A Potential Potential Function for a Variable-Metric Evolution Strategy

Stephan Frank[0009−0008−0112−6023] and
Tobias Glasmachers[0000−0003−1886−1696]

Fakultät für Informatik, Institut für Neuroinformatik, Ruhr-Universität Bochum, Bochum, Germany
{stephan.frank,tobias.glasmachers}@ini.rub.de

Abstract. This paper works towards an analysis of a variable-metric evolution strategy by means of drift analysis. Drift analysis has been effective for proving convergence and analyzing the runtime of a simple (1+1)-ES. We make a first step towards including covariance matrix adaptation (CMA). To this end, we develop a novel class of potential functions for the (1+1)-CMA-ES optimizing two-dimensional convex quadratic functions. We leverage invariances to efficiently sample a representative space of states. We use simulations to gain an empirical estimate of the expected minimal drift induced by the candidate potential function and to tune potential function parameters. Our results indicate that the tuned potential function is negative and uniformly bounded away from zero, which yields linear convergence.

1 Introduction

Variable metric evolution strategies like the covariance matrix adaptation evolution strategy (CMA-ES) [8,11,9] are among the best performing methods for difficult black-box optimization problems [7,3]. However, due to their randomized nature and the lack of convergence guarantees, they are sometimes considered unreliable heuristics. The lack of analytical convergence guarantees can be a barrier to the adoption of state-of-the-art methods like CMA-ES. Even though the developers of such algorithms have a good understanding of how the methods will perform on a problem, it can be difficult to convey this understanding to practitioners. We therefore believe that developing theoretical performance guarantees is an important line of research. On that route, we pursue theory-guided empirical analysis as an intermediate goal.

Runtime analysis of evolutionary algorithms is a well-established field [16]. It is very well developed for optimization in discrete domains, where nearly all recent results were established by means of drift analysis techniques [12,13]. The desire to understand the optimization behavior of evolution strategies is not new [4]. In recent years, there was significant progress in transferring drift techniques to the analysis of continuous optimization [10,5,1,14,2,15]. We witnessed an impressive generalization in terms of problems, starting from the simple sphere function and arriving at large function classes like all strongly convex functions with Lipschitz gradient. However, in terms of algorithms, only rather simplistic evolution strategies without covariance matrix adaptation (CMA) were analyzed. While being quite flexible in principle, the apparent challenge of the drift-based approach is to identify a suitable potential function. The present paper is concerned with the question of how to design a suitable potential for a variable metric ES.

In the present paper, we aim to make progress towards analyzing variable-metric evolution strategies by means of drift. We believe that analyzing convergence through empirical means and the help of drift analysis can be valuable in addition to the traditional method of using analytical proofs. The natural first step in the analysis is the quest for a Lyapunov potential capturing the quite involved algorithm dynamics sufficiently well. We propose to address the problem of designing a suitable potential function with an iterative method based on empirical analysis. Our approach can lead to a better understanding of potential functions and guide the development of analytical proofs, by allowing for piecewise advancements on potential functions, supported by empirical performance data. The approach can be adapted rather easily to a wide range of algorithms and objective functions, expanding the range of problems for which runtime and convergence guarantees can be established.

Algorithms Our general methodology is not bound to a specific algorithm. It can hence be applied, e.g., to a fully fledged state-of-the-art implementation of CMA-ES. Instead, we use a simplified version of the (1+1)-CMA-ES [9] as outlined in Algorithm 1. This is a natural choice if we wish to leverage existing results, since the literature on analyzing evolution strategies with drift is focused on elitist selection algorithms.

Algorithm 1: Simplified variant of (1+1)-CMA-ES

```
Input: $d \in \mathbb{N}$, $f : \mathbb{R}^d \to \mathbb{R}$, $m \in \mathbb{R}^d$, $\sigma > 0$, $c_{\text{cov}} \in (0,1]$
$p_{\text{target}} = \frac{2}{11}$
while stopping condition not met do
    $z \sim \mathcal{N}(0, C)$
    $x \leftarrow m + \sigma \cdot z$
    if $f(x) \leq f(m)$ then
        $m \leftarrow x$ ; $p_{\text{succ}} \leftarrow 1$ ; $C \leftarrow (1 - c_{\text{cov}}) \cdot C + c_{\text{cov}} \cdot zz^T$
    else
        $p_{\text{succ}} \leftarrow 0$
    $\sigma \leftarrow \sigma \cdot \exp\left(\frac{1}{d} \cdot (\frac{p_{\text{succ}} - p_{\text{target}}}{1 - p_{\text{target}}})\right)$
```

Contributions In this paper, we make the following contributions:

- We propose an experimental methodology supporting the design of a potential function capturing the dynamics of a variable-metric evolution strategy.
- We introduce the target step size of the (1+1)-CMA-ES as a key concept for constructing a suitable potential function.
- Based on the target step size, we define a potential function.
- We provide systematic empirical evidence for the suitability of the novel potential function. At the same time, we are in the position to highlight its weak spots, which might need to be addressed in future work.

Taking the above together, we provide a drift potential function that *potentially* can give rise to an analysis of the optimization behavior of a variable metric ES.

2 Theoretical Background

This section introduces the necessary background. First, we give a brief bird's eye introduction to drift analyses and its challenges. We then turn to invariance properties of CMA-ES, which are instrumental to the design of a Lyapunov potential.

2.1 Drift Analysis

Drift analysis goes back to Hajek [6]. It was later adapted to the specific needs of runtime analysis of randomized search heuristics [12,13]. The general idea of a drift theorem is to connect a statement about the expected single-step reduction of a potential function to the expected number of steps it takes to reduce the potential to a target value. Drift is a powerful concept, since the analysis is reduced to statements about the single-step behavior of the algorithm. Furthermore, drift theorems are not limited to expected values – they can also bound quantiles and hence control the tails of the runtime distribution. For further details, we refer the interested reader to [12,13].

Applying a drift argument amounts to the following steps: we define a potential function ϕ, show that the algorithm exhibits a certain type of expected progress with respect to that potential, and apply a drift theorem to turn the stepwise progress into a runtime bound. Let S denote the state space of the algorithm, s_t the sequence of algorithm states, and $\phi : S \to \mathbb{R}$ the potential. In the simplest case, the expected progress $\mathbb{E}[\phi(s_{t+1}) - \phi(s_t) | s_t = s]$ is bounded from below by a negative constant $-b$. If the progress is also bounded (or its tails controlled in a suitable way), then the so-called additive drift applies, yielding an expected runtime of $\mathbb{E}[T] \leq \frac{a}{b} + \text{const}$, where $a = \phi(s_0) - \phi_{\text{target}}$ is the potential difference to be crossed and T is the so-called first hitting time of the event $\phi(s_T) \leq \phi_{\text{target}}$.

Defining a suitable potential function is an art, not a science. The job of the potential is to capture progress of the algorithm across the whole state space S. For an evolution strategy, this is a non-trivial task because even elitist algorithms make nearly no progress if the step size is either much too small or much too large, or if the covariance matrix is unsuitable. Then with high probability, the algorithm adapts its distribution parameters towards more suitable values, hence bringing them closer to the regime where progress towards the optimum is achieved. Strategy adaptation does not yield immediate progress in terms of objective function improvement, but rather in terms of the potential to achieve such improvements in the future. Therefore, a suitable potential function needs to capture not only the goal of minimizing the objective function when the step size is well adapted, but also the goal of adapting step size and covariance matrix towards a regime where this is the case. We will discuss a corresponding potential function design in section 3.

Compared with a simple (1+1)-ES, this task is considerably harder when covariance matrix adaptation is involved. The ability to adapt the covariance matrix has the benefit of gaining invariance to affine transformations, which yields more general results in terms of the class of objective functions covered. The price to pay is that it takes away some symmetries, which increases the dimension of the normalized state. Moreover, CMA interacts with step size adaptation in non-trivial ways. However, depending on the tightness of the resulting bound, we may or may not need to capture all of these dependencies in a potential function.

2.2 Invariances

The goal of this section is to reduce the dimension of the state space. We will describe the reduced space by means of a normal form with easy-to-interpret state variables. The reduction also makes sampling a grid of states feasible.

We consider the (1+1)-CMA-ES with parameters (m, C) of its multi-variate Gaussian sampling distribution $\mathcal{N}(m, C)$,[1] optimizing an objective function $f : \mathbb{R}^d \to \mathbb{R}$. The parameters (m, C) and the objective function f define a state of the algorithm, in the sense that this information determines the distribution of successor states. Therefore, we pack them into the tuple $\theta = (m, C, f)$. Given a state θ, we denote the state after a single iteration of (1+1)-CMA-ES as $\theta' = (m', C', f)$. The following definition captures the invariance properties of the (1+1)-CMA-ES algorithm:

Definition 1. *We say that two tuples $\theta_1 = (m_1, C_1, f_1)$ and $\theta_2 = (m_2, C_2, f_2)$ are equivalent, and we write $\theta_1 \sim_T \theta_2$, if there exist an affine transformation $T(x) = Ax + b$ and a strictly monotonically increasing function $h : \mathbb{R} \to \mathbb{R}$ such that it holds*

1. $m_2 = T(m_1) = Am_1 + b,$ *(affine invariance, mean)*
2. $C_2 = A^T C_1 A,$ *(affine invariance, covariance matrix)*
3. $f_2 = h \circ f_1 \circ T^{-1}.$ *(strictly monotone fitness transformations)*

In other words, given a representative $\theta = (m, C, f)$ of an equivalence class, then all other members of that class are of the form $(Am + b, A^T CA, h \circ f \circ T^{-1})$. The following lemma clarifies how the definition relates to invariance:

[1] Under slight misuse of notation, we incorporate the step size into the covariance matrix at this point, writing C instead of $\sigma^2 C$ from now on. The parameter σ is re-introduced in the normal form, see equation (1).

Lemma 1. *Consider a sequence of points $x_1, \dots, x_n$ ordered by their f_1-ranking: $f_1(x_1) \leq f_1(x_2) \leq \cdots \leq f_1(x_n)$. Then the transformed points $y_1 = T(x_1), \dots, y_n = T(x_n)$ have an equivalent f_2-ranking, i.e., it holds $f_2(y_1) \leq f_2(y_2) \leq \cdots \leq f_2(y_n)$.*

Proof. The proof amounts to plugging the definition into the formulas of the lemma. We obtain $f_2(y_k) = f_2(T(x_k)) = h(f_1(T^{-1}(T(x_k)))) = h(f_1(x_k))$ and we note that h does not change the ranking. □

Plugging properties (1) and (2) of the definition into the PDF of the multivariate normal distribution yields the same result as the transformation theorem for densities applied to T. Hence, the PDFs are simply transformed into each other by means of T. Together with the above lemma, this implies that performing a step from θ_1 to θ'_1 and the three-step sequence of transforming θ_1 into θ_2, performing a step from θ_2 to θ'_2, and finally transforming θ'_2 back into θ'_1, yield the same result if the same randomness is used, and the same distributions in any case. In short: if we understand the algorithm dynamics starting in θ_1 then the insight immediately transfers to θ_2 and to the whole equivalence class.

We use this notion of invariance in two different ways, namely to formulate a potential that respects invariances, and for efficient sampling. In both cases, the goal is to reduce the number of cases for the subsequent analysis. This is achieved by analyzing only one state per equivalence class.

From now on, we consider a general convex quadratic objective , unction $f(x) = \frac{1}{2}(x - x^*)^T H(x - x^*) + c$ with optimizer x^*, Hessian H, and optimal value c. This case is of great interest, since it is a second order approximation of a local optimum of a twice continuously differentiable objective function. The first application of invariance is to simplify f. We can set $c = 0$ since the offset does not impact the ranking. Setting $T(x) = Ax + b$ with $A = H^{-1/2}$ and $b = -Ax^*$, we see that (m, C, f) is equivalent to $(Am + b, ACA^T, s)$, where $s(x) = \frac{1}{2}\|x\|$ is the sphere function. In other words, in order to understand the optimization behavior of a variable-metric evolution strategy on all convex quadratic objective functions, it suffices to consider the sphere function, as long as we consider general initial conditions. This greatly simplifies the task of designing a drift potential.

The remaining invariances are known as rotation and scale invariance. They refer to transformations for which A is a scaled orthogonal matrix and b is zero. Since the objective function is fixed, we write $\theta = (m, C)$ in the following, with Θ forming the space of all algorithm states. For the two-dimensional case, we define a section through the quotient space $\Theta / \sim_T$. Equivalently, it can be considered a normal form for states. First of all, we use the scaling degree of freedom to turn m into a unit vector. We then use the rotation degree of freedom to diagonalize the covariance matrix C. By swapping the axes of the coordinate system and flipping the axes individually, we can transform m to the form $m = (\cos(\alpha), \sin(\alpha))$ with $\alpha \in [0, \pi/2]$. Furthermore, we can decompose the (diagonal) covariance matrix into scale and shape components. Hence, without loss of generality, we can write all relevant states in the form

$$m = \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix} \qquad C = \sigma^2 \begin{pmatrix} \kappa & 0 \\ 0 & \frac{1}{\kappa} \end{pmatrix} \tag{1}$$

with *normalized step size* $\sigma > 0$, *approach angle* $\alpha \in [0, \pi/2]$, and *eccentricity* $\kappa \geq 1$ encoding the shape of the distribution. An algorithm for transforming any state into the normal form is found in the online supplement[2].

3 Construction of a Lyapunov Potential Function

To construct a potential function that yields drift everywhere in the state space, each part of the algorithm's actions needs to be rewarded. By design (elitism), the distribution mean never moves away from the optimum. This ensures that the algorithm will never lose progress in terms of distance to the optimum. However, for increasingly bad parameter settings (too small or large σ or too large κ) the progress rate quickly approaches zero. In order to avoid vanishing drift, we want the potential function to account for the quality of the strategy parameters. We design building blocks for such a function in the following.

3.1 Target σ

For each setting of α and κ we define a *target* σ. The meaning of the target is that the algorithm adjusts the parameter towards this value while (artificially) keeping the other parameters fixed. This does not mean that the target value is optimal in terms of optimization progress; instead it reflects the adaptation actually performed by the algorithm. We denote the target as a function $\sigma^*(\alpha, \kappa)$. We omit the parameters in the following.

3.2 Potential Function

Our potential function is defined in terms of the normal form:

$$V(\theta) = V(m, \kappa, \sigma) = \underbrace{\log(||m||)}_{(1)} + \underbrace{v_1 \cdot |\log(\kappa)|}_{(2)} + \underbrace{v_2 \cdot f_\kappa\left(\left|\log\left(\frac{\sigma}{\sigma^*}\right)\right|\right)}_{(3)}$$

$$f_\kappa(x) = \begin{cases} 1 - \exp\left(-s \cdot \log\left(\frac{x}{w_\kappa}\right)^2\right) & \text{if } x \geq w(\kappa), \quad s = \frac{1}{100}, w_\kappa = 1 + \frac{\log(\kappa)}{10} \\ 0 & \text{else} \end{cases}$$

Here, $v_1, v_2 > 0$ are tuning parameters to be determined later. There are three dimensions in which the CMA-ES can make progress, namely the distance to the optimum m, the eccentricity of the matrix κ, and the (normalized) step size σ. Those dimensions are taken care of in the potential function separately:

[2] https://github.com/RUB-INI-Theory-of-Machine-Learning/EmpiricalDriftAnalysis/blob/main/Empirical_Drift_Analysis___Supplements.pdf

m: The term $\log(||m||)$ describes optimization progress by finding a point that is closer to the optimum. For "well-adapted" parameters of the algorithm, it should make significant expected progress in this sense. We refer to this term as the $\log(||m||)$-*term*.

κ: The second term of the potential function determines progress by adjusting the eccentricity κ towards the minimal eccentricity of 1, encoding an isotropic distribution. We refer to this term as the κ-*term*.

σ: The third term of the potential function measures progress by adjusting the step size σ towards the stable step size σ^*. The term should become dominant if σ is far away from σ^*. We refer to this term as the σ-*term*. The activation function f_κ asymptotically approaches the identity, but it is flat in a neighborhood of zero, with a differentiable transition (see Figure 1).

4 Experiments

This section describes the experimental setup to gain empirical data on two types of quantities: the target step size and a lower bound of the expected drift of the potential function.

4.1 Target Step Size

To obtain values for the target step size σ^*, we conduct the following experiment: First we prepare a lattice of 64 linearly spaced α-values from 0 to $\pi/2$ and 2048 geometrically spaced κ-values from 1 to 2000. Then, for every point on the lattice, we transform the normal form into the parameter form (σ is set to 1 in the beginning) and initialize the algorithm with this state. We then conduct a step of the algorithm and transform the resulting state back into the normal form, however, omitting the rescaling to $|m| = 1$, and resetting α and κ to their initial value. After 50,000 iterations for reaching the limit distribution we record σ for further 1,000,000 iterations. In the end we compute the geometric mean of the recorded values. We end up with a lattice of target step sizes σ^*. For parameters that are not on the grid, we compute σ^* with bilinear interpolation.

4.2 Drift Experiments

We perform three experiments, one where the focus lies on understanding the drifts of the potential function around sensible values for (α, κ, σ) and the other two, with a much wider but less dense grid for investigating the potential function's boundary behaviour, i.e., when σ or κ approach ∞ or σ approaches 0.

Sampling Parameters We define a grid for each parameter (α, κ, σ) and combine those into a three-dimensional product grid. The parameters for the dense grid are shown in table 1 and for the wide grids (κ-grid, σ-grid) in table 2. We then take each of these states and perform a Monte-Carlo simulation to obtain the expected value of the drift at that point.

Table 1: Dense Grid κ

param.	range	steps	spacing
α	[0, $\pi/2$]	24	linear
κ	[1, 10]	128	logarithmic
σ	[0.1, 10]	256	logarithmic

Table 2: Wide Grids (κ-grid, σ-grid)

param.	range	steps (κ/σ)	spacing
α	[0, $\pi/2$]	12/12	linear
κ	[1, 1000]	512/24	logarithmic
σ	[0.01, 100]	24/512	logarithmic

Significance and Precision Since we work with a fixed number of random samples to estimate the drift, we are interested in the quality of those estimations. To that end we perform a one-sided t-test. Let d denote the (observed) population mean. For a fixed candidate precision ϵ, the null hypothesis is that the distributions of d and $d + \epsilon$ are overlapping, while the alternative hypothesis is that $d + \epsilon$ is significantly larger than d. We then perform a golden-section search to find a value of ϵ where the t-test returns a p-value of at most 0.001. This establishes a 99.9% confidence that the observed drifts are not smaller than the precision ϵ. However, due to multiple testing, this does not imply that the overall confidence is 99.9%. In the dense grid run, we evaluate $786,432$ samples, out of which we would statistically expect the null hypothesis to be true for approximately 787.

Potential Function Parameters To determine the optimal weights v_1, v_2 of the second and third term in the potential function we used CMA-ES. We used the sum of the experimentally obtained drift-values and the precision-values as drift-values and optimized for the largest drift-value to be minimal, i.e., for the gap between uniform drift and zero to be maximal.

5 Results

In this section, we show our empirical results. We start with the target step size σ^*. Following that, we analyze the results of the dense grid dataset (Table 1). Finally, we present the results of the wide grid experiments (Table 2) to examine the boundary behaviour.

5.1 Target Parameters

In Figure 2 we present the target step size σ^* for 64 linearly spaced values of α between 0 and $\pi/2$. We notice significant differences with respect to the approach angle α, and a decreasing trend for growing eccentricity κ.

5.2 Drift Analysis

We observe that the difference between close α values is small, indicating that the drift is continuous. Furthermore, it is also monotonic for the most part. For

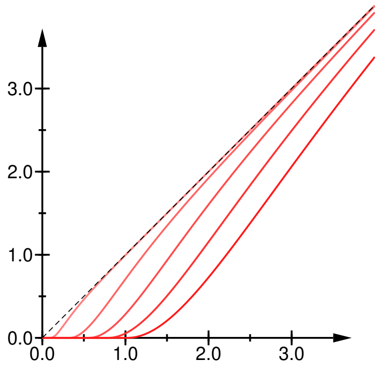


Fig. 1: Graphs of the "activation" function f_κ for $\kappa \in \{1, 10, 100, 1000, 10000\}$, from left to right.

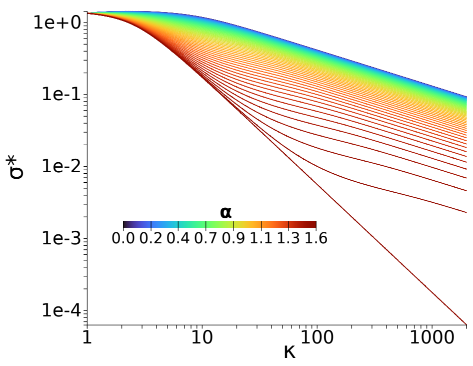


Fig. 2: Graphs of Target Sigma, as a Function Functions of the Remaining Parameters.

that reason, we will only present the extreme cases of $\alpha = 0$ and $\alpha = \pi/2$ in the following figures, as this provides a sufficient impression of the results. The plots do not include the precision values added to them, since these are always at least two orders of magnitude smaller than the drift values. The plots of the full data as well as the precision values can be examined in the supplementary material.[3]

Figure 3 shows the legend used for the following heatmap plots. The color scale of each plot is scaled to the minimum and maximum drift for that plot. A blue color indicates negative values, i.e., positive drift (desirable), while red color indicates positive values, i.e., negative drift (undesirable). Values close to 0 appear white.

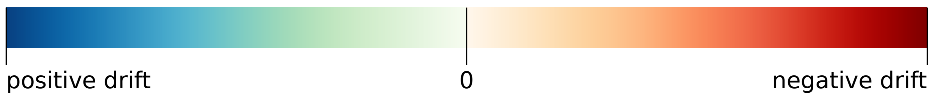
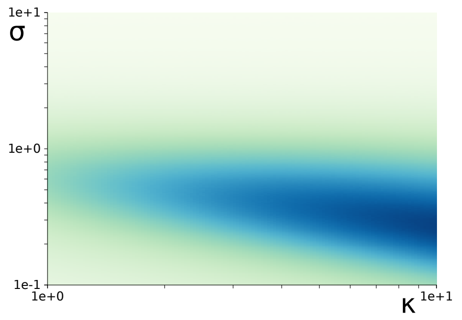


Fig. 3: The Heatmap Scale

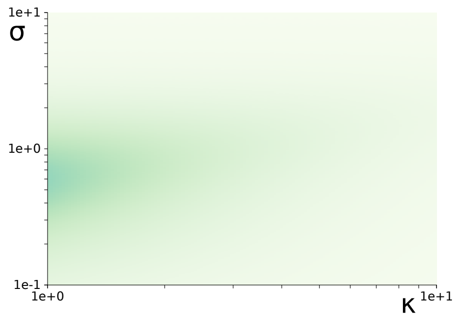
Weights The resulting weights from the CMA-ES optimization for the drift terms differ for each experiment, however $v_1 = 1.7$ and $v_2 = 3.14$ prove to yield good results across all datasets. All following plots use these weights.

[3] https://github.com/RUB-INI-Theory-of-Machine-Learning/EmpiricalDriftAnalysis/tree/main/plots

The log(||m||)-term – Optimization Progress We first present the result of the overall drift from the dense grid run. We see that for $\alpha = 0$ the overall drift is positive. Furthermore, for large values of κ we notice that for $0.05 < \sigma < 1$ a region with especially large drift emerges. For $\alpha = \pi/2$ we also see an overall positive drift with slightly larger drift values for very small κ values and $0.3 < \sigma < 1$. The minimal drift (gap to zero) in this dataset is ≈ -0.00198.



(a) $\alpha = 0$



(b) $\alpha = \pi/2$

Fig. 4: The $\log(||m||)$-term Progress

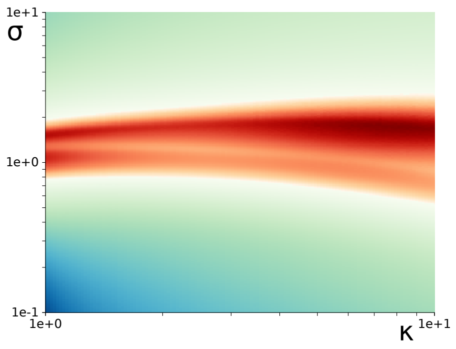
However, we notice for both α values and for increasing σ values decreasing drift, while for very small σ the drift diminishes. Since the goal of (additive) drift analysis is to find a potential function that provides a lower bound on the drift, this trend defeats that purpose. Therefore we will now look at the results where the σ-term of the potential function is added.

Adding the σ-term – Behaviour on the Boundaries The σ-term of the potential function rewards progress for changing the σ value towards σ^*. Figure 5 shows the drift of the σ-term in isolation.

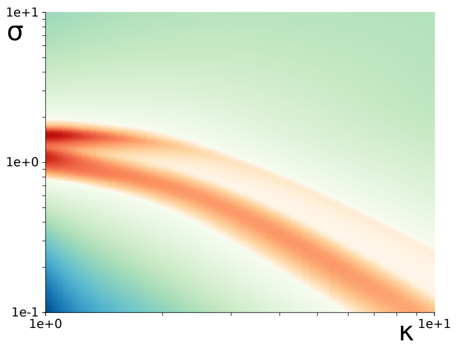
We notice a large red strip in the results where the drift is negative. This corresponds to the target σ values. When the algorithm's σ parameter is already at or very close to the target value σ^* then any change results in negative drift.

The negative drift around σ^* stems from a *moving target* problem. Even though the algorithm adapts its σ parameter towards σ^*, in the new algorithm state that value has changed (because α and κ were adapted by the algorithm). This effect is present everywhere, however in the cases where the drift becomes negative this effect is so strong that it dominates the otherwise favourable σ adaptation of the algorithm. We used a filter f_k in this term to alleviate this effect. Although the effect is still present, it produces far less negative drift than without the filter.

Besides that, we also witness that for large values of σ the σ-term shows a stable drift. In Figure 6 we see the combined drift of the $\log(||m||)$-term and the σ-term. In Figure 7 we plotted the influence of the σ term on the overall drift. We

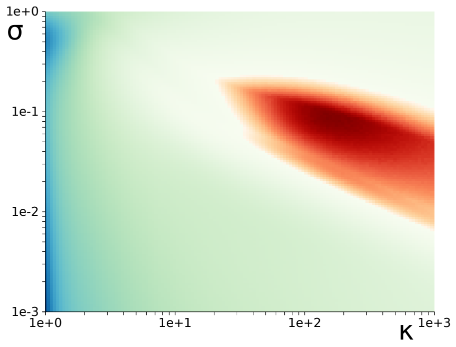


(a) $\alpha = 0$

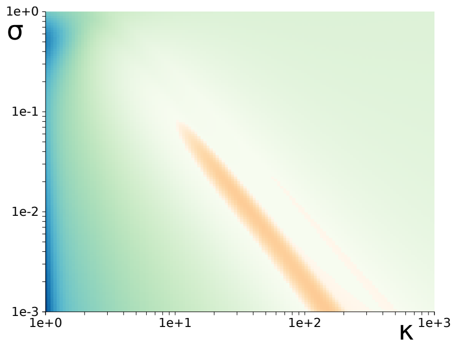


(b) $\alpha = \pi/2$

Fig. 5: Sigma Progress



(a) $\alpha = 1.5404$

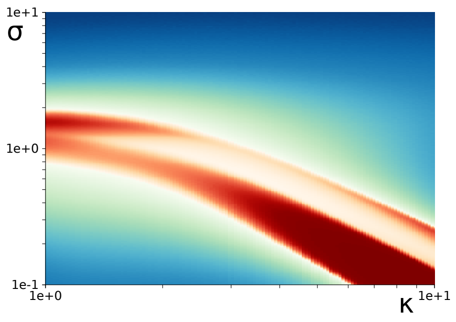


(b) $\alpha = \pi/2$

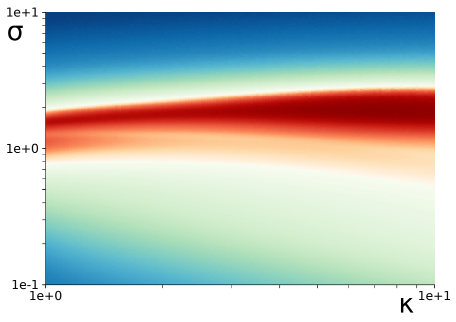
Fig. 6: Drift of the $\log(||m||)$-term and the σ-term combined. Note that this graph shows a larger grid to present the problematic areas.

notice that for large σ the third term dominates the overall drift. This is what we hoped to see, as this counteracts the diminishing drift of the $\log(||m||)$-term. This also makes sense from the perspective of an ES, since the search distribution is misaligned and needs to improve. The same effect can be observed for small σ in the bottom left corner, where the moving target problem does not dominate the drift. In Figure 6a and 6b, we see that for large α, large κ and small σ the drift from the first term is not large enough to make up for the negative drift of the third term. Because of that the κ-term is necessary, which we will add next.

Adding the κ-term – Fixing Moving Targets In Figure 8 we present the drift for the κ-term in isolation. Similar to the negative drifts for the σ-term, there also exist regions with negative drift. Since this term only gratifies the eccentricity to become smaller, naturally there are configurations where the algorithm has negative drift. This is due to the fact that for small α and reasonable



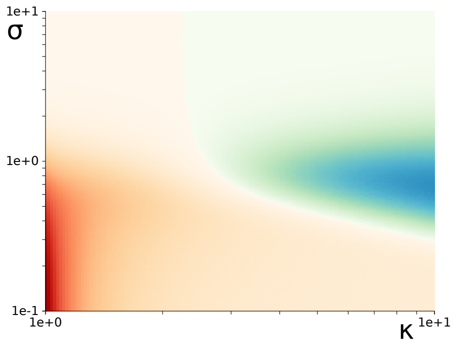
(a) $\alpha = 0$



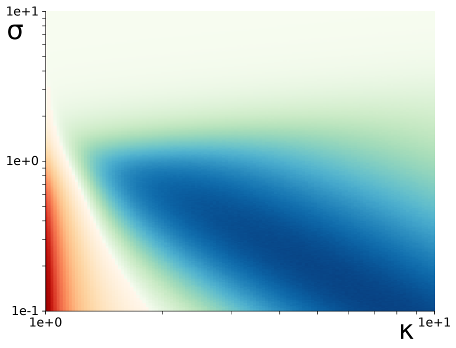
(b) $\alpha = \pi/2$

Fig. 7: The Sigma Influence in a Percentile View. The influence is the drift of the σ-term divided by the sum of the absolute values of the $\log(||m||)$-term and the σ-term.

σ the algorithm is making better progress by becoming more eccentric. When κ becomes exceedingly large the algorithm does not profit from eccentricity anymore and is inclined to make κ smaller. For large α this point is reached for smaller κ. Furthermore, when $\kappa = 1$ the algorithm can only get worse. This corresponds the red regions on the bottom left.



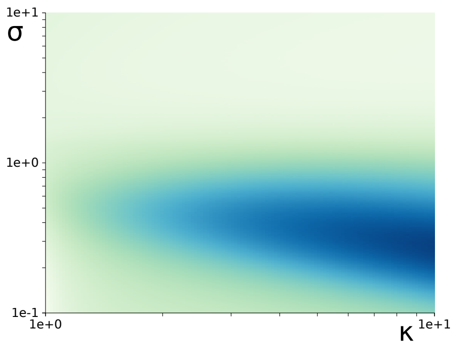
(a) $\alpha = 0$



(b) $\alpha = \pi/2$

Fig. 8: Kappa Progress

The Final Result - Adding It all together We now add the κ term to see the drift of the complete potential function, which is shown in Figure 9. For small α we continue to see a region with strong drift for larger κ and $\sigma < 1$. Overall we see a moderate amount of drift everywhere, and no regions with negative drift. The minimal drift value for the complete potential function is ≈ -0.0063 which is three times as much as the minimal drift value of just the $\log(||m||)$-term in the same region of the state space.

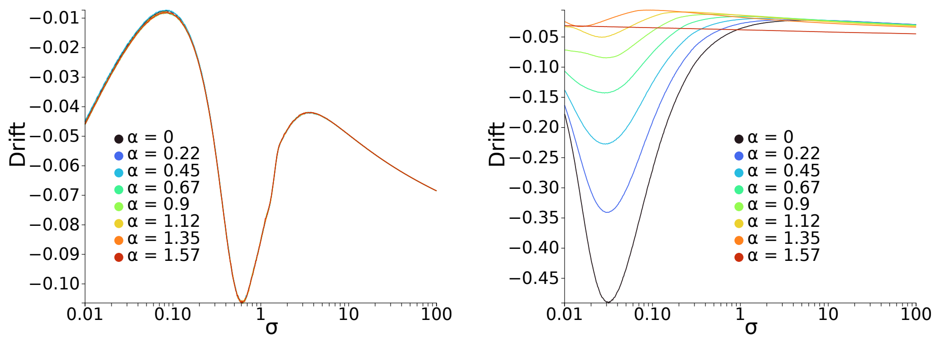


(a) $\alpha = 0$ (b) $\alpha = \pi/2$

Fig. 9: The Drift of the Complete Potential Function $V(\theta)$

5.3 Asymptotic Behaviour

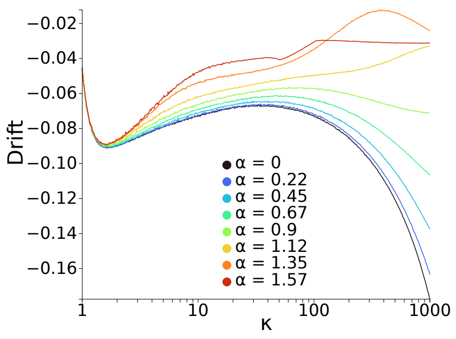
Even though we observed an improved drift for extreme values of κ and σ, there is still a small decrease visible as the parameters become more extreme. We believe that this trend will saturate such that we can guarantee a lower bound for the drift. In Figures 10 and 11 we present the result of the experiments that probe deeper into the parameter space (Table 2).



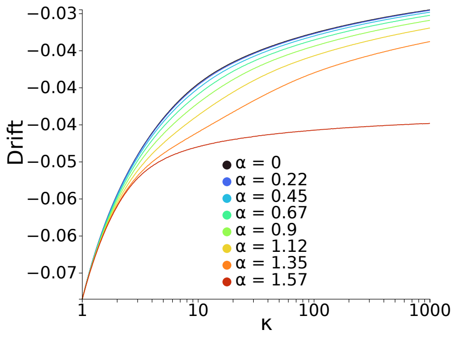
(a) $\kappa = 1$ (b) $\kappa = 1000$

Fig. 10: The Drift for a Single κ along the σ Axis

We leave Figure 10 for visual inspection to the reader. However, we believe to identify the asymptotic behaviour of the drift. Even though in Figure 10b towards smaller σ the drift is decreasing, we suspect that if the experiment had an even wider grid, we would observe a behaviour similar to the one in Figure 10a.



(a) $\sigma = 0.01$



(b) $\sigma = 100$

Fig. 11: The Drift for a Single σ along the κ Axis

Figure 11a displays increasing drifts for most α and at least stable drifts for some α, while Figure 11b shows decreasing drifts at $\sigma = 100$ and at least suggests an asymptotic behaviour towards large κ. A more detailed investigation of asymptotic effects needs better noise handling, and will be subject to future work.

6 Conclusion

We have filled a gap in the analysis of evolution strategies by proposing a drift potential function for a variable metric evolution strategy. In general, designing a potential for drift analysis is a difficult task. Our function involves an auxiliary function, namely target states of the scale parameter σ of the search distribution, which is interesting to investigate in its own right. Our empirical analysis shows that the novel potential works well in the sense that it yields negative drift everywhere, and that it is bounded away from zero.

Naturally, our result has limitations. We consider only two-dimensional search spaces since our sampling-based approach scales badly to higher dimensions. While Monte Carlo simulations leave space for random effects, our huge sample size yields high confidence. Inter- and extrapolation from a fixed parameter grid may induce inaccuracies. However, the smoothness of all observed effects indicates that the grid is well-chosen, and that our results do indeed generalize—at least qualitatively—to the full continuous and unbounded state space.

Although our study is empirical in nature we believe that it can serve three distinct goals: it increases the trust into the reliability of variable metric evolution strategies like CMA-ES, it enhances our understanding of their behavior, and it paves the way for an actual mathematical convergence proof based on drift arguments.

References

1. Youhei Akimoto, Anne Auger, and Tobias Glasmachers. Drift theory in continuous search spaces: expected hitting time of the (1+1)-es with 1/5 success rule. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 801–808, 2018.
2. Youhei Akimoto, Anne Auger, Tobias Glasmachers, and Daiki Morinaga. Global linear convergence of evolution strategies on more than smooth strongly convex functions. *SIAM Journal on Optimization*, 32(2):1402–1429, 2022.
3. Pauline Bennet, Carola Doerr, Antoine Moreau, Jeremy Rapin, Fabien Teytaud, and Olivier Teytaud. Nevergrad: black-box optimization platform. *ACM SIGEVOlution*, 14(1):8–15, 2021.
4. Hans-Georg Beyer. *The Theory of Evolution Strategies.* Springer Science & Business Media, 2001.
5. Claudia R Correa, Elizabeth F Wanner, and Carlos M Fonseca. Lyapunov design of a simple step-size adaptation strategy based on success. In *Parallel Problem Solving from Nature–PPSN XIV: 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings*, pages 101–110. Springer, 2016.
6. Bruce Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied probability*, 14(3):502–525, 1982.
7. Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, pages 1689–1696, 2010.
8. Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.
9. Christian Igel, Thorsten Suttorp, and Nikolaus Hansen. A computational efficient covariance matrix update and a (1+1)-cma for evolution strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, volume 1, pages 453–460, 07 2006.
10. Jens Jägersküpper. Algorithmic analysis of a basic evolutionary algorithm for continuous optimization. *Theoretical Computer Science*, 379(3):329–347, 2007.
11. Stefan Kern, Sibylle D Müller, Nikolaus Hansen, Dirk Büche, Jiri Ocenasek, and Petros Koumoutsakos. Learning probability distributions in continuous evolutionary algorithms–a comparative review. *Natural Computing*, 3(1):77–112, 2004.
12. Per Kristian Lehre and Carsten Witt. General drift analysis with tail bounds. *arXiv preprint arXiv:1307.2559*, 2013.
13. Johannes Lengler. Drift analysis. *Theory of evolutionary computation: Recent developments in discrete optimization*, pages 89–131, 2020.
14. Daiki Morinaga and Youhei Akimoto. Generalized drift analysis in continuous domain: linear convergence of (1+ 1)-es on strongly convex functions with lipschitz continuous gradients. In *Proceedings of the 15th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*, pages 13–24, 2019.
15. Daiki Morinaga, Kazuto Fukuchi, Jun Sakuma, and Youhei Akimoto. Convergence rate of the (1+ 1)-evolution strategy on locally strongly convex functions with lipschitz continuous gradient and their monotonic transformations. Technical Report arXiv:2209.12467, arXiv.org, 2022.
16. Peter S Oliveto and Xin Yao. Runtime analysis of evolutionary algorithms for discrete optimization. In *Theory of Randomized Search Heuristics: Foundations and Recent Developments*, pages 21–52. World Scientific, 2011.