Proceedings of the 28th German Conference on Artificial Intelligence,
September 11-14, 2005

Noise Robustness by Using Inverse Mutations

Ralf Salomon

Faculty of Computer Science and Electrical Engineering
University of Rostock, 18051 Rostock, Germany
Email: ralf.salomon@etechnik.uni-rostock.de

Abstract. Recent advances in the theory of evolutionary algorithms have indicated that a hybrid method known as the evolutionary-gradient-search procedure yields superior performance in comparison to contemporary evolution strategies. But the theoretical analysis also indicates a noticeable performance loss in the presence of noise (i.e., noisy fitness evaluations). This paper aims at understanding the reasons for this observable performance loss. It also proposes some modifications, called *inverse mutations*, to make the process of estimating the gradient direction more noise robust.

1 Introduction

The literature on evolutionary computation discusses the question whether or not evolutionary algorithms are gradient methods very controversially. Some [10] believe that by generating trial points, called offspring, evolutionary algorithms stochastically operate along the gradient, whereas others [4] believe that evolutionary algorithms might deviate from gradient methods too much because of their high explorative nature.

The goal of an optimization algorithm is to locate the optimum $\boldsymbol{x}^{(\text{opt})}$ (minimum or maximum in a particular application) of an N-dimensional objective (fitness) function $f(x_1, \ldots, x_N) = f(\boldsymbol{x})$ with x_i denoting the N independent variables. Particular points $\boldsymbol{x}$ or $\boldsymbol{y}$ are also known as search points. In essence, most procedures differ in how they derive future test points $\boldsymbol{x}^{(t+1)}$ from knowledge gained in the past. Some selected examples are gradient descent, Newton's method, and evolutionary algorithms. For an overview , the interested reader is referred to the pertinent literature [3, 6–9, 14].

Recent advances in the theory of evolutionary algorithms [1] have also considered hybrid algorithms, such as the evolutionary-gradient-search (EGS) procedure [12]. This algorithm fuses principles from both gradient and evolutionary algorithms in the following manner:

- it operates along an explicitly estimated gradient direction,
- it gains information about the gradient direction by randomly generating trial points (offspring), and
- it periodically reduces the entire population to a single search point.

A description of this algorithm is presented in Section 2, which also briefly reviews some recent theoretical analyses. Arnold [1] has shown that the EGS procedure performs superior, i.e., sequential efficiency, in comparison to contemporary evolution strategies.

But Arnold's analysis [1] also indicates that the performance of EGS progressively degrades in the presence of noise (noisy fitness evaluations). Due to the high importance of noise robustness in real-world applications, Section 3 briefly reviews the relevant literature and results.

Section 4 is analyzing the reason for the observable performance loss. A first result of this analysis is that the procedure might be extended by a second, independently working step size in order to de-couple the gradient estimation from performing the actual progress step.

A main reason for EGS suffering from a performance loss in the presence of noise is that large mutation steps cannot be used, because they lead to quite imprecise gradient estimates. Section 5 proposes the usage of inverse mutations, which use each mutation vector $\boldsymbol{z}$ twice, originally and in the inverse direction $-\boldsymbol{z}$. Section 5 shows that inverse mutations are able to compensate for noise due to a mechanism, which is called genetic repair in other algorithms. Section 6 concludes with a brief discussion and an outlook for future work.

2 Background: Algorithms and Convergence

This section summarizes some background material as far as necessary for the understanding of this paper. This includes a brief description of the procedure under consideration as well as some performance properties on quadratic functions.

2.1 Algorithms

As mentioned in the introduction, the evolutionary-gradient-search (EGS) procedure [12] is a hybrid method that fuses some principles of both gradient and evolutionary algorithms. It periodically collapses the entire population to a single search point, and explicitly estimates the gradient direction in which it tries to advance to the optimum. The procedure estimates the gradient direction by randomly generating trial points (offspring) and processing *all* trial points in calculating a weighted average. In its simplest form, EGS works as follows[1]:

1. Generate i=1, ..., λ offspring $\boldsymbol{y}_t^{(i)}$ (trial points)

$$\boldsymbol{y}_t^{(i)} = \boldsymbol{x}_t + \sigma \boldsymbol{z}_t^{(i)} \tag{1}$$

[1] Even though further extensions, such as using a momentum term or generating non-isotropic mutations, significantly accelerate EGS [13], they are not considered in this paper, because they are not appropriately covered by currently available theories. Furthermore, these extensions are not rotationally invariant, which limits their utility.

from the current point $\boldsymbol{x}_t$ (at time step t), with $\sigma>0$ denoting a step size and $\boldsymbol{z}^{(i)}$ denoting a mutation vector consisting of N independent, normally distributed components.

2. Estimate the gradient direction $\tilde{\boldsymbol{g}}_t$:

$$\tilde{\boldsymbol{g}}_t = \sum_{i=1}^{\lambda} \left(f(\boldsymbol{y}_t^{(i)}) - f(\boldsymbol{x}_t) \right) \left(\boldsymbol{y}_t^{(i)} - \boldsymbol{x}_t \right) \ . \qquad (2)$$

3. Perform a step

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \sqrt{N} \frac{\tilde{\boldsymbol{g}}_t}{\|\tilde{\boldsymbol{g}}_t\|} = \boldsymbol{x}_t + \boldsymbol{z}_t^{(\mathrm{prog})} \ , \qquad (3)$$

with $\boldsymbol{z}^{(\mathrm{prog})}$ denoting the progress vector.

It can be seen that Eq. (2) estimates an approximated gradient direction by calculating a weighted sum of *all* offspring regardless of their fitness values; the procedure simply assumes that for offspring with negative progress, positive progress can be achieved in the opposite direction and sets those weights to negative values. This usage of *all* offspring is in way contrast to most other evolutionary algorithms, which only use ranking information of some selected individuals of some sort. Further details can be found in the literature [1, 5, 12].

EGS also features some dynamic adaptation of the step size σ. This, however, is beyond the scope of this paper, and the interested reader is referred to the literature [12].

This paper compares EGS with the $(\mu/\mu, \lambda)$-evolution strategy [5, 10], since the latter yields very good performance and is mathematically well analyzed. The $(\mu/\mu, \lambda)$-evolution strategy maintains μ parents, applies global-intermediate recombination on all parents, and applies normally distributed random numbers to generates λ offspring, from which the μ best ones are selected as parents for the next generation. In addition, this strategy also features some step size adaptation mechanism [1].

2.2 Progress on Quadratic Functions

Most theoretical performance analyses have yet been done on the quadratic function $f(x_1, \ldots, x_n) = \sum_i x_i^2$, also known as the sphere model. This choice has the following two main reasons: First, other functions are currently too difficult to analyze, and second, the sphere model approximates the optimum's vicinity of many (real-world) applications reasonably well. Thus, the remainder of this paper also adopts this choice.

As a first performance measure, the rate of progress is defined as

$$\varphi = f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1}) \qquad (4)$$

in terms of the best population members' objective function values in two subsequent time steps t and $t{+}1$. For standard (μ, λ)-evolution strategies operating on the sphere model, Beyer [5] has derived the following rate of progress φ:

$$\varphi \approx 2Rc_{\mu,\lambda}\sigma - N\sigma^2 \ , \qquad (5)$$

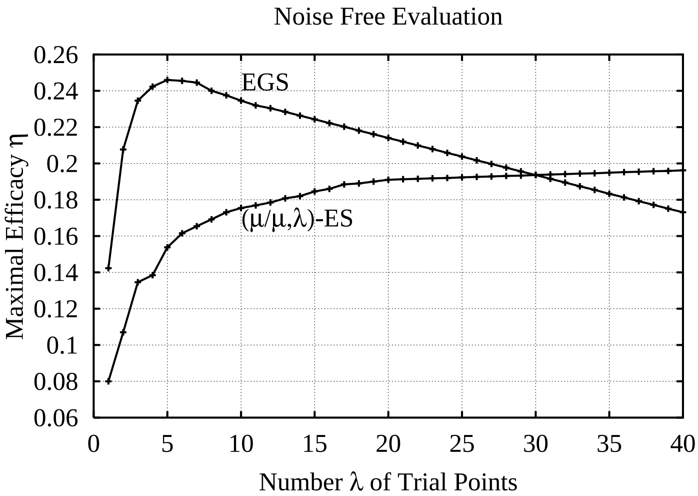

Fig. 1. Rate of progress for EGS and $(\mu/\mu, \lambda)$-evolution strategies according to Eqs. (7) and (8). Further details can be found in [1].

with $R = \|\boldsymbol{x}_g\|$ denoting the distance of the best population member to the optimum and $c_{\mu,\lambda}$ denoting a constant that subsumes all influences of the population configuration as well as the chosen selection scheme. Typical values are: $c_{1,6}$=1.27, $c_{1,10}$=1.54, $c_{1,100}$=2.51, and $c_{1,1000}$=3.24. To be independent from the current distance to the optimum, normalized quantities are normally considered (i.e., relative performance):

$$\varphi^* = \sigma^* c_{\mu,\lambda} - 0.5(\sigma^*)^2 \text{ with}$$
$$\varphi^* = \varphi \frac{N}{2R^2}, \sigma^* = \sigma \frac{N}{R} \quad . \tag{6}$$

Similarily, the literature [1, 5, 10] provides the following rate of progress formulas for EGS and the $(\mu/\mu, \lambda)$-evolution strategies:

$$\varphi^*_{EGS} \approx \sigma^* \sqrt{\frac{\lambda}{1 + {\sigma^*}^2/4}} - \frac{{\sigma^*}^2}{2}, \tag{7}$$

$$\varphi^*_{\mu/\mu,\lambda} \approx \sigma^* c_{\mu/\mu,\lambda} - \frac{\sigma^*}{2\mu} \tag{8}$$

The rate of progress formula is useful to gain insights about the influences of various parameters on the performance. However, it does not consider the (computational) costs required for the evaluation of all λ offspring. For the common assumption that all offspring be evaluated sequentially, the literature often uses a second performance measure, called the efficiency $\eta = \varphi^*/\lambda$. In other words, the efficiency η expresses the sequential run time of an algorithm.

Figure 1 shows the efficiency of both algorithms according to Eqs. (7) and (8). It can be seen that for small numbers of offspring (i.e., $\lambda{\approx}5$), EGS is most efficient (in terms of sequential run time) and superior to the $(\mu/\mu, \lambda)$-evolution strategy.

3 Problem Description: Noise and Performance

Subsection 2.2 has reviewed the obtainable progress rates for the undisturbed case. The situation changes, however, when considering noise, i.e., noisy fitness evaluations. Noise is present in many (if not all) real-world applications. For the viability of practially relevant optimization procedures, noise robustness is thus of high importance.

Most commonly, noise is modeled by additive $N(0,\sigma_\epsilon)$ normally distributed random numbers with standard deviation σ_ϵ. For noisy fitness evaluations, Arnold [1] has derived the following rate of progress

$$\varphi^*_{EGS} \approx \sigma^* \sqrt{\frac{\lambda}{1+{\sigma^*}^2/4+{\sigma^*_\epsilon}^2/{\sigma^*}^2}} - \frac{{\sigma^*}^2}{2} \;, \tag{9}$$

with $\sigma^*_\epsilon = \sigma_\epsilon N/(2R^2)$ denoting the normalized noise strength.

Figure 2 demonstrates how the presence of noise σ^*_ϵ reduces the rate of progress of the EGS procedure. Eq. (9) reveals that positive progress can be achieved only if $\sigma^*_\epsilon \leq \sqrt{4\lambda+1}$ holds. In other words, the required number of offspring (trial points) to estimate the gradient direction grows quadratically.

Another point to note is that the condition $\sigma^*_\epsilon \leq \sqrt{4\lambda+1}$ incorporates the *normalized* noise strength. Thus, if the procedure approaches the optimum, the

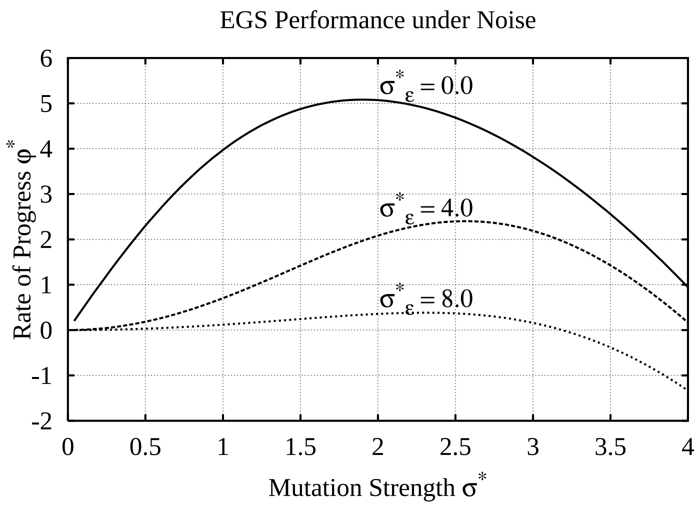

Fig. 2. The rate of progress φ^* of EGS progressively degrades under the presence of noise σ^*_ϵ. The example has used λ=25 offspring. Further details can be found in [1].

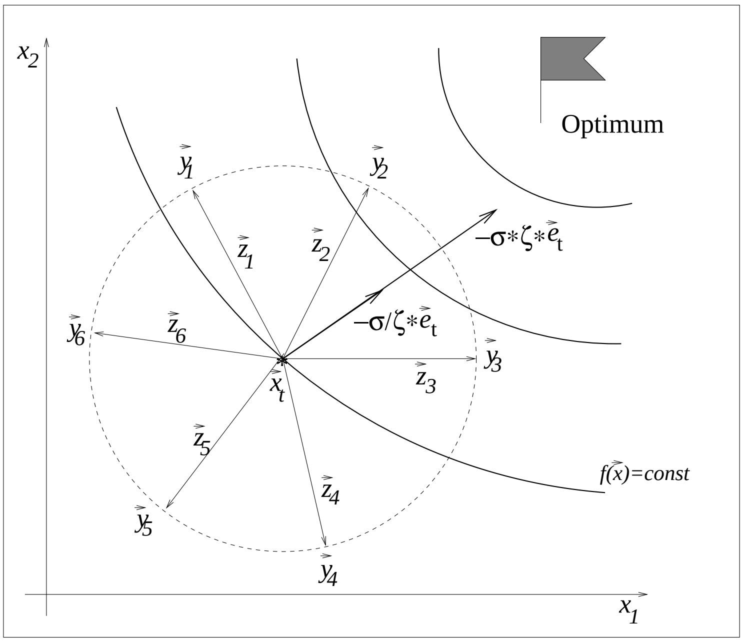

Fig. 3. This figure shows how the EGS procedure estimates the gradient direction $\tilde{\boldsymbol{g}}$ by averaging weighted test points $\boldsymbol{y}$.

distance R decreases and the noise strength increases. Consequently, the procedure exhibits an increasing performance loss as it advances towards the optimum.

By contrast, the $(\mu/\mu, \lambda)$-evolution strategy benefits from an effect called genetic repair induced by the global-intermediate recombination, and is thus able to operate with larger mutation step sizes σ^*. For the $(\mu/\mu, \lambda)$-evolution strategy, the literature [2] suggest that only a linear growth in the number of offspring λ is required.

4 Analysis: Imprecise Gradient Approximation

This section analysis the reasons for the performance loss described in Section 3. To this end, Figure 3 illustrates how the EGS procedure estimates the gradient direction according to Eq. (2): $\tilde{\boldsymbol{g}}_t = \sum_{i=1}^{\lambda}(f(\boldsymbol{y}_t^{(i)}) - f(\boldsymbol{x}_t))(\boldsymbol{y}_t^{(i)} - \boldsymbol{x}_t)$. The following two points might be mentioned here:

1. The EGS procedure uses the same step size σ for generating trial points and performing a step from $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t+1}$.
2. For small step sizes σ, the probability for an offspring to be better or worse than its parents is half chance. For larger step sizes, though, the chances for better offspring are steadily decreasing, and are approaching zero for $\sigma \geq 2R$.

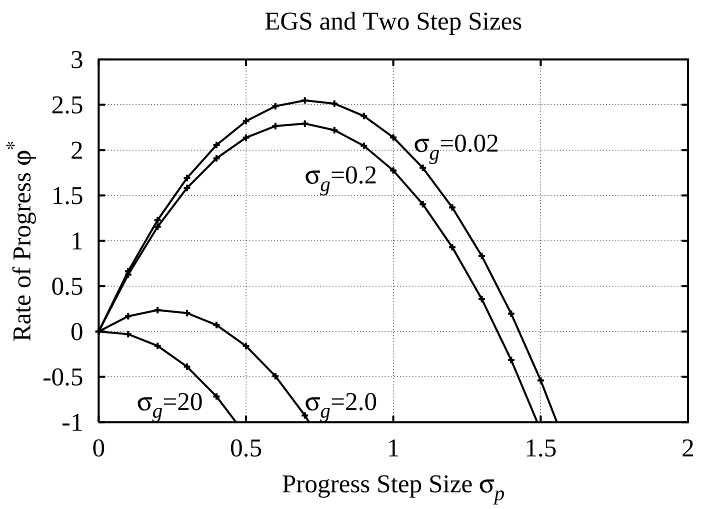

Fig. 4. Normalized rate of progress φ^* when using different step sizes σ_g and σ_p for the example $N = \lambda = 10$.

It is obvious that for small λ, unequal chances have a negative effect on the gradient approximation accuracy.

Both points can be further elaborated as follows: A modified version of the EGS procedure uses two independent step sizes σ_g and σ_p for generating test points (offspring) $\boldsymbol{y}_t^{(i)}{=}\boldsymbol{x}_t{+}\sigma_g \boldsymbol{z}_t^{(i)}$ and performing the actual progress step $\boldsymbol{x}_{t+1}{=}\boldsymbol{x}_t$ $+\sigma_p\sqrt{N}\tilde{\boldsymbol{g}}_t/\|\tilde{\boldsymbol{g}}_t\|$ according to Eqs. (1) and (3), respectively. Figure 4 illustrates the effect of these two step sizes for the example $R = 1$, $N = 10$ dimensions and $\lambda = 10$ trial points. It can be clearly seen that the rate of progress φ^* drastically degrades for large step sizes σ_g. Since Figure 4 plots the performance for 'all possible' step sizes σ_p, it can be directly concluded that the accuracy of the gradient estimation significantly degrades for step size σ_g being too large; if the estimation $\tilde{\boldsymbol{g}} = \boldsymbol{g}$ was precise, the attainable rate of progress would be $\varphi^* = (f(\boldsymbol{x}_0) - f(\boldsymbol{0}))N/(2R^2) = (1-0)10/2 = 5$.

This hypothesis is supported by Fig. 5, which shows the angle $\cos\alpha = \boldsymbol{g}\tilde{\boldsymbol{g}}/(\|\boldsymbol{g}\|\|\tilde{\boldsymbol{g}}\|)$ between the true gradient $\boldsymbol{g}$ and its estimate $\tilde{\boldsymbol{g}}$. It can be clearly seen that despite being dependent on the number of trial points λ, the gradient estimate's accuracy significantly depends on the step size σ_g. The qualitative behavior is more or less equivalent for all three number of trial points $\lambda \in \{5, 10, 25\}$: it is quite good for small step sizes, starts degrading at $\sigma_g \approx R$, and quickly approaches zero for $\sigma_g > 2R$. In addition, Fig. 6 shows how the situation changes when noise is present. It can be seen that below the noise level, the accuracy of the gradient estimate degrades.

In summary, this section has show that it is generally advantageous to employ two step sizes σ_g and σ_p and that the performance of the EGS procedure degrades

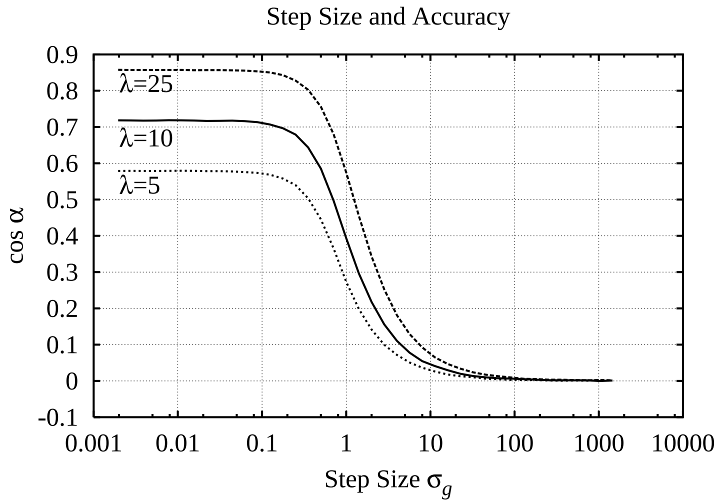

Fig. 5. $\cos\alpha$ between the true gradient $\boldsymbol{g}$ and its estimate $\tilde{\boldsymbol{g}}$ as a function of the number of trial points λ.

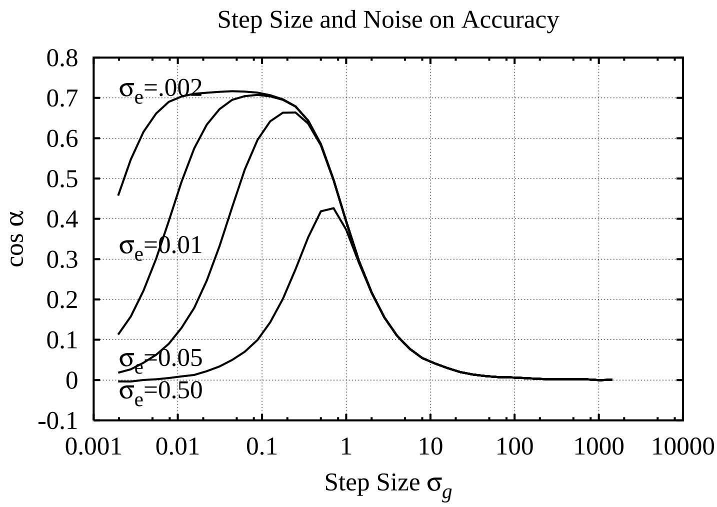

Fig. 6. $\cos\alpha$ between the true gradient $\boldsymbol{g}$ and its estimate $\tilde{\boldsymbol{g}}$ as a function of the noise strength $\sigma_e \in \{0.002, 0.01, 0.05, 0.5\}$. In this example, $N = \lambda = 10$ and $R = 1$ were used.

when the step size σ_g is either below the noise strength or above the distance to the optimum.

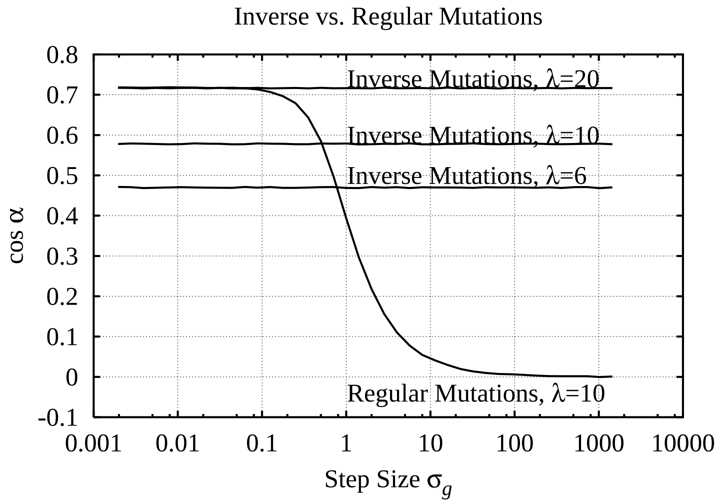

Fig. 7. $\cos\alpha$ between the true gradient $\boldsymbol{g}$ and its estimate $\tilde{\boldsymbol{g}}$ for various numbers of trial points λ. For comparison purposes, also the regular case with $\lambda = 10$ is shown. In this example, $N = 10$ and $R = 1$ were used.

5 Inverse Mutations

The previous section has identified two regimes, $\sigma_g < \sigma_e$ and $\sigma_g > R$ that yield poor gradient estimates. This problem could be tackled by either resampling already generated trial points or by increasing the number of different trial points. Both options, however, have high computational costs, since the performance gain would grow at most with $\sqrt{\lambda}$ (i.e., reduced standard deviation or Eq. (9)). Since the performance gain of the $(\mu/\mu, \lambda)$-evolution strategy grows linearly in λ (due to genetic repair [1]), these options are not further considered here.

For the second problematic regime $\sigma_g > R$, this paper proposes *inverse mutations*. Here, the procedure still generates λ trial points (offspring). However, half of them are mirrored with respect to the parent, i.e., they are pointing towards the opposite direction. Inverse mutations can be formally defined as:

$$\begin{aligned} \boldsymbol{y}_t^{(i)} &= \boldsymbol{x}_t + \sigma_g \boldsymbol{z}_t^{(i)} \text{ for } i = 1 \ldots \lceil \lambda/2 \rceil \\ \boldsymbol{y}_t^{(i)} &= \boldsymbol{x}_t - \sigma_g \boldsymbol{z}_t^{(i - \lceil \lambda/2 \rceil)} \text{ for } i = \lceil \lambda/2 \rceil + 1, \ldots \lambda \end{aligned} \tag{10}$$

In other words, each mutation vector $\boldsymbol{z}^{(i)}$ is used twice, once in its original form and once as $-\boldsymbol{z}^{(i)}$.

Figure 7 illustrates the effect of introducing inverse mutations. The performance gain is obvious: The observable accuracy of the gradient estimate is constant over the entire range of σ_g values. This is in sharp contrast to the regular case, which exhibits the performance loss discussed above. The figure, however, indicates a slight disadvantage in that the number of trial points should be twice

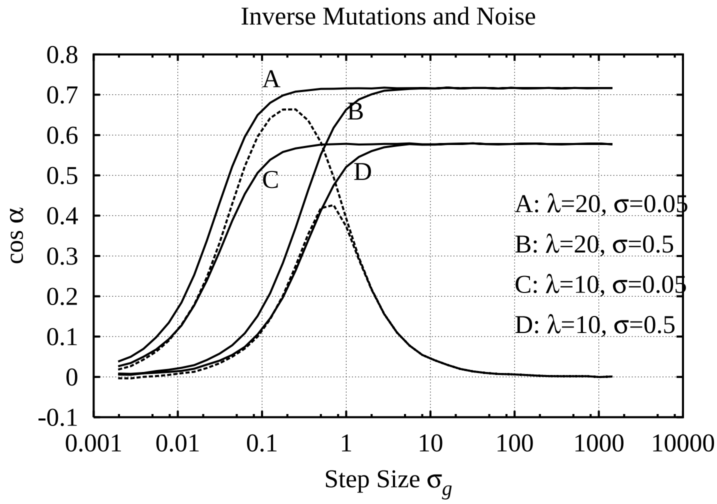

Fig. 8. $\cos\alpha$ between the true gradient $\boldsymbol{g}$ and its estimate $\tilde{\boldsymbol{g}}$ for $\lambda \in \{10, 20\}$ and $\sigma_e \in \{0.05, 0.5\}$. For comparison purposes, also the regular case with $\lambda = 10$ and $\sigma_e \in \{0.05, 0.5\}$ (dashed lines). In this example, $N = 10$ and $R = 1$ were used.

as much in order to gain the same performance as in the regular case. In addition, the figure also shows the accuracy for $\lambda = 6$, which is smaller than the number of search space dimensions; the accuracy is with $\cos\alpha \approx 0.47$ still reasonably good.

Figure 8 illustrates the performance inverse mutations yield in the presence of noise. For comparison purposes, the figure also shows the regular case for $\lambda = 10$ and $\sigma_e \in \{0.05, 0.5\}$ (dashed lines). Again, the performance gain is obvious: the accuracy degrades for $\sigma_g < \sigma_e$ but sustains for $\sigma_g > R$, which is in way contrast to the regular case.

Figure 9 shows the rate of progress φ^* of the EGS procedure with two step sizes σ_g and σ_p and inverse mutations for the example $N = 40$ and λ=24 and a normalized noise strength of $\sigma^*_\epsilon = 8$. When comparing the figure with Fig. 2, it can be seen that the rate of progress is almost that of the undisturbed case, i.e., $\sigma^*_\epsilon = 0$. It can be furthermore seen that the performance starts degrading only for too small a step size σ_g. It should be mentioned here that for the case of $\sigma^*_\epsilon = 0$, the graphs are virtually identical with the cases $\sigma^*_\epsilon = 8$ and $\sigma_g = 32$, and are thus not shown.

Figure 10 demonstrates that the proposed method is not restricted to the sphere model; also the general quadratic case $f(\boldsymbol{x}) = \sum_i \xi^i x_i^2$ benefits from using inverse mutations. The performance gain is quite obvious when comparing graph "C", i.e., regular mutations, with graphs "A" and "B", i.e., inverse mutations. When using regular mutations, the angle between the true and estiamted gradient is rather eratic, whereas inverse mutations yield virtually the same accuracy as compared to the sphere model. It might be worhtwhile to note

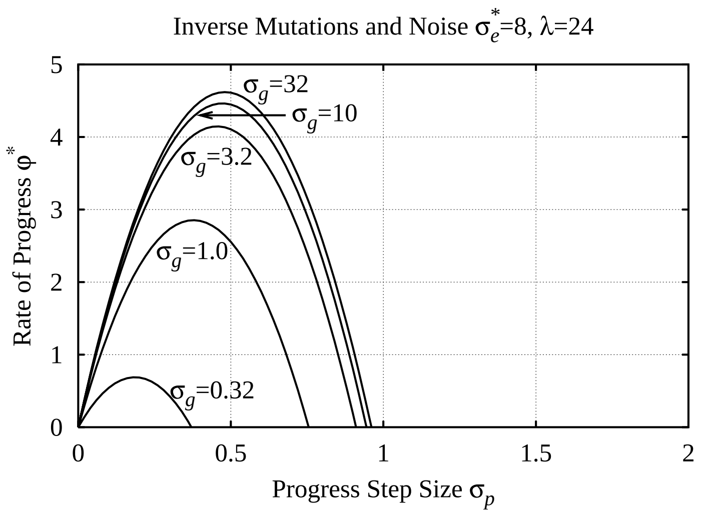

Fig. 9. The rate of progress φ^* of EGS with two step sizes and inverse mutations under the presence of noise $\sigma^*_\epsilon = 8$. The example has used $N = 40$ and λ=24. In comparison with Fig. 2, the performance is not affected by the noise.

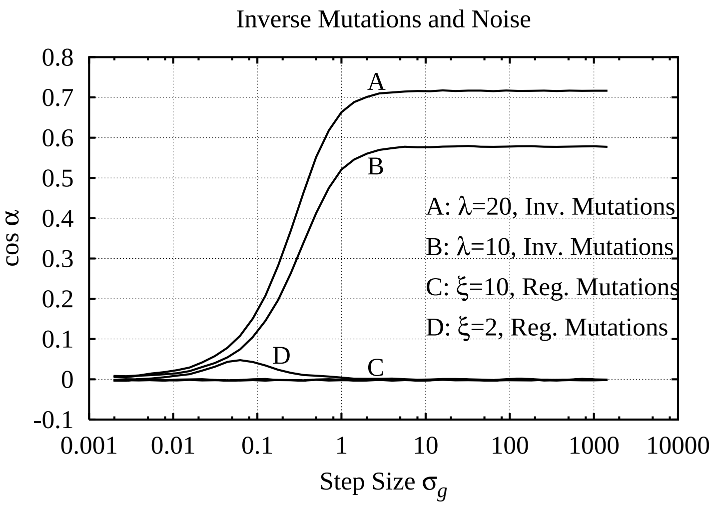

Fig. 10. $\cos\alpha$ between the true gradient $\boldsymbol{g}$ and its estimate $\tilde{\boldsymbol{g}}$ for $\lambda \in \{10, 20\}$ and $\sigma_e = 0.5$ for the test function $f(\boldsymbol{x}) = \sum_i \xi^i x_i^2$ when using inverse mutations (labels "A" and "B") as well as regular mutations (labels "C" and "D"). For graphs "A"-"C", $\xi = 10$, and for graph "D", $\xi = 2$ was used. In all examples, $N = 10$ and $R = 1$ were used.

that regular mutations are able to yield some reasonable accurace only for small excentricities, e.g., $\xi = 2$ in graph "D".

Even though inverse mutations are able to accurately estimate the gradient *direction*, that do not resort to second order derivatives – that in the case of excentric quadratic fitness functions, the gradient does not point towards the minimum. Please note that as all gradient-based optimization procedures that do not utilize second order derivatives suffer from the same problem.

6 Conclusions

This paper has briefly reviewed the contemporary $(\mu/\mu, \lambda)$-evolution strategy as well as a hybrid one known as the evolutionary-gradient-search procedure. The review of recent analyses has emphasized that EGS yields superior efficiency (sequential run time), but that it significantly degrades in the presence of noise. The paper has also analyzed the reasons of this observable performance loss. It has then proposed two modifications: inverse mutations and using two independent step sizes σ_g and σ_p, which are used to generate the λ trial points and to perform the actual progress step, respectively. With these two modifications, the EGS procedure yields a performance, i.e., the normalized rate of progress φ^*, which is very noise robust; the performance is almost not effected as long as the step size σ_g for generating trial points is sufficiently large.

The achievements reported in this paper have been motivated by both analyses and experimental evidence, and have be validated by further experiments. Future work will be devoted to conducting a mathematical analysis.

References

1. D. Arnold, An Analysis of Evolutionary Gradient Search, *Proceedings of the 2004 Congress on Evolutionary Computation.* IEEE, 47-54, 2004.
2. D. Arnold and H.-G. Beyer, Local Performance of the $(\mu/\mu, \lambda)$-Evolution Strategy in a Noisy Environment, in W. N. Martin and W. M. Spears, eds., *Proceeding of Foundation of Genetic Algorithms 6 (FOGA),* Morgan Kaufmann, San Francisco, 127-141, 2001.
3. T. Bäck, U. Hammel, and H.-P. Schwefel, Evolutionary Computation: Comments on the History and Current State. *IEEE Transactions on Evolutionary Computation,* 1(1), 1997, 3-17.
4. H.-G. Beyer. On the Explorative Power of ES/EP-like Algorithms, in V.W. Porto, N. Saravanan, D. Waagen, and A.E. Eiben, (eds), *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming (EP'98),* Springer-Verlag, Berlin, Heidelberg, Germany, 323-334, 1998.
5. H.-G. Beyer. *The Theory of Evolution Strategies.* Springer-Verlag Berlin, 2001.
6. D.B. Fogel. *Evolutionary Computation: Toward a New Philosophy of Machine Learning Intelligence..* IEEE Press, NJ, 1995.
7. D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, MA, 1989.
8. D.G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, Menlo Park, CA, 1984.

9. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes.* Cambridge University Press, 1987.
10. I. Rechenberg, *Evolutionsstrategie* (Frommann-Holzboog, Stuttgart, 1994).
11. R. Salomon and L. Lichtensteiger, Exploring different Coding Schemes for the Evolution of an Artificial Insect Eye. *Proceedings of The First IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks,* 2000, 10-16.
12. R. Salomon, Evolutionary Algorithms and Gradient Search: Similarities and Differences, *IEEE Transactionson Evolutionary Computation,* **2**(2): 45-55, 1998.
13. R. Salomon, Accelerating the Evolutionary-Gradient-Search Procedure: Individual Step Sizes, in T. Bäck, A. E. Eiben, M. Schoenauer, and H.-P. Schwefel, (eds.), *Parallel Problem Solving from Nature (PPSN V)*, Springer-Verlag, Berlin, Heidelberg, Germany, 408-417, 1998.
14. H.-P. Schwefel. *Evolution and Optimum Seeking.* John Wiley and Sons, NY. 1995.