\, \mathrm{d}\delta\, \mathrm{d}\mathbf{y} \\ &= \int_{\mathbf{y}} \mathrm{E}[f|\mathbf{y}]\, p(\mathbf{y}, t)\, \mathrm{d}\mathbf{y}. \end{aligned} \tag{17}$$

Equation (17) shows that the average number of instances of schemata increases/decreases depending on $\mathrm{E}[f|\mathbf{y}]$ instead of $f(\mathbf{y})$. Thus, although Eq. (16) does not use explicit sampling, the canonical genetic algorithm works implicitly on the expected fitness. Note, that this derivation is only valid for fitness-proportional selection on which the schema theorem is based.

D. Empirical Evaluation of a GA with Fitness-Proportional Selection

The empirical results for the canonical GA [12] with a 10 bit binary representation for each of the $N = 40$ parameters, fitness-proportional selection (as required by the analysis of [10]), and uniform crossover with rate $p_c = 1$ are shown in Figure 5. The presented results show the best performance over several trials with different mutation rates. Even for large population sizes (10^3 in the upper figure and 10^4 in the middle figure) the global optimizer state cannot be reached. Having a closer look, the results are even more disappointing: the quality improvement by increasing the population size by a factor of 10 (i.e., from 10^3 to 10^4) reduces the error by just 20%.

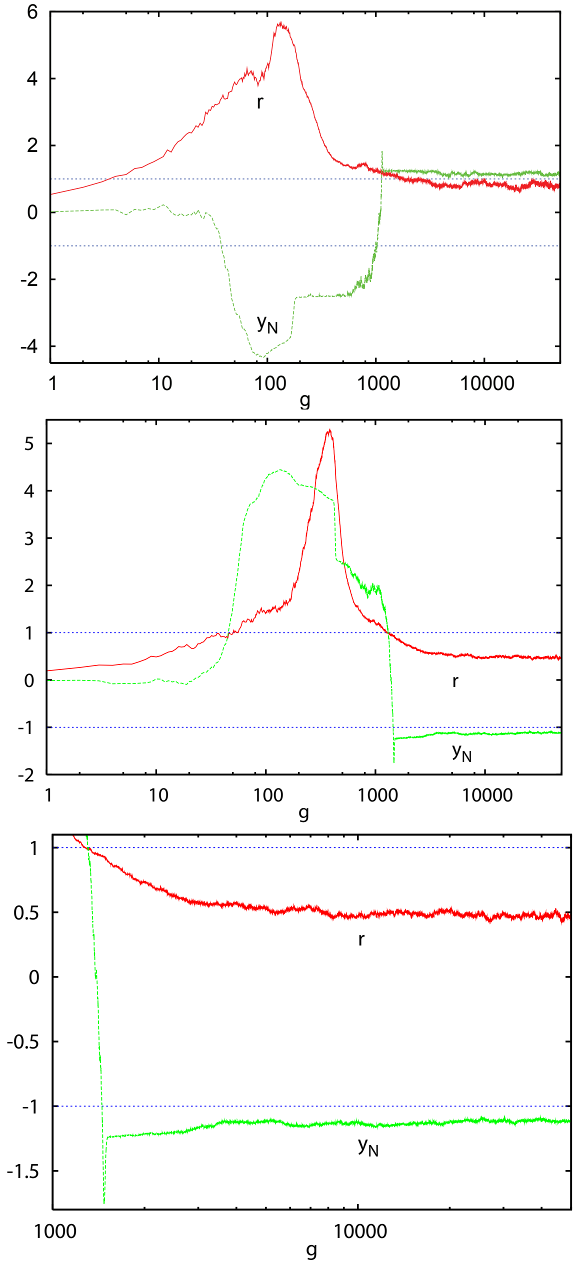

Fig. 5. Dynamics of the evolution of a binary-coded GA on f_2. The function specific parameters are $b = 1$, $\varepsilon = 2$, and $N = 40$. The horizontal (blue dotted) lines represent the global optimizer states. The green curve shows the y_N values and the red curve the distance of the remaining $N-1$ components to the optimizer state. A 10-bit representation has been used for each parameter. The population size has been set to 10^3 in the upper figure (mutation rate $p_m = 10^{-4}$) and 10^4 in the middle figure (mutation rate $p_m = 5 \cdot 10^{-5}$). The lower figure shows the dynamics for the population size of 10^4 from generation 1000 onwards.

As an alternative, we also consider real-coded GAs [13], [14], [15]. However, it was interesting to note that the performance of real-coded GAs with *fitness-proportional* selection is rather worse than that of binary-coded GAs. Furthermore, they appear extremely sensitive to the choice of the crossover parameters, e.g. α in BLX-α [13]. Therefore, graphs have not been included here.

We can summarize that our empirical results do not support the conclusion by Tsutsui and Gosh [10] that one can approximate the optimizer state of a noisy optimization problem arbitrarily close with large enough populations.

IV. Linear Analysis

A. General Considerations

As one can infer from the $y_N^{(g)}$ plots in Figs. 2 and 3, the respective robust optimizer state $\hat{y}_N$ seems to be an attractor state for f_4 but not for f_2. In order to understand this observation, one should consider the noisy fitness landscape in the vicinity of the robust optimizer state. To this end, it suffices to consider small perturbations (or mutations) $\mathbf{z}$ of $\hat{\mathbf{y}}$. Using Taylor expansion we generally have

$$f(\hat{\mathbf{y}} + \mathbf{z}|\boldsymbol{\delta}) = f(\hat{\mathbf{y}}|\boldsymbol{\delta}) + \sum_{i=1}^{N} \left.\frac{\partial f(\mathbf{y}|\boldsymbol{\delta})}{\partial y_i}\right|_{\mathbf{y}=\hat{\mathbf{y}}} z_i + \dots \tag{18}$$

and the *observed* fitness deviation Δf from the robust optimizer state $\hat{\mathbf{y}}$ becomes

$$\Delta f = f(\hat{\mathbf{y}} + \mathbf{z}|\boldsymbol{\delta}) - f(\hat{\mathbf{y}}|\boldsymbol{\delta}) = \sum_{i=1}^{N} \left.\frac{\partial f(\mathbf{y}|\boldsymbol{\delta})}{\partial y_i}\right|_{\mathbf{y}=\hat{\mathbf{y}}} z_i + \dots \tag{19}$$

Now assume the ES at generation g in the parental (centroid) state $\langle \mathbf{y} \rangle = \hat{\mathbf{y}}$ and consider (small) arbitrary mutations $\mathbf{z}$. Unless $\left.\frac{\partial f(\mathbf{y}|\boldsymbol{\delta})}{\partial y_i}\right|_{\mathbf{y}=\hat{\mathbf{y}}} \equiv 0$, the ES will generate states $\Delta f \neq 0$. Since maximization is considered (a similar arguments holds for minimization), the ES selects those $\mathbf{z}$ mutations which are associated with $\Delta f > 0$. As a result, the parental $\hat{\mathbf{y}}$ will be left. Conversely, the condition

$$\left.\frac{\partial f(\mathbf{y}|\boldsymbol{\delta})}{\partial y_i}\right|_{\mathbf{y}=\hat{\mathbf{y}}} = 0 \tag{20}$$

guarantees *local* stability of the robust optimizer state $\hat{\mathbf{y}}$. However, while $\left.\frac{\partial f(\mathbf{y}|\boldsymbol{\delta})}{\partial y_i}\right|_{\mathbf{y}=\hat{\mathbf{y}}} \neq 0$ is the reason for the departure of selected offspring states from the parental state $\hat{\mathbf{y}}$, the *expected steady state* of the ES (i.e. averaging the parental states over a long time period) can still be at $\hat{\mathbf{y}}$. This is so because $\frac{\partial f(\mathbf{y}|\boldsymbol{\delta})}{\partial y_i}$ is a random vector due to $\boldsymbol{\delta}$. Therefore, depending on the distribution of $\boldsymbol{\delta}$ and the ES selection operator used, the expected value of the $(\mu/\mu_I, \lambda)$-ES specific Δf can vanish. This is what has been observed in Fig. 4 for some certain ε values and the truncation ratio $\vartheta = 0.6$.

B. Test Function f_2

We will now apply the general framework from above to f_2. The first order derivatives are

$$i = 1, \dots, N-2: \quad \frac{\partial f_2}{\partial y_i} = -2\frac{y_i}{y_N^2 + b}, \tag{21}$$

$$\frac{\partial f_2}{\partial y_{N-1}} = -2\frac{y_{N-1} + \delta}{y_N^2 + b}, \tag{22}$$

$$\frac{\partial f_2}{\partial y_N} = 2y_N \left(\frac{(y_{N-1} + \delta)^2 + \sum_{i=1}^{N-2} y_i^2}{(y_N^2 + b)^2} - 1 \right). \tag{23}$$

Using (5) we obtain immediately

$$\varepsilon \le b: \quad \forall i \neq N-1 \quad : \quad \left.\frac{\partial f(\mathbf{y}|\delta)}{\partial y_i}\right|_{\mathbf{y}=\mathbf{0}} = 0$$

$$\left.\frac{\partial f(\mathbf{y}|\delta)}{\partial y_{N-1}}\right|_{\mathbf{y}=\mathbf{0}} = -2\frac{\delta}{b} \tag{24}$$

and for $\varepsilon > b$ using $\hat{y}_N^2 + b = \varepsilon$ one gets

$$\varepsilon > b: \quad \forall i, \dots, N-2: \quad \left.\frac{\partial f(\mathbf{y}|\delta)}{\partial y_i}\right|_{\mathbf{y}=\hat{\mathbf{y}}} = 0, \tag{25}$$

$$\varepsilon > b: \quad \left.\frac{\partial f(\mathbf{y}|\delta)}{\partial y_{N-1}}\right|_{\mathbf{y}=\hat{\mathbf{y}}} = -2\frac{\delta}{\varepsilon}, \tag{26}$$

$$\varepsilon > b: \quad \left.\frac{\partial f(\mathbf{y}|\delta)}{\partial y_N}\right|_{\mathbf{y}=\hat{\mathbf{y}}} = \pm 2\sqrt{\varepsilon - b}\left(\frac{\delta^2}{\epsilon^2} - 1\right). \tag{27}$$

As for the case $\varepsilon \le b$, Eq. (24), one sees that the derivative of the Nth component vanishes, therefore, the system is locally stable w.r.t. fluctuations in the y_N direction. But, even for this case there is the y_{N-1} component the derivative of which is not equal to zero. This results in a certain tendency to leave the $\hat{\mathbf{y}}$ state, however, due to $\delta \sim \varepsilon\mathcal{N}(0,1)$ (the noise model considered), the influence via the y_{N-1}-component gets smaller for smaller ε. This can be qualitatively confirmed in the plots of Fig. 4.

In the case $\varepsilon > b$, the variance of the Nth component (27) increases monotonously with ε (while the expected value remains zero). That is, one cannot expect that the ES is able to approximate the robust optimizer $\hat{\mathbf{y}}$ (5) of f_2 arbitrarily exact. This raises the question why the ES works on f_4.

C. Test Function f_4

The first order derivatives are

$$i = 1, \dots, N-1: \quad \frac{\partial f_4}{\partial y_i} = -2\frac{y_i + \delta_i}{y_N^2 + b} \tag{28}$$

and

$$\frac{\partial f_4}{\partial y_N} = 2y_N \left(\frac{\sum_{i=1}^{N-1} (y_i + \delta_i)^2}{(y_N^2 + b)^2} - 1 \right). \tag{29}$$

Using (9) we immediately obtain for $\varepsilon\sqrt{N-1} \le b$:

$$\forall i, \dots, N-1: \quad \left.\frac{\partial f(\mathbf{y}|\boldsymbol{\delta})}{\partial y_i}\right|_{\mathbf{y}=\mathbf{0}} = -2\frac{\delta_i}{b} \tag{30}$$

and

$$\left.\frac{\partial f(\mathbf{y}|\boldsymbol{\delta})}{\partial y_N}\right|_{\mathbf{y}=\mathbf{0}} = 0. \tag{31}$$

For the case $\varepsilon\sqrt{N-1} > b$ we find using $\hat{y}_N^2 + b = \varepsilon\sqrt{N-1}$, Eq. (9),

$$\varepsilon\sqrt{N-1} > b: \quad \forall i, \dots, N-1:$$

$$\left.\frac{\partial f(\mathbf{y}|\boldsymbol{\delta})}{\partial y_i}\right|_{\mathbf{y}=\hat{\mathbf{y}}} = -2\frac{\delta_i}{\varepsilon\sqrt{N-1}} \tag{32}$$

and

$$\varepsilon\sqrt{N-1} > b:$$

$$\left.\frac{\partial f(\mathbf{y}|\boldsymbol{\delta})}{\partial y_N}\right|_{\mathbf{y}=\hat{\mathbf{y}}} = \pm 2\sqrt{\varepsilon\sqrt{N-1} - b}\left(\frac{\sum_{i=1}^{N-1}\delta_i^2}{(N-1)\epsilon^2} - 1\right). \tag{33}$$

Let us consider the case $\varepsilon\sqrt{N-1} \leq b$, Eqs. (30, 31). While in (31) the derivative w.r.t. the Nth component is zero, the other derivatives in (30) are not equal to zero. Having a closer look at δ_i/b taking (7) into account, one sees that $\delta_i/b = \varepsilon/b \cdot \mathcal{N}_i(0,1)$. Therefore, the variance of a single component is $(\varepsilon/b)^2$. Taking the condition $\varepsilon\sqrt{N-1} \leq b$ into account, we can find an upper bound for the variance using $\varepsilon \leq b/\sqrt{N-1}$

$$\mathrm{Var}[\delta_i/b] = (\varepsilon/b)^2 \leq 1/(N-1) = \mathcal{O}(1/N). \tag{34}$$

That is, each of the $(N-1)$ derivatives asymptotically approach zero if the parameter space dimension $N \to \infty$.

As to the $\varepsilon\sqrt{N-1} > b$ case, we immediately conclude from (32) that the derivatives asymptotically vanish for the first $N-1$ components. While the expected value of the y_N-component (33) is zero, the variance of (33) needs further considerations. We calculate

$$\begin{aligned}
&\mathrm{Var}\left[\left.\tfrac{\partial f(\mathbf{y}|\boldsymbol{\delta})}{\partial y_N}\right|_{\mathbf{y}=\hat{\mathbf{y}}}\right] \\
&= 4(\varepsilon\sqrt{N-1} - b)\ \mathrm{Var}\left[\frac{\sum_{i=1}^{N-1}\delta_i^2}{(N-1)\epsilon^2} - 1\right] \\
&= 4(\varepsilon\sqrt{N-1} - b)\ \mathrm{Var}\left[\frac{\sum_{i=1}^{N-1}\mathcal{N}_i(0,1)^2}{(N-1)}\right] \\
&= 4\frac{\varepsilon\sqrt{N-1} - b}{(N-1)^2}\ \mathrm{Var}\left[\sum_{i=1}^{N-1}(\mathcal{N}_i(0,1))^2\right] \\
&= 4\frac{\varepsilon\sqrt{N-1} - b}{N-1}\ \mathrm{Var}\left[\mathcal{N}(0,1)^2\right] \\
&= 8\frac{\varepsilon\sqrt{N-1} - b}{N-1} = \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)
\end{aligned} \tag{35}$$

and see that also the N-th component vanishes asymptotically. In other words, the (global) robust optimizer $\hat{\mathbf{y}}$ of f_4 is an attractor for the steady state of the ES in the asymptotic limit case. This is in accordance with both the empirical observations, considering N-dimensionalities not too small (e.g., $N = 30$), and the steady state fitness error theory developed in [2].

D. Discussion

At first glance it appears as a surprise that the ES evolving on the two functions f_2, Eq. (2), and f_4, Eq. (6), exhibits qualitatively different behaviors. The functional differences of f_2 and f_4 seem rather small. Function f_4 has been introduced in [2] to allow for a theoretical steady state analysis of the behavior of the ES. The goal was to obtain a function f that allows for a decomposition of f into a deterministic part, being the conditional expectation of f and an asymptotically $(N \to \infty)$ *normally* distributed noise term. This was not possible for f_2. However, at that time this was regarded rather a purely technical problem than a qualitative difference. As we have seen now, normality is neither a necessary nor a sufficient condition for the correct or non-correct working of the ES as a robust optimizer strategy. Instead, the first derivatives of f at the robust optimizer state $\hat{\mathbf{y}}$ must vanish (at least) asymptotically

$$\left.\frac{\partial f(\mathbf{y}|\boldsymbol{\delta})}{\partial y_i}\right|_{\mathbf{y}=\hat{\mathbf{y}}} \to 0. \tag{36}$$

This motivates the quest for possible countermeasures if condition (36) is not fulfilled. Up to now, the only solution seems to be to fall back to resampling strategies. That is, one uses

$$\langle f\rangle_\kappa(\mathbf{y}) := \frac{1}{\kappa}\sum_{k=1}^{\kappa} f(\mathbf{y}|\boldsymbol{\delta}_k), \tag{37}$$

where the $\boldsymbol{\delta}_k$ are samples of the random variates $\boldsymbol{\delta}$. This is a trivial solution for $\kappa \to \infty$, since

$$\langle f\rangle_\kappa(\mathbf{y}) \to \mathrm{E}[f|\mathbf{y}] \tag{38}$$

may be regarded as the definition of the expected value. However, what is of interest here is the small κ case. Replacing $f(\mathbf{y})$ by $\langle f\rangle_\kappa(\mathbf{y})$ in (19) and (36) leads finally to

$$\frac{1}{\kappa}\sum_{k=1}^{\kappa}\left.\frac{\partial f(\mathbf{y}|\boldsymbol{\delta}_k)}{\partial y_i}\right|_{\mathbf{y}=\hat{\mathbf{y}}} \to 0. \tag{39}$$

If we apply this condition to critical derivatives of f_2 in Eq. (24), we obtain

$$\varepsilon \leq b: \quad \frac{1}{\kappa}\sum_{k=1}^{\kappa}\left.\frac{\partial f(\mathbf{y}|\boldsymbol{\delta}_k)}{\partial y_{N-1}}\right|_{\mathbf{y}=\mathbf{0}} = -\frac{2}{\kappa}\sum_{k=1}^{\kappa}\frac{\delta_k}{b} \tag{40}$$

resulting in a variance reduction by a factor of $1/\kappa$. The same holds for the case $\varepsilon > b$, Eq. (26) and (27), e.g., the variance of (27) becomes

$$\varepsilon > b: \quad \mathrm{Var}\left[\frac{1}{\kappa}\sum_{k=1}^{\kappa}\left.\frac{\partial f(\mathbf{y}|\boldsymbol{\delta}_k)}{\partial y_N}\right|_{\mathbf{y}=\hat{\mathbf{y}}}\right] = \frac{8}{\kappa}(\varepsilon - b) \tag{41}$$

and the steady state of the ES can approximate the global robust optimizer arbitrarily exact by increasing κ. While this option can be applied in any case, the question remains how the solution quality scales with κ. The answer to this question, however, depends on the function to be optimized. It cannot be answered by the linear analysis presented in this section.

V. Resampling

In Fig. 4, it has been shown that the expected steady state y_N can be controlled to approach the global optimizer to a certain extend by choosing the right truncation ratio ϑ. This is clearly an *ad hoc* solution. On the other hand, we have shown in Section IV-D that an ES with resampling (37) yields the global optimizer when the number of samples $\kappa \to \infty$. However, the question arises how the resampling performs for $\kappa < \infty$, especially for small κ, e.g. for $\kappa = 10$. Figure 6 compares the ES (truncation ratio $\vartheta = 0.4$) without and with resampling ($\kappa = 10$) on the level of function evaluations. First, we observe that the evolution strategy with resampling approaches the global optimizer state $\hat{\mathbf{y}}_N$ very well. Thus, our predictions from Section IV-D are confirmed: the algorithm benefits from moderate resampling sizes ($\kappa = 10$). Second, however, we clearly see the immense reduction in efficiency (note the logarithmic scale) when using resampling.

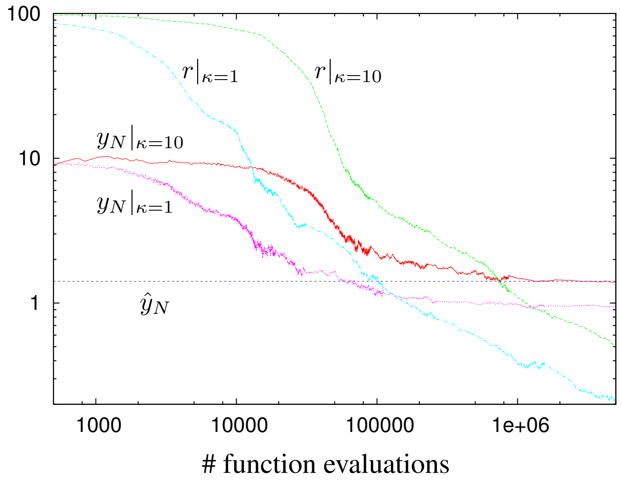

Fig. 6. Comparison of the dynamics of ES ($\vartheta = 0.4$) without ($\kappa = 1$) and with resampling ($\kappa = 10$) on f_2 with $b = 1$, $\varepsilon = 3$, and $N = 100$. Note the logarithmic scale on the horizontal axis: As to the r-dynamics, the efficiency of the ES with resampling is reduced by a factor of almost 10. Reaching the steady state y_N^{ss} takes also longer for the resampling strategy by a considerable factor.

In order to keep the resampling size as small as possible, the κ scaling behavior should be determined. However, this goes beyond the scope of this paper.

VI. Conclusion

In this paper, we have presented empirical findings on the convergence behavior of evolution strategies and genetic algorithms on two functions that belong to the class of FNIMs (functions with noise induced multi-modality). Furthermore, we have presented some theoretical analysis of the functions in the vicinity of the global optimizer state and were able to explain why the evolution strategy did not converge to $\hat{\mathbf{y}}_N$ for function f_2. Although, quantitatively this analysis cannot be applied to the binary-coded GA with fitness-proportional selection, it might – qualitatively – also indicate why the GA exhibits similar problems. The failure of the GA is not restricted to the classical canonical GA, it turned out that this also holds for real-coded GAs (accompanied by severe stability problems when using fitness-proportional selection).

One way out of this dilemma is to introduce explicit sampling as a strategy. This is particularly noticeable because up to now were was strong evidence that larger population sizes should be chosen instead of explicit sampling of the fitness function both in the domain of $(\mu/\mu_I, \lambda)$-ESs [16], [2] as well as genetic algorithms [17], [10]. Indeed, from Fig. 6, we clearly observe that explicit sampling is inefficient, however, at the same time, it seems to be – up to now – the only (secure) means to cope with functions like f_2 which exhibit noise-induced instabilities close to the global optimizer state. To answer the question posed in the title of this paper, we have to conclude ”it depends on the function”. However, generally the optimization problem is not given in analytical form and it will be hard – if not impossible – to determine the stability of the unknown global optimizer. At the same time, we can observe from Fig. 6, that it might be sufficient to resample only later in the search process. Thus, similarly to the adaptive population size algorithm suggested in [9] for evolution strategies, it might be sensible to derive adaptive sampling algorithms. However, it remains unclear which criterion they should employ in order to decide whether and when to increase or decrease the sample size. After all, it must be the aim to keep the sample size small.

References

[1] B. Sendhoff, H.-G. Beyer, and M. Olhofer, “On Noise Induced Multi-Modality in Evolutionary Algorithms,” in *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning – SEAL*, L. Wang, K. Tan, T. Furuhashi, J.-H. Kim, and F. Sattar, Eds., vol. 1, 2002, pp. 219–224.

[2] 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*, 2006.

[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. New York: World Scientific, 2004, pp. 152–172.

[4] T. Sonoda, Y. Yamaguchi, T. Arima, M. Olhofer, B. Sendhoff, and H.-A. Schreiber, “Advanced high turning compressor airfoils for low Reynolds number condition, Part 1: Design and optimization,” *Journal of Turbomachinery*, vol. 126, pp. 350–359, 2004.

[5] 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, 7*, K. De Jong, R. Poli, and J. Rowe, Eds. San Francisco, CA: Morgan Kaufmann, 2003, pp. 307–328.

[6] H.-G. Beyer and D. Arnold, “Qualms Regarding the Optimality of Cumulative Path Length Control in CSA/CMA-Evolution Strategies,” *Evolutionary Computation*, vol. 11, no. 1, pp. 19–28, 2003.

[7] N. Hansen, A. Ostermeier, and A. Gawelczyk, “On the Adaptation of Arbitrary Normal Mutation Distributions in Evolution Strategies: The Generating Set Adaptation,” in *Proc. 6th Int’l Conf. on Genetic Algorithms*, L. J. Eshelman, Ed. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1995, pp. 57–64.

[8] N. Hansen and 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.

[9] H.-G. Beyer and B. Sendhoff, "Evolution strategies for robust optimization," in *IEEE World Congress on Computational Intelligence*. IEEE Press, 2006.

[10] S. Tsutsui and A. Gosh, "Genetic algorithms with a robust solution searching scheme," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 3, pp. 201–208, 1997.

[11] 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 Systems, Man and Cybernetics Conference – SMC'99*, vol. 3. IEEE Press, 1999, pp. 585–591.

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

[13] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval schemata," in *Foundations of Genetic Algorithms, 2*, L. D. Whitley, Ed. San Mateo, CA: Morgan Kaufmann, 1993, pp. 187–202.

[14] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, pp. 115–148, 1995.

[15] H.-M. Voigt, H. Mühlenbein, and D. Cvetković, "Fuzzy recombination for the Breeder Genetic Algorithm," in *Proc. 6th Int'l Conf. on Genetic Algorithms*, L. J. Eshelman, Ed. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1995, pp. 104–111.

[16] H.-G. Beyer, "Actuator Noise in Recombinant Evolution Strategies on General Quadratic Fitness Models," in *GECCO-2004: Proceedings of the Genetic and Evolutionary Computation Conference*, K. Deb et al., Ed., vol. LNCS Volume 3102. Heidelberg: Springer-Verlag, 2004, pp. 654–665.

[17] 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.