Noisy Optimization: A Theoretical Strategy Comparison of ES, EGS, SPSA & IF on the Noisy Sphere

S. Finck
Vorarlberg University of Applied Sciences
Hochschulstrasse 1
Dornbirn, Austria
steffen.finck@fhv.at

Hans-Georg Beyer
Vorarlberg University of Applied Sciences
Hochschulstrasse 1
Dornbirn, Austria
hans-georg.beyer@fhv.at

Alexander Melkozerov
Department of Computer Science
University of Ulm
Ulm, Germany
alexander.melkozerov@uni-ulm.de

ABSTRACT

This paper presents a performance comparison of 4 direct search strategies in continuous search spaces using the noisy sphere as test function. While the results of the Evolution Strategy (ES), Evolutionary Gradient Search (EGS), Simultaneous Perturbation Stochastic Approximation (SPSA) considered are already known from literature, Implicit Filtering (IF) as the fourth strategy is firstly analyzed in this paper. After a short review of ES, EGS, and SPSA, the derivation of the quality gain formula of IF is sketched. Using the results, a comparison of the strategies is performed that worked out the similarities and differences of the strategies.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*parameter learning*; G.1.6 [**Numerical Analysis**]: Optimization—*performance measures*

General Terms

Algorithms, Theory, Experimentation, Performance

Keywords

Evolution Strategies, Evolutionary Gradient Search, Simultaneous Perturbation Stochastic Approximation, Implicit Filtering, performance analysis and comparison, progress rate, quality gain

1. INTRODUCTION

There exists a wealth of different strategies for optimizing large and complex models (e.g., traffic scheduling, biochemical processes, portfolio optimization, etc.). Typically, in such models the gradient is not available or computationally too expensive. Thus, (deterministic) optimization strategies which rely on exact gradient information can not be used, instead so-called direct search methods [12][1] are the methods of choice. The common behavior of these strategies is that the search through the domain is guided by the information obtained from sending "intelligent" queries to the model. No information on the underlying structure of the model is necessary. Furthermore, these strategies can also be applied in cases when there are uncertainties in the model's objective function (noisy optimization) and the outcome of the objective function is a random variable. When treated by standard deterministic optimization strategies, the behavior of such strategies might exhibit undesirable behaviors, e.g. premature convergence, divergence, or oscillating behavior.

There exist different categories of direct search strategies, ranging from derivative approximation to stochastic strategies and nature-inspired strategies. Practitioners then face the problem to select the best-performing strategy for their problem of interest. Hence, there is a need for information about the performance of the optimization strategies. One approach to evaluate the performance of strategies relies on benchmarking experiments, e.g. the BBOB framework [10] for continuous optimization. On the other hand, investigations on the theoretical level provide further information not (necessarily) available from experimental comparisons.

There are different theoretical approaches to the performance evaluation problem. One approach considers function classes which satisfy certain assumptions on the function properties, e.g. differentiability. The results obtained in such manner represent bounds on the runtime which are (often) expressed in terms of order notation. Thus, they contain hidden constants and it is hard to obtain statements about the influence of the strategy and model parameters on the performance of the strategy. The approach presented in this paper is more concerned with this influence. By determining the functional dependency between the parameters and the performance, one can obtain (approximate) optimal choices for the strategy-specific parameters which then could be used by practitioners. However, such information can only be obtained if the function class considered is expressed in analytical form.

The paper is organized as follows: In Section 2 the steps and assumptions of the framework for the theoretical analysis will be presented. The fitness environment for the com-

[1]Sometimes these strategies are also referred to as zeroth-order methods.

GECCO 2011 Dublin, Ireland

parison will be detailed in Section 3. In Section 4 the strategies considered will be introduced and in Section 5 compared with each other. Finally, conclusions and an outlook for future research will be given in Section 6.

2. THEORETICAL APPROACH

The theoretical analysis approach considered in this paper is based on methods from the analysis of dynamical systems. To illustrate the applicability of the dynamical systems analysis, one can imagine the optimization problem as physical environment. In this environment the optimization process represents a physical process which is governed by the forces acting and the equations of motion. Both can be expressed with help of the strategy and model parameters.

For all strategies, the first step is to derive an expression which represents a (local, expected) progress measure for a single iteration step. Common progress measures are the change in the fitness space, *quality gain*, or the change in the search space, *progress rate*. In most cases, the derivation of the performance measure is a demanding task. Often it is somewhat easier to aim at asymptotic expressions for infinite search space dimensionality, i.e., $N \to \infty$. Such expressions are usually much simpler and allow for a better interpretation of the results w.r.t. the influence of strategy-specific parameters. Thus, one is rather interested in simplified (due to the asymptotic) expressions from which a meaning can be inferred than in exact formulations which can not be easily interpreted.

The progress measure obtained represents the functional dependency between the strategy-specific parameters and the performance of the strategy. It can be used as starting point for obtaining convergence criteria and optimal strategy-specific parameters. The progress measure itself is a difference equation which in most cases depends on the current location within the search space. By introducing normalizations one is able to obtain values which attain a steady state (or a steady state distribution) after some iterations. These values can then be used to obtain the long-term dynamic behavior by solving the corresponding differential equations. From the long-term behavior one can derive statements about the runtime of the strategy.

Another advantage of the dynamical systems approach is that one can easily handle noisy optimization problems. From the physical viewpoint, it represents an additional force acting on the system. For most strategies and optimization problems considered so far, the noisy case is the generalization of the noise-free case. For noisy optimization problems one is typically interested in the minimal achievable distance to the optimizer, *residual location error*, or how the performance scales with the noise intensity, depending on the noise model considered.

In the following the fitness environment and the strategies considered will be described. The description will also detail the most important technical aspects used in the dynamical systems analysis.

3. FITNESS ENVIRONMENT

In this paper the noisy sphere model

$$f(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{x} + \sigma_\epsilon \mathcal{N}(0,1) \tag{1}$$

is considered. In Eq. (1) $f : \mathbb{R}^{N\times 1} \mapsto \mathbb{R}$, $\sigma_\epsilon \in \mathbb{R}_+$ is the so-called noise strength, and $\mathcal{N}(0,1) \in \mathbb{R}$ is a standard normally distributed random variate. There exists no correlation between successive evaluations of Eq. (1). In continuous optimization, the sphere can be seen as a landscape in the vicinity of a (local) optimizer of many optimization problems. It is amenable to rigorous mathematical analysis and the results will provide valuable insight into the behavior of the strategy considered. While there is absolutely no guarantee for generalization of the results to other optimization problems, the results can be reused in the analysis of more complex quadratic functions, see e.g. [7].

Concerning the noise, two different models are considered. In the first model, *constant noise variance*, σ_ϵ does not depend on the current location. Thus, the noise is more pronounced close to the optimizer and its influence can be neglected far away from the optimizer (if $\mathbf{x}^{\mathrm{T}}\mathbf{x} \gg \sigma_\epsilon \mathcal{N}(0,1)$). The second model considered, *constant normalized noise variance*, reduces σ_ϵ with decreasing distance to the optimizer. It is defined by

$$\sigma_\epsilon^* := \sigma_\epsilon \frac{N}{2R^2}, \tag{2}$$

where $R := \|\mathbf{x}\|$ is the distance to the optimizer and σ_ϵ^* is the normalized noise strength. In this model the optimizer is noise-free and σ_ϵ increases quadratically with the distance from the optimizer. Note, for this model the following simplification is made: The value of σ_ϵ is assumed to be constant within a single generation[2] as long as the points evaluated by the strategy are close to the current solution. This allows to simplify the math involved.

4. STRATEGIES

In this paper four different strategies are compared with each other: Evolution Strategy (ES) [13], Evolutionary Gradient Search (EGS) [5], Simultaneous Perturbation Stochastic Approximation (SPSA) [15], and Implicit Filtering (IF) [9]. The latter three employ different forms of gradient approximation, while ESs do not use gradient approximation in any way. All strategies are considered in their basic form and certain assumptions about the choice of the strategy-specific parameters are made. Therefore, the results will reflect (to a certain extent) optimal performance of the strategies. In the following the strategies are described and the equations used in the comparison are stated.

4.1 Evolution Strategies

Evolution Strategies (ESs) are loosely based on the concept of Darwinian evolution. Starting with an initial candidate solution $\mathbf{x}^{(g)} \in \mathbb{R}^{N\times 1}$, with g being the generation counter and N the search space dimensionality, a mutation operator is applied to procreate the offspring population of size λ. The mutation operator is an additive operator which is expressed as $\sigma\mathcal{N}(\mathbf{0},\mathbf{I})$ where σ is the mutation strength and $\mathcal{N}(\mathbf{0},\mathbf{I}) \in \mathbb{R}^{N\times 1}$ is a vector consisting of iid standard normally distributed components. Each offspring $\mathbf{y}_l \in \mathbb{R}^{N\times 1}, l \in \{1,\ldots,\lambda\}$ is evaluated, hence (at least) λ function evaluations are performed in each generation. The next step is the selection of the μ best offspring. The selection criterion is the fitness value $f(\mathbf{y}_l)$. The ranking of the offspring (in the case of minimization) is given by following

[2]Throughout the text the terms *iteration* and *generation* are used interchangeably.

notation

$$f_{1;\lambda} \leq f_{2;\lambda} \leq \ldots \leq f_{\mu;\lambda} \leq f_{\mu+1;\lambda} \leq \ldots f_{\lambda;\lambda}.$$

The order notation $i;j$ characterizes the i*th*-best individual out of a population of j individuals. Note, all results will also hold for maximization due to the maximization problem being equal to the minimization problem with inverted sign. The μ best offspring are recombined to obtain the new parental point. In this paper the intermediate recombination scheme

$$\mathbf{x}^{(g+1)} = \frac{1}{\mu} \sum_{m=1}^{\mu} \mathbf{y}_{m;\lambda}^{(g)}$$

is considered. It determines the centroid of the selected offspring. The technical notation for such a strategy is $(\mu/\mu_I, \lambda)$-ES.

Applying the dynamical systems approach, one must express the stochastic mapping

$$\left\{\mathbf{x}^{(g)}, \mathbf{s}^{(g)}\right\} \mapsto \left\{\mathbf{x}^{(g+1)}, \mathbf{s}^{(g+1)}\right\},$$

where $\mathbf{s}$ contains the strategy-specific parameters, e.g., the mutation strength. The mapping itself is a homogenous Markov process. An approximation of this process can be derived from the Chapman-Kolmogorov equations yielding the progress measure as an expected value. One assumption made is $N \to \infty$ which allows to neglect $N^{-\alpha}$-terms for $\alpha > 2$ in the derivation process (within Taylor series expansions). Another effect of this assumption is that it suffices to characterize the mapping by its expectation since the variance is proportional to $1/N$.

Considering the progress rate $\varphi = \mathrm{E}\left[R^{(g)} - R^{(g+1)}\right]$ one obtains [6]

$$\varphi^{*\,N\to\infty}{=} c_{\mu/\mu,\lambda} \frac{\sigma^*}{\sqrt{1+\vartheta^2}} - \frac{{\sigma^*}^2}{2\mu} \tag{3}$$

for the normalized progress rate $\varphi^* = \varphi\frac{N}{R}$. In Eq. (3) $c_{\mu/\mu,\lambda}$ is the generalized progress coefficient [6], $\sigma^* = \sigma\frac{N}{R}$ is the normalized mutation strength, and $\vartheta = \frac{\sigma^*_\epsilon}{\sigma^*}$ is the noise-to-signal ratio. For $N \to \infty$ the normalized progress rate and the normalized quality gain q^* are identical. The quality gain q and its normalization q^* are defined as

$$q := \mathrm{E}\left[f(\mathbf{x}^{(g)}) - f(\mathbf{x}^{(g+1)})\right] \quad \text{and} \quad q^* := q\frac{N}{2f\left(\mathbf{x}^{(g)}\right)}. \tag{4}$$

In the analysis it is assumed that σ^* remains constant throughout the evolutionary process. Therefore, no effects of mutation strength adaptation schemes will be considered.

4.2 Evolutionary Gradient Search

EGS is a hybrid strategy combining principles from evolutionary search and gradient approximation. Starting with an initial solution, 2λ offspring are created and evaluated. In [3] it was shown that using symmetrical points yields a more robust performance (especially in noisy optimization) than the original version from [14]. Using the same mutation operator as for ES, the offspring are procreated by

$$\begin{aligned}\mathbf{z}_l &= \mathcal{N}_l(\mathbf{0}, \mathbf{I})\\ \mathbf{y}_l &= \mathbf{x} + \sigma\mathbf{z}_l\\ \mathbf{y}_{l+\lambda} &= \mathbf{x} - \sigma\mathbf{z}_l\end{aligned}$$

for $l \in \{1, \ldots, \lambda\}$. The same notation as for ES holds. Instead of selection and recombination the gradient is approximated next. The approximation steps are

$$\begin{aligned}\mathbf{z}^{\mathrm{avg}} &= \sum_{l=1}^{\lambda} \left[f\left(\mathbf{y}_{l+\lambda}\right) - f\left(\mathbf{y}_l\right)\right] \mathbf{z}_l\\ \mathbf{z}^{\mathrm{prog}} &= \frac{\sqrt{N}}{\kappa} \frac{\mathbf{z}^{\mathrm{avg}}}{\left\|\mathbf{z}^{\mathrm{avg}}\right\|},\end{aligned}$$

where κ is a rescaling factor. The update of the solution is

$$\mathbf{x}^{(g+1)} = \mathbf{x}^{(g)} + \sigma\mathbf{z}^{\mathrm{prog}}.$$

One can see that the gradient approximation is used only for the direction of the update, while the step length equals $\sigma\frac{\sqrt{N}}{\kappa}$.

Due to the similarity to ES, the steps in the dynamical systems analysis are equal and the same assumptions are used. Again, the derived progress measure is an expected value. One obtains [3]

$$q^{*\,N\to\infty}{=} \frac{1}{\kappa}\left[\sigma^*\sqrt{\frac{\lambda}{1+\frac{\vartheta^2}{2}}} - \frac{{\sigma^*}^2}{2\kappa}\right]. \tag{5}$$

The same notations and normalization as before were used. As for ESs the normalized mutation strength σ^* is assumed to be constant for the analysis.

4.3 Simultaneous Perturbation Stochastic Approximation

The Simultaneous Perturbation Stochastic Approximation (SPSA) [15] is a stochastic gradient approximation strategy. Using a perturbation vector $\mathbf{\Delta}$, the gradient is approximated by means of a central difference scheme. The iid components of $\mathbf{\Delta}$ are (commonly) chosen from the ± 1 symmetric Bernoulli distribution. The advantage of this approximation scheme is that one needs only 2 function evaluations per iteration step to approximate the gradient independently of the search space dimensionality N.

Assuming SPSA is at state $\mathbf{x}^{(g)}$, the points $\mathbf{x}^{(g)} \pm c^{(g)}\mathbf{\Delta}^{(g)}$ are evaluated. Then the gradient approximation

$$\mathbf{g}^{(g)} = \frac{f\left(\mathbf{x}^{(g)} + c^{(g)}\mathbf{\Delta}^{(g)}\right) - f\left(\mathbf{x}^{(g)} - c^{(g)}\mathbf{\Delta}^{(g)}\right)}{2c^{(g)}}\mathbf{\Delta}^{(g)}$$

is performed. The term $c^{(g)}$ is the current step size factor for the approximation step. For noisy optimization it might be advantageous to use the average of several gradient samples. For such a scenario $c^{(g)}$ remains constant for all gradient samples within a single iteration step, however, $\mathbf{\Delta}^{(g)}$ is drawn anew for each sample. Afterwards, the update of the solution is performed by

$$\mathbf{x}^{(g+1)} = \mathbf{x}^{(g)} - a^{(g)}\mathbf{g}^{(g)},$$

where $a^{(g)}$ is the step size factor for the update step.

The choice of the step size factors (strongly) influence the performance of SPSA. For noise-free quadratic functions the

choice for $c^{(g)}$ is only relevant for numerical issues due to the use of the central gradient approximation scheme. For noisy functions, the choice of $c^{(g)}$ will influence the resulting noise level due to the appearing ratio $\sigma_\epsilon/c^{(g)}$. The step size factor $a^{(g)}$ is commonly chosen as

$$a^{(g)} = a^{(0)}(g + A)^{-\alpha}, \tag{6}$$

where $a^{(0)} \in \mathbb{R}$ is the initial step size factor (chosen by the user), $A \in \mathbb{R}$ is the stability factor, and $0 \leq \alpha \leq 1$ is the reduction rate of the step size factor sequence. The optimal settings of these parameters depend strongly on the optimization problem. For the analysis on the sphere model $A = 0$ is assumed. For $a^{(g)}$ the optimal values determined in [8] are used which do not contain the parameter α.

In [8] the normalized quality gain q^* for SPSA was determined. The analysis was based on the determination of the expectation of the gradient and subsequent decomposition of the gradient into a part pointing towards the optimizer and a perpendicular part. The assumption $N \to \infty$ is almost exclusively used to handle the noise terms (which are Gaussian distributed). For the sphere model one obtains

$$q^* \overset{N\to\infty}{=} 2a^{(g)}N\left(1 - \frac{a^{(g)}}{W}(N + W - 1)\right) - \frac{{a^{(g)}}^2 R^2 {\sigma_\epsilon^*}^2}{W{c^{(g)}}^2}, \tag{7}$$

where W is the number of gradient approximations per iteration.

4.4 Implicit Filtering

Implicit filtering (IF) [16, 11] is a deterministic direct search algorithm which approximates the gradient of the fitness function using either forward or central difference formula. In this analysis we will consider the central difference gradient approximation

$$\frac{\partial}{\partial x_i} f(\mathbf{x}) \approx \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h}, \tag{8}$$

where h is the difference increment and $\mathbf{e}_i$ are the unit vectors building the identity matrix $(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N) = \mathbf{I}$. The IF algorithm initializes the iteration counter $g = 0$, the initial search point $\mathbf{x}^{(0)}$, creates a decreasing sequence of difference increments $\{h_n\}$, selects the first $h = h_0$, and calls a steepest descent algorithm passing $\mathbf{x}^{(g)}$ and h to it.

The steepest descent algorithm uses Armijo's rule [1] for line search: it calculates the step length $d = \beta^m$, where $\beta \in (0, 1)$ is the line search parameter and $m = 0, 1, \ldots, m_{max}$ is the step length index, and samples a new point

$$\mathbf{x}^{(g+1)} = \mathbf{x}^{(g)} - d\nabla_h f(\mathbf{x}^{(g)}) \tag{9}$$

with $\nabla_h f(\mathbf{x}^{(g)})$ obtained with Eq. (8). After that, the sufficient decrease condition (in terms of Armijo's rule)

$$f(\mathbf{x}^{(g+1)}) - f(\mathbf{x}^{(g)}) < -\alpha d \left\| \nabla_h f(\mathbf{x}^{(g)}) \right\|^2 \tag{10}$$

is checked, where $\alpha \in (0, 1)$. If (10) is satisfied, then $\mathbf{x}^{(g+1)}$ is accepted, the index g is incremented, and the steepest descent algorithm is executed again. At most, this procedure may be repeated k_{max} times in the case when the sampled $\mathbf{x}^{(g+1)}$ is accepted each time. Afterwards the next h of the sequence $\{h_n\}$ will be chosen.

If the condition (10) is false, the step length d is reduced by incrementing the step length index m, once again a point is sampled and (10) is checked. The steepest descent algorithm reduces d m_{max} times maximum. If no $\mathbf{x}^{(g+1)}$ is accepted after m_{max} step length reductions, the line search fails and the steepest descent algorithm cancels its operation. The high-level IF algorithm switches to the next, smaller h in the sequence $\{h_n\}$ and invokes the steepest descent algorithm with this new h value. Since the difference increment h gets smaller during the run of the IF algorithm, difference formula (8) provides increasingly better gradient approximations which improve the quality of newly sampled $\mathbf{x}^{(g+1)}$ in (9).

We begin with the analysis of the steepest descent step (9). Due to space restrictions the derivations can only be sketched[3]. For the derivations it is assumed that α is sufficiently small such that (10) is fulfilled. Choosing the fitness gain (4) as progress measure, one obtains from (9) and (1)

$$\begin{aligned} q^{(g)} &= \mathrm{E}\left[\|\mathbf{x}^{(g)}\|^2 - \|\mathbf{x}^{(g)} - d\nabla_h f(\mathbf{x}^{(g)})\|^2\right] \\ &= \mathrm{E}\left[2d\mathbf{x}^{(g)\mathrm{T}}\nabla_h f(\mathbf{x}^{(g)}) - d^2\|\nabla_h f(\mathbf{x}^{(g)})\|^2\right] \\ &= 2d\mathbf{x}^{(g)\mathrm{T}}\mathrm{E}\left[\nabla_h f(\mathbf{x}^{(g)})\right] - d^2\mathrm{E}\left[\|\nabla_h f(\mathbf{x}^{(g)})\|^2\right]. \end{aligned} \tag{11}$$

In (11), the gradient $\nabla_h f$ is needed. Using (8) in conjunction with the fitness model (1), one obtains for the ith component of $\nabla_h f$

$$\begin{aligned} \left(\nabla_h f(\mathbf{x}^{(g)})\right)_i &= \frac{\|\mathbf{x} + h\mathbf{e}_i\|^2 + \sigma_\epsilon\mathcal{N}_1 - \|\mathbf{x} - h\mathbf{e}_i\|^2 - \sigma_\epsilon\mathcal{N}_2}{2h} \\ &= 2\left(\mathbf{x}^{(g)}\right)_i + \frac{1}{\sqrt{2}h}\sigma_\epsilon\mathcal{N}_i. \end{aligned} \tag{12}$$

Here we have taken into account that the difference of two iid normal variates is $\mathcal{N}(0, 2)$ distributed. That is, $\nabla_h f(\mathbf{x}^{(g)}) = 2\mathbf{x}^{(g)} + \frac{\sigma_\epsilon}{\sqrt{2}h}\mathcal{N}(\mathbf{0}, \mathbf{I})$. Inserting this result in (11), the expected values must be calculated. As for the first term one immediately obtains from (12)

$$\mathrm{E}\left[\nabla_h f(\mathbf{x}^{(g)})\right] = 2\mathbf{x}^{(g)}, \tag{13}$$

while the second expectation term in (11) yields after some calculations

$$\mathrm{E}\left[\|\nabla_h f(\mathbf{x}^{(g)})\|^2\right] = 4\|\mathbf{x}^{(g)}\|^2 + \frac{\sigma_\epsilon^2 N}{2h^2}. \tag{14}$$

Using Eqs. (13) and (14) in Eq. (11) leads to

$$\begin{aligned} q^{(g)} &= 4d\|\mathbf{x}^{(g)}\|^2 - 4d^2\|\mathbf{x}^{(g)}\|^2 - \frac{d^2\sigma_\epsilon^2 N}{2h^2} \\ &= 4d(1 - d)R^2 - \frac{d^2 N\sigma_\epsilon^2}{2h^2}, \end{aligned} \tag{15}$$

where $R = \|\mathbf{x}^{(g)}\|$ is the distance to the optimizer. Further we apply the quality gain (4) and noise strength normalizations (2) to get

$$q^* = 2Nd(1 - d) - {\sigma_\epsilon^*}^2\frac{d^2}{h^2}R^2. \tag{16}$$

One sees that the quality gain consists of two parts, a positive gain part (note $0 < d \leq 1$) and a negative loss part. The latter is only associated with noise. In the case of vanishing noise, IF is without any loss. Actually, choosing d

[3]The full derivation will be part of a forthcoming publication.

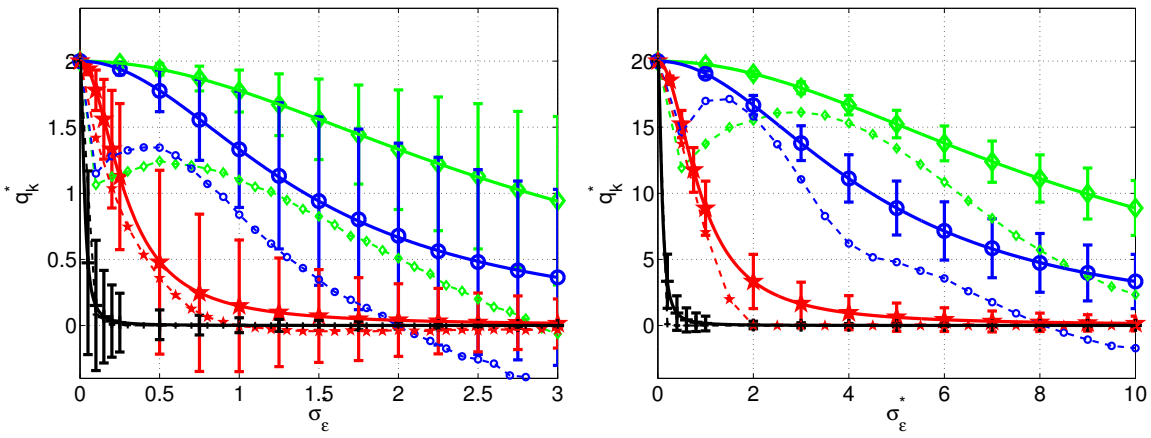

Figure 1: IF experiments on the noisy sphere for $N = 4$ (left) and $N = 40$ (right). Solid lines correspond to theoretical predictions of Eq. (16), while points with error bars represent the experimental results averaged over 10^4 runs for the following difference increment values: $+$ $h/R = 10^{-2}$, $\star$ $h/R = 0.1$, $\circ$ $h/R = 0.5$ and $\diamond$ $h/R = 1$. Dashed curves with points depict the experimental runs of the original IF algorithm.

optimally to 1/2, the optimum is reached in one generation. The noisy case is more interesting and it makes sense to compare it with SPSA (7). This reveals remarkable similarities: The first term does not depend on the location in the search space, whereas the second term depends on $f(\mathbf{x}) = R^2$. Furthermore, the first term does only depend on the step size factor $a^{(g)}$ and d, respectively. The second term associated with the noise has again the same structure. The step size factor a and d, respectively, appear squared in the numerator. The increment step lengths c and h appear squared in the denominator. In both IF and SPSA the loss term depends on the squared normalized noise strength and the current function value $f(\mathbf{x}) = R^2$. Convergence (in expectation) is obtained if $q^* > 0$ is ensured. Using (16) this implies $(\sigma_\epsilon^* R)^2 < 2Nh^2(1-d)/d$. Noting that, according to Eq. (2) $\sigma_\epsilon^* = \text{const.}$ describes the fitness proportional noise case, in that case convergence to the optimizer is ensured for sufficiently small distances R to the optimizer (given fixed d and h). Conversely, when initializing IF too far away from the optimizer, the strategy can diverge.

The normalized quality gain formula (16) allows for determination of the optimal step length d_{opt} which maximizes q^*. Calculating the derivative of q^* and equating the result to zero yields

$$d_{opt} = \frac{1}{2} \frac{1}{1 + \frac{1}{8}\left(\frac{\sigma_\epsilon N}{hR}\right)^2}. \tag{17}$$

Moreover, inserting (17) into (16) results in the equation for the maximal quality gain

$$q^*_{max} = \frac{N}{2} \frac{1}{1 + \frac{1}{8}\left(\frac{\sigma_\epsilon N}{hR}\right)^2}, \tag{18}$$

from which we directly obtain that the IF is able to attain $q^*_{max} = N/2$ maximum in the absence of noise.

The next step in the analysis is to transfer the theoretical result (16) derived for the single steepest descent step to the complete IF algorithm. We take the following observation for the noise-free sphere model into account: Substituting the step length calculation formula $d = \beta^m$ in the IF algorithm with $d = d_{opt}$, where d_{opt} is given by Eq. (17), provides maximal possible quality gain in the steepest descent step (9). This means that the optimal step length is selected instantly without step length reductions, the sufficient decrease condition is therefore true and $\mathbf{x}^{(g+1)}$ is accepted in the first iteration of the steepest descent algorithm. Thus, setting $d = d_{opt}$ results in the best performance scenario of the IF algorithm when the first sampled $\mathbf{x}^{(g+1)}$ has improved fitness associated with it, and the need to spend additional fitness function evaluations for line search is eliminated. The complete IF algorithm transforms into a simplified IF without the line search subroutine and the condition (10).

The situation is more complicated for the noisy sphere model. There is no guarantee that $\mathbf{x}^{(g+1)}$ will be accepted even if $d = d_{opt}$, but averaging over individual IF runs should still yield the results matching Eq. (16). We check this hypothesis in the following experiments. We run the simplified IF algorithm with step length reduction rule substituted with the optimal value $d = d_{opt}$ calculated using Eq. (17). Further we set $k_{max} = 1$ and $m_{max} = 0$. Using random initial search points, we measure q^* by using the non-noisy fitness function values at $\mathbf{x}^{(g)}$ and $\mathbf{x}^{(g+1)}$ for a given value of h. The experimental results averaged over 10^4 runs are presented in Fig. 1. For comparison, we also run the same experiments for the original IF algorithm (Fig. 1, dashed curves) with the original step length reduction rule $d = \beta^m$ and parameter settings $k_{max} = 1$, $m_{max} = 2$, $\alpha = 10^{-4}$, $\beta = 0.5$. Note that the original IF algorithm uses for these settings on average more fitness function evaluations (FEs) per run than the simplified IF, which uses $\#\text{FEs} = 2N$ fitness function evaluations per iteration for the gradient approximation.

As expected, Eq. (16) predicts the exact normalized quality gain of the IF with optimal step length for the noise-free sphere. Moreover, Eq. (16) correctly predicts for $\sigma_\epsilon^* = 0$ q^* of the original IF algorithm which uses $2N + 3$ FEs since it requires $2N+2$ FEs for the steepest descent algorithm and 1 FE for one step length reduction in order to sample a point with improved $f(\mathbf{x})$.

Considering the $\sigma_\epsilon^* > 0$ data in Fig. 1, we conclude that our hypothesis about averaging over IF runs is true: the prediction quality of Eq. (16) is quite good for the simplified IF

Table 1: Maximal efficiency and corresponding strategy-specific parameter

strategy	optimal parameter	max. efficiency
$(\mu/\mu_I, \lambda)$-ES	$\mu/\lambda \approx 0.27$, $\sigma^* = \mu c_{\mu/\mu,\lambda}$	≈ 0.202 $(\mu \to \infty)$
EGS	$\sigma^* = \kappa\sqrt{\lambda}$	0.25
SPSA	$a = \frac{W}{2(N+W-1)}$, $W = 1$	0.25
IF	$d = \frac{1}{2}$	0.25

with $N = 4$ and $N = 40$. Next to the curves for the simplified IF, results for the original IF are given by the dashed curves in Fig. 1. While these curves do not quantitatively coincide with the theoretical predictions by (16), one can interpret the theoretical predictions as upper bound on the performance. Note, the original IF curves did not use d_{opt} (17).

5. COMPARISON

The comparison is based on the following four equations: Eq. (3) (ES), Eq. (5) (EGS), Eq. (7) (SPSA), and Eq. (16) (IF). Further, the results will only consider scale-invariant mutation strength adaptation ($\sigma^* = $ const.) for EGS and ES, and the step size factors in IF and SPSA connected with the update step will set to be optimal for the noise-free sphere. For IF we additionally assume $2N + 2 \approx 2N$.

5.1 Results for the Noise-free Sphere

First, the comparison of the four strategies is performed on the noise-free sphere model, i.e., $\sigma_\epsilon = 0$ in (1). While one can simply compare the normalized quality gain expressions, which all have theoretical maximum of $q^* = N/2$, such a comparison will neglect the "effort" of the strategy. A performance measure accounting for that is (serial) efficiency

$$\eta = \frac{q^*}{\#\text{FEs}}, \tag{19}$$

where #FEs is the number of function evaluations per iteration step. Note, η does not scale between 0 and 1 as for example efficiency of physical processes. The theoretical maximal efficiency is $\eta = N/2$ (by using q^* of above form) if the maximal progress, i.e., $f(\mathbf{x}^{(g+1)}) = f(\mathbf{x}_{\text{opt}}) = 0$ will be achieved for arbitrary $\mathbf{x}^{(g)}$ with 1 function evaluation. However, due to the strategies either attaining maximal normalized quality gain values lower than $N/2$ and/or using #FEs of some order of N, η is typically in the range $0 < \eta \leq 1/2$. The best efficiency on the sphere determined so far (within direct search strategies) is attained by the (λ_{opt})-ES with $\eta = 0.5$ for $\lambda \to \infty$ [4].

Inspecting the equations of interest, one can observe a common feature. The positive gain term is linear in the respective update step step size factor, namely σ (ES and EGS), $a^{(g)}$ (SPSA), and d (IF). On the other hand, the negative gain term is quadratic in the respective step size factor. That means, there exist an optimal step size factor for each strategy and an upper bound on the step size factor until which progress towards the optimizer can be achieved. Since the comparison will be based on these optimal values, the following results should be interpreted as upper bounds on the performance of the strategies.

In Table 1 the maximal efficiencies and the corresponding strategy-specific parameters are listed. The three strategies using some type of gradient approximation have the same maximal efficiency of $\eta = 0.25$ which is about 25% higher than the efficiency for the $(\mu/\mu_I, \lambda)$-ES. Comparing the optimal step size factors, one can observe a difference. For EGS and ES the normalized mutation strength is chosen $\sigma^* \propto$ #FEs, and in SPSA $a^{(g)} \propto$ #FEs$/N$. This yields for these three strategies a normalized quality gain of $q^* \propto$ #FEs. On the other hand, in IF the normalized quality gain is $q^* \propto N$ (16), while #FEs $\propto N$. This shows a difference in the basic ideas of the strategies. In IF one hopes to achieve a high quality gradient estimate which will be used for the update. In SPSA and EGS, a low quality gradient estimates suffices and one hopes that the resulting estimation errors will cancel out during the iteration process. The questions now is how these ideas work if one considers noisy optimization.

5.2 Results for the Noisy Sphere

For the comparison on the noisy sphere the setting $\sigma^*_\epsilon = $ const. will be considered first. If one looks at the equations of interest one can observe that the noise term is coupled with a (single) step size factor for each strategy. For each strategy holds that increasing this step size factor will reduce the noise level. For EGS and $(\mu/\mu_I, \lambda)$-ES this factor is the normalized mutation strength σ^*. For IF and SPSA, it is the step size factor used in the gradient approximation step, i.e. h (IF) or $c^{(g)}$ (SPSA). Therefore, the first analysis will look at the change in η by increasing the respective step size factor coupled with the noise term for $\sigma^*_\epsilon = 1$.

The respective curves are shown in the left-hand plot of Fig. 2. One can observe two different general behaviors. On the one side, for EGS and $(\mu/\mu_I, \lambda)$-ES the efficiency will be maximal for a certain σ^* and there exist an upper bound on the choice of σ^*. On the other hand, for IF and SPSA increasing h or $c^{(g)}$, respectively, increases the efficiency up to the noise-free value. Depending on the location, there exist a lower bound on these step size factors. These differences can be explained by realizing that in EGS and $(\mu/\mu_I, \lambda)$-ES σ^* is used for creating the test points *and* updating the solution. In IF and SPSA the respective step size factors for these two steps are uncoupled.

Another observation can be made from the left-hand plot in Fig. 2. Unlike ES and EGS, the normalized quality gains for IF and SPSA still depend on R and N for $\sigma^*_\epsilon > 0$.[4] The influence of both factors is about the same. Thus, the same conclusions can be drawn by analyzing either the case of keeping N constant and changing R or vice versa[5]. Therefore, two curves are shown for IF and SPSA in Fig. 2 representing two different values of R. Note, if $\mathbf{x}^{(g)} = (1, 1, 1, 1)^{\text{T}}$, the resulting distance to the optimizer will be $R = 2$. From the curves, one can conclude that by increasing R the minimal necessary value for h or $c^{(g)}$ (i.e. $\eta > 0$) will increase and

[4]This is partially an effect of the different types of step size factors (coupled with the update step) employed. In ESs and EGS, normalized values of the step size factor are considered, while in IF and SPSA these values are non-normalized.However, using the same normalization for the step size factors in IF and SPSA would still result in expressions depending on the current search point location.

[5]Note, *increasing* N for $R = $ const. will be the same as *reducing* R for $N = $ const. w.r.t. the conclusions drawn.

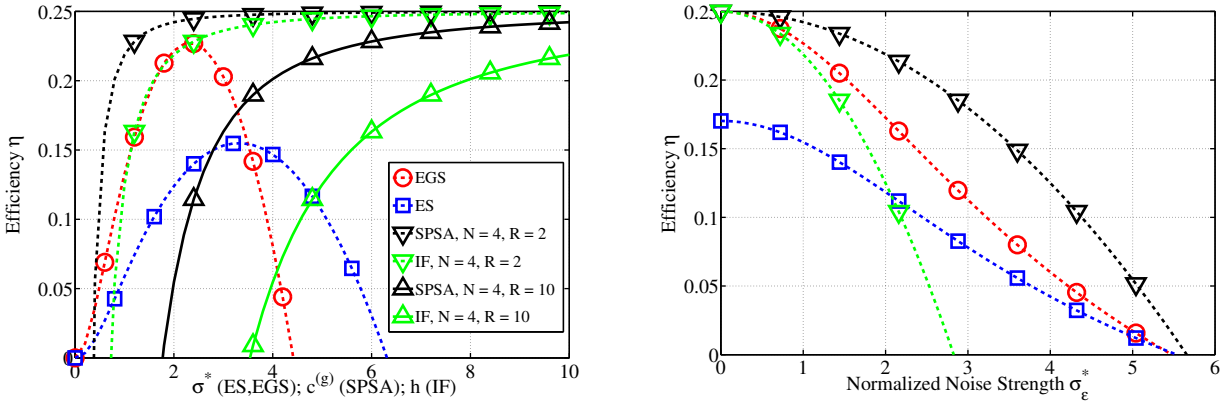

Figure 2: Efficiencies η on the noisy sphere with $\sigma_\epsilon^* = \text{const.}$ for the $(3/3_I, 10)$-ES, EGS with $\lambda = 5$, SPSA with $W = 1$ and $a = 1/(2N)$, and IF with $d = 1/2$. *Left:* The noise level is $\sigma_\epsilon^* = 1$ and the step size factor coupled with the noise is increased. *Right:* The noise level σ_ϵ^* is increased and the step size factors $c^{(g)} = R$ and $h = R$ are used, respectively. The IF curves with $R = 2$ and $R = 10$ are identical. The same holds for SPSA.

that the same value of h and $c^{(g)}$ (as used for the smaller R) will yield a reduced efficiency. Thus, one could argue that h (or $c^{(g)}$) should be chosen large such that even for large R $\eta > 0$ is achieved. However, this could result in numerical errors/reduced accuracy close to the optimizer. A better choice is to reduce the respective step size factor during the optimization process such that $\eta > 0$. This is also consonant with the completely implemented SPSA and IF algorithms where both $c^{(g)}$ and h decrease during the run. By investigating (7) and (16), one may choose $h = R$ and $c^{(g)} = R$, respectively, to achieve such a decreasing sequence. For larger values one will observe a better efficiency value, while for smaller values the noise term will be multiplied by a factor larger than 1.

The next scenario considered will look at the change in efficiency for increasing σ_ϵ^* while keeping the strategy-specific parameters constant. The values for σ^*, d, and $a^{(g)}$ are chosen to be optimal for $\sigma_\epsilon^* = 0$. The motivation for this setup is that the optimal noise-free performance will serve as baseline performance. The influence of the different strategy parameters will be discussed later. The corresponding curves for all strategies are shown in the right-hand plot of Fig. 2. The same markers and linestyles as before were used. With the chosen setup, i.e., $h = R$ and $d = 1/2$ for IF, the curves for $R = 2$ and $R = 10$ are identical. The same holds for SPSA. All strategies are only able to achieve progress up to a certain value of σ_ϵ^* for a given set of strategy parameters. Further, the efficiency decreases faster for IF than for the other three strategies considered. Note, for $\sigma_\epsilon^* = 0$ the $(\mu/\mu_I, \lambda)$-ES achieves the maximal efficiency for $\mu = 3$ and $\lambda = 10$. Due to the suboptimal population ratio, the maximal efficiency does not achieve $\eta = 0.2$.

The last scenario to be considered concerns the question how close each strategy can attain the optimizer if $\sigma_\epsilon = \text{const.}$ holds. This can be obtained by determining the maximal value σ_ϵ^* for which $\eta \geq 0$ is satisfied and then solving the corresponding equation for R. This stationary distance will be called residual location error and denoted as R_∞. The results are listed in Table 2. Concerning the influence of the noise strength, one can observe that σ_ϵ is linear in R_∞^2 for EGS and $(\mu/\mu_I, \lambda)$-ES, while it is quadratic for IF and SPSA. Note, in the case of SPSA $a^{(g)} \propto 1/N$ must hold, which ensures the root term to be positive.

The equations for R_∞ also reveal (to a certain extent) which strategy parameters can be used to handle noisy information. For the $(\mu/\mu_I, \lambda)$-ES, one can increase the number of offspring λ while holding the population ratio μ/λ constant. This in turn yields higher optimal values of the normalized mutation strength due to $\sigma^*_{\text{opt}} \propto \mu$ and a decrease in the noise-to-signal ratio ϑ. However, the increase is limited if one considers finite search space dimensionalities [2]. In EGS, one can increase the number of points λ, too. Similar effects as for the $(\mu/\mu_I, \lambda)$-ES will be observed. Additionally, one can increase κ which results in smaller R_∞ and for the noise model with $\sigma_\epsilon^* = \text{const.}$ the maximal efficiency value will shift to larger normalized noise strengths. The effect of κ is that the test points created will have a larger distance from the current solution, while the length of the update step remains unchanged. Again, there exist an upper bound on the choice of κ for finite N. In SPSA, in addition to $c^{(g)}$, one could increase the number of gradient samples or decrease $a^{(g)}$ to reduce R_∞. Increasing the number of gradient approximations, however, is only beneficial once SPSA is close to the R_∞, before that a significant decrease in the efficiency would be observed. Commonly, a pre-defined decreasing sequence (6) for $a^{(g)}$ will be used. However, as shown in [8] this might result in suboptimal performance. In IF, decreasing d improves the quality of the solution albeit at the cost of a reduced normalized quality gain (16).

6. CONCLUSION

This paper contributed to the performance comparison of four strategies on the noisy sphere. The results provide insight on "good" parameter settings and allow to *directly* compare the performances with each other. For the $(\mu/\mu_I, \lambda)$-ES, EGS, and SPSA we used available results, while for (a simplified version of) IF new theoretical results have been given and the derivation steps have been sketched.

The strategies considered differ in their basic approaches. Three of them (IF, SPSA, and EGS) rely on different kinds

Table 2: Residual location error R_∞

strategy	$(\mu/\mu_I, \lambda)$-ES	EGS	SPSA	IF
R_∞	$\sqrt{\frac{N\sigma_\epsilon}{4\mu c_{\mu/\mu,\lambda}}}$ $(\sigma^* \to 0)$	$\sqrt{\frac{N\sigma_\epsilon}{4\kappa\sqrt{\lambda}}}$ $(\sigma^* \to 0)$	$\frac{\sigma_\epsilon}{2c^{(g)}}\sqrt{\frac{N}{2(W(1/a^{(g)}-1)-N+1)}}$	$\frac{\sigma_\epsilon}{2h}\sqrt{\frac{dN}{2(1-d)}}$

of gradient approximations to improve the current solution. From these, EGS and SPSA use low quality gradient estimate which is computationally cheap (low number function evaluations necessary). Due to the iterative process, the estimation errors cancel out over time. In IF a high quality gradient approximation is used which is expensive, however, less iterations than in SPSA and EGS should be necessary to achieve a good solution quality. Finally, ESs are considered which do not use gradient approximation in any form.

Considering the noise-free sphere, the strategies with gradient approximation achieved the same efficiency which was higher than the one achieved by ESs. For the noisy sphere, IF and SPSA can achieve a higher efficiency under the assumption that the respective step size factor associated with the noise is chosen correctly. For both strategies one can choose the respective factor proportional to the current distance to the optimizer, which on the sphere model equals the root of the function value of the current search point. Another difference is the order of the noisy influence. For EGS and ES the noise appears as linear term in the quality gain, while it is quadratic for the other two strategies.

For all strategies optimal step size factors can be determined depending on the noise level, the distance to the optimizer, and the resulting strategy parameters. For $\sigma_\epsilon > 0$ these values can usually not be determined without information typically not available to the strategy (e.g., distance to the optimizer or noise level). However, these values can serve as baseline for comparison with the actually employed step size factor adaptation schemes. For ES, EGS, and SPSA it was shown that the employed schemes only achieve a portion of the optimal performance. The corresponding analysis for the full IF version including Armijo's rule is still missing. An interesting question here is, if one can derive more rigorous statements about the performance of pre-defined and decoupled schemes for the step size factor(s) vs. schemes based on self-adaptation principles.

Finally, the question is how these results transfer to other test functions. This is an open question which can only be answered by performing the analysis on those functions. However, experience shows that in continuous optimization the results for the sphere can be (partially) reused for the analysis of more general quadratic functions, e.g. $f = \mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x}$. Further, the results presented here are a first step for the analysis of such functions.

7. ACKNOWLEDGMENTS

Support by the Austrian Science Fund (FWF) under grant P22649-N23 is gratefully acknowledged. Alexander Melkozerov would like to thank Dr. Hans A. Kestler and the German Academic Exchange Service (DAAD) for support of his work.

8. REFERENCES

[1] L. Armijo. Minimization of functions having Lipschitz-continuous first partial derivatives. *Pacific Journal of Mathematics*, 16:1–3, 1966.

[2] D. V. Arnold. *Local Performance of Evolution Strategies in the Presence of Noise*. Ph.D. Thesis, University of Dortmund, Dortmund, 2001.

[3] D. V. Arnold. An analysis of evolutionary gradient search. In *Proceedings of the CEC'04 Conference*, pages 47–54, Piscataway, NJ, 20-23 June 2004. IEEE.

[4] D. V. Arnold. Weighted Multirecombination Evolution Strategies. *Theoretical Computer Science*, 361(1):18–37, 2006.

[5] D. V. Arnold and R. Salomon. Evolutionary Gradient Search Revisited. *IEEE Trans. Evolutionary Computation*, 11(4):480–495, 2007.

[6] H.-G. Beyer. *The Theory of Evolution Strategies*. Natural Computing Series. Springer, Heidelberg, 2001.

[7] H.-G. Beyer and S. Finck. Performance of the $(\mu/\mu_I, \lambda)$-σSA-ES on a Class of PDQFs. *IEEE Transactions on Evolutionary Computation*, 14(3):400–418, 2010.

[8] S. Finck and H.-G. Beyer. Performance Analysis of Simultaneous Perturbation Stochastic Approximation on the Noisy Sphere Model. *Theoretical Computer Science*, 2010. submitted.

[9] P. Gilmore and C. Kelley. An Implicit Filtering Algorithm for Optimization of Functions with Many Local Minima. *SIAM Journal on Optimization*, 5:269–285, 1995.

[10] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Technical Report RR-7215, INRIA, 2010.

[11] C.T. Kelley. *Iterative Methods for Optimization*. SIAM, Philadelphia, 1999.

[12] R. M. Lewis, V. Torczon, and M. W. Trosset. Direct Search Methods: Then and Now. *Journal of Computational and Applied Mathematics*, 124(1–2):191–207, 2000.

[13] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Verlag, Stuttgart, 1973.

[14] R. Salomon. Evolutionary Search and Gradient Search: Similarities and Differences. *IEEE Transactions on Evolutionary Computation*, 2(2):45–55, 1998.

[15] J. C. Spall. Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, March 1992.

[16] T. Winslow, R. Trew, G. P., and C. Kelley. Simulated Performance Optimization of GaAs MESFET Amplifiers. In *Proceedings IEEE/Cornell Conference on Advanced Concepts in High Speed Devices and Circuits*, pages 393–402. IEEE, 1991.