Additive Drift as an Optimization Problem for the (1+1)-ES

Alexander Jungeilges
Institute for Neural Computation
Faculty of Computer Science
Ruhr-University Bochum
Bochum, Germany
alexander.jungeilges@ini.rub.de

Tobias Glasmachers
Institute for Neural Computation
Faculty of Computer Science
Ruhr-University Bochum
Bochum, Germany
tobias.glasmachers@ini.rub.de

Abstract

We present a novel method for constructing a close-to-optimal potential function for drift analysis of evolution strategies and other numerical optimization algorithms. It is based on a piecewise linear ansatz for the potential function, defined by potential values on a discrete grid of states together with a corresponding interpolation scheme. We compute or approximate the algorithm dynamics on the grid and then construct a linear program from expected progress and (weighted) state transitions. The solution of the optimization problem gives rise to parameters of the potential function maximizing (additive) drift. Under mild assumptions, the proceeding yields an arbitrarily close approximation of the optimal potential function at the expense of growing computational demands. As a proof of concept we apply the method to the (1+1) evolution strategy on the two-dimensional sphere function.

CCS Concepts

• **Theory of computation → Random search heuristics**; **Theory of randomized search heuristics**.

ACM Reference Format:
Alexander Jungeilges and Tobias Glasmachers. 2025. Additive Drift as an Optimization Problem for the (1+1)-ES. In *Foundations of Genetic Algorithms XVIII (FOGA '25), August 27–29, 2025, Leiden, Netherlands.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3729878.3746612

1 Introduction

Drift analysis is a major paradigm for establishing runtime guarantees for randomized search heuristics, and for evolutionary algorithms in particular [13, 14]. The method is particularly successful for algorithms operating on discrete domains, but it was also applied to evolution strategies (ES) [1]. Drift arguments were used implicitly already by Jägersküpper [9] to analyze the convergence behavior of the (1+1)-ES on spherical and convex quadratic functions. He provided the first linear convergence proof, yet, his results were strictly asymptotic and provided no explicit runtime bound.

Akimoto et al. [1] established the groundwork on which this work is built. They developed drift theory on a continuous domain and provided the first non-asymptotic analysis, proving linear convergence and a convergence rate of $\Theta(1/d)$ for the sphere function in (finite) dimension d. A crucial part was the construction of a non-trivial potential function for proving two drift conditions for the upper and lower bound on the runtime. The proof involved the somewhat unnatural construction of a truncated process (cutting off particularly successful steps) for controlling drift in light of the domain of the potential, which is unbounded from below.

The studies [2, 15, 16] refine the methodology and greatly extended its scope to a wide class of problems, in particular to all strongly convex objective functions with Lipschitz continuous gradients. However, a similar generalization to other algorithms turns out to be challenging.

Nonetheless, the grand goal of establishing linear convergence of CMA-ES at a problem-independent rate was very recently achieved with an analysis based on Markov chains [7]. While their analysis also employed a potential function and analyzed its one-step behaviour, the results were derived through various properties of the normalized homogeneous Markov chain. It would be very useful to obtain a corresponding result based on drift, because the method allows for much stronger statements about the actual runtime, about the time it takes to adapt the covariance matrix, and about the dependency of these times on the problem and algorithm parameters. In contrast, the convergence rate derived in [7] remains implicit. To distinguish between a Markov chain framework and drift analysis, we denote the potential function as the function designed for applying a drift theorem.

As of today there is no drift analysis even of the simplest variant of the covariance matrix adaptation evolution strategy (CMA-ES) with unrestricted covariance matrix. In particular, the only attempt of constructing a potential function suitable for showing that CMA-ES can converge at a problem-independent rate is limited to an empirical study based on Monte Carlo simulations [6].

In general, a major challenge and a potential road-blocker for drift analysis is the construction of a suitable potential function that a) exhibits drift, and b) is simple enough to allow for the application of analytic arguments. In this paper, we establish a novel approach for constructing such a potential function. Our potential function gives rise to uniformly positive drift suitable for applying an additive drift theorem. Its special property is that the resulting drift is arbitrarily close to *optimal*. By that property we mean that the resulting runtime bounds are close to optimal, or optimally tight across all runtime bounds that are achieved by drift.

At its core, our approach is rooted in an empirical analysis framework. Still, it is suitable for obtaining rigorous runtime guarantees. We sketch the full route for turning our analysis into provable drift and resulting runtime guarantees. In this paper, we put a particular emphasis on the arising numerical challenges.

In the next section we present a general form of a potential function describing the behavior of elitist evolution strategies, as

FOGA '25, Leiden, Netherlands

ACM ISBN 979-8-4007-1859-5/2025/08
https://doi.org/10.1145/3729878.3746612

well as a corresponding drift theorem. We then establish our core construction, yielding a close-to-optimal potential function. We demonstrate an application of the framework to the (1+1)-ES in section 4, and present corresponding numerical results in section 5.

2 Algorithm & Methodology

In this section, a probabilistic elitist adaptive algorithm will be formulated. We further introduce the first hitting time through an additive drift theorem and discuss the notion of an optimal potential function.

2.1 A generic stochastic search algorithm

We introduce an algorithm template that fits many elitist evolution strategies as well as several other algorithms in the continuous domain. We leave several mechanisms open, like the population size λ, the offspring distribution (i.e., whether or not a full covariance matrix is used), and how the distribution is updated. We model our algorithm as a stochastic recursively defined sequence on a state space Θ. In each iteration $t \in \mathbb{N}$, λ new candidate solutions are drawn from a sampling distribution $\mathcal{P}(\theta^{(t)})$, with $\theta^{(t)} \in \Theta$ denoting the current state of the algorithm. Θ encompasses the full algorithm state including all adaptable parameters of the sampling distribution. For example, the state of the (1+1)-ES with 1/5-success rule [1] is $(m^{(t)}, \sigma^{(t)}) \in \mathbb{R}^d \times \mathbb{R}^+$. To obtain the recursive sequence, a *transition function* $\mathcal{F}$ is defined. Given $\theta^{(0)}$, future states are computed by

$$\theta^{(t+1)} = \mathcal{F}(\theta^{(t)}, X^{(t)}; f), \tag{1}$$

where $X^{(t)}$ is the set of offspring samples in generation t, f is the fitness, and $\mathcal{F}$ encodes the (deterministic) computations performed by the algorithm. An example for $\mathcal{F}$ for the (1+1)-ES with success-based step-size adaptation can be found in [15], which is picked up in section 4.

We demand two properties of $\mathcal{F}$:

- The algorithm shall follow an elitist design, i.e., it maintains a non-increasing sequence of current best solutions $\{m^{(t)}\}_{t\in\mathbb{N}}$, with which stopping criteria of the optimization progress are checked. For a $(\mu+\lambda)$ setup, an additional mechanism (like global intermediate recombination) is required to determine $m^{(t)}$ from $\theta^{(t)}$.
- The algorithm shall be rank-preserving, i.e., $\mathcal{F}$ is invariant under strictly increasing transformations of objective values.

The construction covers evolution strategies with $(1+1)$ or $(1+\lambda)$ selection. It yields a Markov chain $\{\theta^{(t)}\}_{t\in\mathbb{N}}$ describing the state of algorithm, including the underlying sampling distribution.

Algorithm 1 Probabilistic adaptive algorithm

```
input $\theta_0, f : \mathbb{R}^d \to \mathbb{R}$
for $t = 0, 1, 2, \ldots,$ until stopping criterion is met do
    sample $x_1, \ldots, x_\lambda \sim \mathcal{P}(\theta^{(t)})$
    evaluate $f(x_1), \ldots, f(x_\lambda)$
    $\theta^{(t+1)} \leftarrow \mathcal{F}(\theta^{(t)}, \{x_1^{(t)}, \ldots, x_\lambda^{(t)}\}; f)$
```

2.2 Additive Drift in a Continuous Domain

Previous applications of drift theory in continuous domains highlighted the logarithmic distance to the optimum in the potential function as the main contributor for a linear convergence guarantee [1, 2]. Our potential function employs the same strategy. In the case of convergence, it decays towards negative infinity, combined with a mechanism that penalizes sub-optimal parameter values. Additive drift is a natural tool for the analysis of this construction.

Akimoto et al. [1] extended the theory of additive drift towards continuous domains. To deal with the difficulty of an unbounded domain, they introduced the notion of a *truncated process*, which upper-bounds the possible single-step progress made by the actual process. We refer to [1] for a detailed discussion. In our analysis, we instead make use of an alternative additive drift theorem, introduced by Doerr and Kötzing [5, 11]:

Theorem 1 (Additive Drift, Upper Bound with Overshooting). *Let $(X^{(t)})_{t\in\mathbb{N}}$ be an integrable processs over $\mathbb{R}$, and let $T = \inf\{t \in \mathbb{N} \mid X^{(t)} \le \beta\}$ for some $\beta \in \mathbb{R}$.*[1] *Furthermore, there is a $\delta > 0$ such that, for all $t < T$, it holds that*

$$0 < \delta \le \mathbb{E}[X^{(t)} - X^{(t+1)} \mid X^{(0)}, \ldots, X^{(t)}]. \tag{2}$$

Then

$$\mathbb{E}[T] \le \frac{\mathbb{E}[X^{(0)}] - \mathbb{E}[X^{(T)}]}{\delta}. \tag{3}$$

By dropping the requirement of a non-negative process and introducing the overshooting term $\mathbb{E}[X^{(T)}]$ (replacing β) into the classical additive drift theorem, we overcome the difficulty of controlling extreme events.

We argue that this drift theorem is beneficial in two ways: by neglecting the truncated process, we investigate a more realistic model of the algorithm and one-step progress, as there is no "magic" constant that cuts the progress at an arbitrary point. Furthermore, the notation and calculations simplify as only the term $X^{(t)} - E[X^{(t+1)} \mid X^{(0)}, \ldots, X^{(t)}]$ needs to be investigated. As a compensation, the term $E[X^{(T)}]$ needs to be bounded from below.

2.3 Optimal Additive Drift

The following definition captures our notion of an optimal potential function, in the sense of providing a perfectly tight runtime bound. A similar concept is found in [12, 14].

Definition 1. *Consider a Markov chain $(\theta^{(t)})_{t\in\mathbb{N}}$ on Θ with a set Θ^* of terminal states stopping the process. We define the* ***optimal potential function*** *$V^* : \Theta \to \mathbb{R}_0^+$ for the Markov chain as the first hitting time*

$$V^*(\theta) = \mathbb{E}\left[\inf\left\{t \in \mathbb{N} \,\middle|\, \theta^{(0)} = \theta \text{ and } \theta^{(t)} \in \Theta^*\right\}\right]$$

of Θ^*.

Of course, the optimal potential is not necessarily a practical choice for analyzing an actual algorithm because constructing V^* as an explicit function of θ is expected to be an utterly impossible task in all but the simplest cases.

[1] The original theorem in [11] uses a threshold of zero instead of the slightly more flexible formulation using β.

It obviously holds $V^*(\theta) = 0$ for all $\theta \in \Theta^*$. We assume in the following that Θ^* is chosen so that V^* is nowhere infinite (i.e., the chain stops eventually with full probability).

Lemma 1. *Consider the optimal potential function V^* for the Markov chain $\theta^{(t)}$, and assume that $V^*(\theta) < \infty$ for all $\theta \in \Theta$. Let θ' denote the successor state of $\theta \in \Theta \setminus \Theta^*$, and let $\Delta(\theta) = V^*(\theta) - \mathbb{E}[V^*(\theta')]$ denote the additive drift. Then it holds $\Delta(\theta) = 1$ for all $\theta \in \Theta \setminus \Theta^*$.*

Proof. The statement follows immediately from applying the Markov property to the definition of V^* and using the law of total expectation:

$$\begin{aligned} V^*(\theta) &= \mathbb{E}[\inf\{t \in \mathbb{N} \mid \theta^{(0)} = \theta \text{ and } \theta^{(t)} \in \Theta^*\}] \\ &= \mathbb{E}[\inf\{t \in \mathbb{N} \mid \theta^{(1)} = \theta' \text{ and } \theta^{(t)} \in \Theta^*\}] \\ &= \mathbb{E}[\inf\{t \in \mathbb{N} \mid \theta^{(0)} = \theta' \text{ and } \theta^{(t)} \in \Theta^*\}] + 1 \\ &= \mathbb{E}[V^*(\theta')] + 1 \end{aligned}$$

□

The above lemma shows that the drift is constant. We can then set $\delta = 1$ in Theorem 1 and obtain a perfectly tight bound. In this sense the potential V^* is optimal for bounding the expected runtime of the Markov chain for reaching the terminal set. The same argument applies to all potential functions of the form $B \cdot V^* + A$ with constant drift $B > 0$ and setting $\delta = B$. This insight serves as the guiding principle of the construction in the following section.

3 Additive Drift as an Optimization Problem

This section introduces the main methodology of the paper. Based on our generic elitist stochastic search algorithm and the additive drift theorem, a numerically computable formulation of the drift will be derived, based on an optimization problem, that maximizes the pointwise drift across a grid of parameter values.

3.1 Potential Function

The potential function $V(\theta)$ is defined as follows. It generalizes the form that has been used several times for the analysis of the (1+1)-ES with success-based step-size adaptation [1, 2, 15].

Definition 2 (Potential Function). *For $\theta \in \Theta$ the potential function $V(\theta)$ is defined as*

$$V(\theta) = \log(f_\mu(m)) + Q(\theta) \tag{4}$$

where $Q : \Theta \longrightarrow \mathbb{R}_0^+$ is a scale-invariant function and f_μ denotes the spatial suboptimality function defined in [8].

The *spatial suboptimality function* f_μ assigns to each point m the measure μ of the sub-level set $\left\{x \in \mathbb{R}^d \,\middle|\, f(x) \leq f(m)\right\}$, where μ can be chosen as the Lebesgue measure if all level sets are bounded. Under mild regularity assumptions it can be thought of as a normal form of f among all rank-preserving transformations. Including the spatial suboptimality function into the potential was proposed by Morinaga et al. [15] as it naturally extends the class of functions that can be analyzed. They consider the d-th root of the Lebesque measure. On the spherical function centered around 0, $f_\mu(m)$ then reduces to $||x||$ up to a multiplicative constant, recovering the original norm that was used in previous studies.

The term $Q(\theta)$ is used to penalize sub-optimal parameter values of the distribution. It needs to be tailored carefully to the analyzed algorithm and objective function. For the (1+1)-ES with success-based step-size adaptation, several examples exist, ranging from the sphere function to the full class of convex quadratic functions [1, 2, 15].

With respect to the progress of the algorithm, the potential function represents the following behavior: the term $\log(||m||) = \log(f_\mu(m)) + \text{const}$ measures the actual optimization progress and decays to $-\infty$ when approaching the optimum. When the distribution is not well adapted, for example, for the (1+1)-ES, the normalized step-size $d \cdot \sigma/||m||$ is too small or large, the state is penalized by some positive value. Through the self-adaptation $\mathcal{F}$, the algorithm then corrects its parameters, on average, in a direction that reduces the penalty term, therefore yielding drift even if progress of $\log(f_\mu(m))$ is arbitrarily small.

Q will be modeled as a parametric family of functions. We introduce a parameter grid $G \subset \Theta$ as a discretization of our state space Θ. The exact nature of this finite set of supporting points depends on the algorithm and objective function that is analyzed. For example, the one-step progress of the (1+1)-ES with 1/5-success rule is fully described through the normalized step-size, independent of m, and therefore requires only a one-dimensional grid on $\mathbb{R}$, whereas a (1+1)-ES with full covariance matrix adaptation has a much larger state space for $d > 1$. Here, one might consider a high-dimensional grid in a non-Euclidean space.

Proceeding with the general setting, across the grid we obtain corresponding function values $q_1, \ldots, q_n$, with $q_k = Q(\theta_k)$ if $\theta_k \in G$. Hence, for fixed grid points, the class of potential functions is parametrized by a parameter vector $q = (q_1, \ldots, q_n)$, with $n = |G|$. For an arbitrary state θ, an interpolation scheme is used to compute $Q(\theta)$, i.e., for each θ, there exists a weight vector $w(\theta) \in \mathbb{R}^n$, such that it holds $Q(\theta) = q^T w(\theta)$. This leads to the insight that the parameter vector q can be used as a constant vector that is not tied to the state. The parametrized form of the potential for an arbitrary $q \in \mathbb{R}^n$ is then

$$V(\theta) = \log\left(f_\mu(m)\right) + q^T w(\theta). \tag{5}$$

The drift theorem requires only the investigation of the one-step behavior of the algorithm. To simplify the notation we denote the current state $\theta^{(t)}$ by θ (without superscript) and the successor state $\theta^{(t+1)}$ after one step of the algorithm by θ'. Inserting the parametrized potential (5) into the drift inequality (2) yields

$$\begin{aligned} 0 &< B \qquad (6)\\ &\leq \mathbb{E}[V(\theta) - V(\theta') \mid \theta] \\ &= \log(f_\mu(m)) + Q(\theta) - \mathbb{E}[\log(f_\mu(m')) + Q(\theta') \mid \theta] \\ &= \log(f_\mu(m)) + q^T w(\theta) - \mathbb{E}[\log(f_\mu(m')) \mid \theta] - \mathbb{E}[q^T w(\theta') \mid \theta] \\ &= \log(f_\mu(m)) + q^T w(\theta) - \mathbb{E}[\log(f_\mu(m')) \mid \theta] - q^T \mathbb{E}[w(\theta') \mid \theta]. \end{aligned}$$

Computing the expected values requires some form of integration. Denoting $\varphi(x;\theta)$ as the probability density of our distribution $\mathcal{P}(\theta)$ and assuming the expected values exist, the term $\mathbb{E}[\log(f_\mu(m')) \mid \theta]$ is expressed as

$$\mathbb{E}[\log(f_\mu(m')) \mid \theta] = \int_{\mathbb{R}^d} \log(f_\mu(m')) \cdot \varphi(x;\theta)\, dx.$$

The exact formulation of m' depends on the algorithms transition function $\mathcal{F}$. Even for the (1+1)-ES with an isotropic distribution $\mathcal{N}(m, \sigma^2 I)$, closed form solutions of these expected values are not available. Therefore, we rely on numerical integration to approximate these quantities. Furthermore, the integrals are parametrized by the current state θ of the algorithm and have to be evaluated across the whole discretized domain $\theta_1, \dots, \theta_n \in G$.

To clarify the notation we inspect θ' and m' further. Through the transition function (1) we know that $\theta' = \mathcal{F}(\theta, x; f)$. Since the starting state is restricted to the grid, we highlight this dependency of θ' on $\theta_k \in G$ and write $\theta'_k = \mathcal{F}(\theta_k, x; f)$ under slight misuse of notation. It is important to note that the successor state θ'_k is in general not a grid point. Instead, the index indicates that the starting state is fixed to θ_k. The same notation applies to m'_k.

For the sake of generality, the following numerical scheme is assumed. An explicit integration scheme for the (1+1)-ES on the two-dimensional sphere function will be introduced later.

Assumption 1. *For any state $\theta_k \in G$, there exist approximations $\hat{\mathbb{E}}[\log(f_\mu(m'_k)) \mid \theta_k]$ and $\hat{\mathbb{E}}[w(\theta'_k) \mid \theta_k]$ of $\mathbb{E}[\log(f_\mu(m'_k)) \mid \theta_k]$ and $\mathbb{E}[w(\theta'_k \mid \theta_k)]$, respectively, with the property*

$$|\,\mathbb{E}[\log(f_\mu(m'_k)) \mid \theta_k] - \hat{\mathbb{E}}[\log(f_\mu(m'_k)) \mid \theta_k]\,| < \varepsilon, \qquad (7)$$

$$|\,\mathbb{E}[w_i(\theta'_k \mid \theta_k)] - \hat{\mathbb{E}}[w_i(\theta'_k) \mid \theta_k]\,| < \varepsilon, \qquad (8)$$

for some $\varepsilon > 0$.

The assumption highlights a major restriction of the approach: for larger parameter spaces, applying a suitable scheme quickly becomes computationally demanding. Furthermore, our evaluation of the drift inequality (6) will now be restricted to the discretization G. With $k \in \{1, \dots, n\}$, we obtain

$$\begin{aligned} 0 < B &\leq \hat{\mathbb{E}}[V(\theta_k) - V(\theta'_k) \mid \theta_k] \\ &= \log(f_\mu(m_k)) + q^T w(\theta_k) - \hat{\mathbb{E}}[\log(f_\mu(m'_k)) \mid \theta_k] \\ &\quad - q^T \hat{\mathbb{E}}[w(\theta'_k) \mid \theta_k] \\ &= \log(f_\mu(m_k)) - \hat{\mathbb{E}}[\log(f_\mu(m'_k)) \mid \theta_k] \\ &\qquad + q_k - q^T \hat{\mathbb{E}}[w(\theta'_k) \mid \theta_k]. \end{aligned}$$

By $\delta^L_k = \log(f_\mu(m_k)) - \hat{\mathbb{E}}[\log(f_\mu(m'_k)) \mid \theta_k]$, we denote the one-step logarithmic progress of the algorithm, starting from state θ_k. At this point, only the drift bound B and the penalty vector $(q_1, \dots, q_n)$ are unknown. Furthermore, the right hand side of the inequality is a linear function in q. To maximize our pointwise drift, we can tune the penalty vector through a linear program (LP):

$$\max_{B,q} \quad B, \qquad (9)$$

$$\text{s.t.} \quad B \leq \delta^L_k + q_k - q^T \cdot \hat{\mathbb{E}}[w(\theta'_k) \mid \theta_k] \quad \forall\, \theta_k \in G, \qquad (10)$$

The solution of this LP yields an optimized drift lower bounded by B across the parameter grid of states exactly in the sense of the optimal drift defined in the previous section. In fact, Definition 1 and Lemma 1 can be considered an alternative motivation for the construction of the above LP.

The LP solution is not unique because the vector q is well-defined only up to adding the same constant to all components q_k. We therefore demand the additional constraint $\min\{q_k | 1 \leq k \leq n\} = 0$, which is easy to fulfill post-hoc by subtracting the minimal component from a given solution vector q. Analogous to [1], we then obtain $Q(\theta) \geq 0$ by construction.

From a theory perspective we can consider a sequence of grids G such that the grid boundaries tend to infinity and the grid distance tends to zero at the same time. In parallel, we let the approximation error ε tend to zero. Intuitively[2] the resulting sequence of potential functions resulting from the LP will converge to a function of the form $B \cdot V^* + A$. In other words, provided enough compute for solving problems (7), (8) and (9), we can obtain an arbitrarily close approximation of the optimal potential function.

The difference between optimal pointwise drift and optimal runtime has been discussed in [4]. It was made explicit that tuning the hyperparameters of an algorithm so as to achieve optimal pointwise drift does not necessarily result in optimal runtime. In that sense, optimal drift and optimal runtime are not necessarily the same. The discrepancy can be caused by the potential function being a sub-optimal fit for the problem at hand. In our framework one could build a new close-to-optimal potential function for each parameter setting. Then the only gaps that can possibly remain and cause a similar discrepancy are the differences between the actual potential function and expected fitness, which can be made as small as desired with our construction, and an additive term that comes from bounding the amount of overshooting (see Theorem 1).

3.2 Towards Provable Runtime Bounds

The overarching goal of this approach is to provide a computable and provable drift that yields a linear convergence guarantee through Theorem 1. On that way, several problems have to be addressed.

The computed drift underlies an approximation through numerical integration. The stability of the solution of the LP needs to be shown, calculating how this error influences the final drift. Section 4 provides further insights how this can be handled for the (1+1)-ES with 1/5-success rule. The practical applicability is furthermore limited to low-dimensional parameter spaces because the grid size n and the resulting computational demand grows exponentially with the dimension of the parameter space.

Application of the additive drift theorem requires the expected one-step progress to be lower bounded on the whole domain Θ. Yet, the optimization problem only yields drift across a bounded domain G. We therefore need further information on the one-step progress of our potential function in between grid points and its asymptotic behavior outside of the parameter grid.

A possible two-stage approach to handle these difficulties is centered around Lipschitz continuity of the expected one-step progress. While closed-form solutions for the expected values are unavailable, Lipschitz continuity of $\mathbb{E}[\log(f_\mu(m)) \mid \theta]$ and $q^T \mathbb{E}[w(\theta') \mid \theta]$ can be analyzed. Assuming that the drift across the grid points is known up to a small and a-priori controlled uncertainty ε, we can lower bound the drift in between grid points using the Lipschitz property. The bound depends on the grid cell size and on the Lipschitz constant of the drift. It can be made as small as required by choosing a sufficiently dense grid.

[2] A corresponding technically precise statement requires suitable regularity conditions.

Drift outside of the bounded domain can be established in multiple ways. The boundaries of the grid, on which the penalty term is computed, should represent regions where asymptotic effects dominate algorithm dynamics. For the (1+1)-ES, this corresponds to far too small and far too large step-size, or equivalently, success probabilities that are very close to 1/2 or to zero. The choice of a suitable grid generally requires understanding of the algorithm. One approach is then to extend the penalty function linearly and demonstrate drift analytically by leveraging properties of the selected boundary points. Another approach involves explicitly constructing a penalty function tailored to the asymptotic region, guided by both empirical data and theoretical insights. Either way, the design of the discretized domain will be guided by theoretical considerations.

3.3 Solving the Linear Program

Solving the LP (9) is rather easy because all constraints are active. That can be seen as follows. The Lagrangian of the problem is

$$\mathcal{L}(B, q, \lambda) = -B - \sum_{k=1}^{n} \lambda_k \cdot \left(\delta_k^L + q_k - w_k^T q - B \right)$$

with $w_k = \hat{\mathbb{E}}[w(\theta'_k) \mid \theta_k]$, and with dual variables $\lambda_k \geq 0$. The Karush-Kuhn-Tucker (KKT) optimality conditions demand that the derivatives with respect to the primal variables vanish:

$$\frac{\partial}{\partial B} \mathcal{L}(B, q, \lambda) = -1 + \sum_{k=1}^{n} \lambda_k \overset{!}{=} 0 \qquad \Leftrightarrow \quad \sum_{k=1}^{n} \lambda_k = 1$$

$$\frac{\partial}{\partial q_k} \mathcal{L}(B, q, \lambda) = \lambda_k - \sum_{k=1}^{n} \lambda_i (w_k)_i \overset{!}{=} 0 \qquad \Leftrightarrow \quad \lambda = W\lambda$$

Note that the matrix W consisting of the row vectors w_k is a right Markov (or stochastic) matrix if all weights are positive.[3] Even in case of negative weights (caused by extrapolation beyond the grid) the weights in each vector w_k sum to one. Therefore, the vector $(1, \dots, 1)$ is an eigenvector for eigenvalue one. Then the properties $\lambda_k \geq 0$, $\sum_{k=1}^{n} \lambda_k = 1$, and $W\lambda = \lambda$ yield $\lambda_k = 1/n > 0$ for all $k \in \{1, \dots, n\}$, and the KKT complementary condition $\lambda_k \cdot c_k(B, q) = 0$ guarantees that the inequalities (10) are fulfilled with equality.

All inequalities becoming equations means that the LP reduces to a system of linear equations, which has a closed-form solution. Also, the analysis of the numerical error of its solution reduces to the analysis for the corresponding linear system.

4 Example for the (1+1)-ES

This section applies the introduced method to the (1+1)-ES with 1/5-success rule. We briefly introduce the algorithm, its state space and transition function. To fit the setup introduced in the previous section, an interpolation scheme and the resulting expected values are derived. Furthermore, our numerical integration scheme for the two-dimensional sphere function is briefly explained and the numerical stability of the LP for the (1+1)-ES is investigated. Finally, we provide insights into how to deal with the overshooting term $\mathbb{E}[X^{(T)}]$, introduced by Theorem 1.

4.1 The (1+1)-ES with 1/5-success rule

We focus on the (1+1)-ES with one-fifth success rule (see Algorithm 2), which is designed for minimization of a function $f : \mathbb{R}^d \to \mathbb{R}$.

Algorithm 2 (1+1)-ES with 1/5-success rule

```
input $m^{(0)} \in \mathbb{R}^d, \sigma^{(0)} > 0, f : \mathbb{R}^d \to \mathbb{R}$, parameter $\alpha > 0$
for $t = 1, 2, \dots,$ until stopping criterion is met do
    sample $x^{(t)} \sim \mathcal{N}(m^{(t)}, \sigma^{(t)} \cdot \sigma^{(t)} I)$
    if $f(x^{(t)}) \leq f(m^{(t)})$ then
        $m^{(t+1)} \leftarrow x^{(t)}$
        $\sigma^{(t+1)} \leftarrow \sigma^{(t)} \cdot e^{\alpha}$
    else
        $m^{(t+1)} \leftarrow m^{(t)}$
        $\sigma^{(t+1)} \leftarrow \sigma^{(t)} \cdot e^{-\alpha/4}$
```

While it's the simplest evolution strategy, it performs online adaptation of the sampling distribution, an important feature that also the state-of-the-art algorithm, CMA-ES, uses. Given the algorithm template introduced in section 2, the (1+1)-ES fulfills the criteria in the following way. We define the state of the algorithm at iteration t as $\theta^{(t)} = \left(m^{(t)}, \log(\sigma^{(t)}) \right) \in \Theta = \mathbb{R}^d \times \mathbb{R}$. The offspring x is sampled from the isotropic normal distribution $\mathcal{N}(m, \sigma^2 I)$. The algorithm's transition function $\mathcal{F}$ is [16]

$$\begin{aligned} \theta' = \mathcal{F}(\theta, x; f) &= (x, \log(\sigma) + \alpha) \cdot \mathbb{1}\{f(x) \leq f(m)\} \\ &\quad + (m, \log(\sigma) - \alpha/4) \cdot \mathbb{1}\{f(x) > f(m)\}. \end{aligned} \tag{11}$$

The step-size adaptation mechanism was first introduced by Rechenberg and later simplified by Kern et al. [10]. It adapts the step-size to maintain a success probability of roughly 1/5, where the value of α controls the speed and the stability of the adaptation of $\sigma^{(t)}$. We analyze the algorithm on the sphere function $f : \mathbb{R}^d \to \mathbb{R}, x \mapsto ||x||^2$. Due to the computational effort, the numerical integration scheme further limits the analysis to $d = 2$ dimensions. With the definition of the potential function in equation (4), the spatial optimality function was used to measure the logarithmic optimization progress of the algorithm. For the sphere function centered around the origin, $\log(f_\mu(m))$ reduces to $\log(||m||)$, up to an irrelevant additive constant.

We define the normalized step-size $\overline{\sigma}$ as $\overline{\sigma} = \sigma / ||m - x^*||$, where x^* denotes the optimum of f. It accumulates the effect of σ and m into one variable. As our interest lies in the algorithm's behavior dependent on $\log(\sigma)$, the logarithmic normalized step-size $\log(\overline{\sigma}) = \log(\sigma / ||m - x^*||) = \log(\sigma) - \log(||m||)$, with $x^* = (0, \dots, 0)$, will be investigated.

Akimoto et al. [1] highlight the independence of the success probability and one-step progress of the mean m on the sphere function. This simplifies the application of our method considerably, since it reduces the state space to just one variable $\log(\overline{\sigma})$. Without loss of generality we fix our starting state to the first unit vector e_1 and obtain $\log(\overline{\sigma}) = \log(\sigma)$. On the sphere function, the logarithmic norm of the successor point $\log(||m'||)$ equals

$$\begin{aligned} \log(||m'||) &= \log(||x||) \cdot \mathbb{1}\{||x|| \leq 1\} + \log(||e_1||) \cdot \mathbb{1}\{||x|| > 1\} \\ &= \log(||x||) \cdot \mathbb{1}\{||x|| \leq 1\}. \end{aligned} \tag{12}$$

[3] It describes the transition probabilities of the Markov chain on the grid θ_k that arises from using the weight vectors w_k as transition probabilities, instead of as interpolation weights.

Due to $m = e_1$, the one-step logarithmic progress

$$\delta^L = \log(||e_1||) - \log(||m'||) = -\log(||x||) \cdot \mathbb{1}\{||x|| \leq 1\} \qquad (13)$$

is fully controlled by the sample $x \sim \mathcal{N}(e_1, e^{2\log(\sigma)} I)$, and therefore by $\log(\sigma)$.

Before considering the successor state $\log(\overline{\sigma}')$, we shorten the notation by introducing $s = \log(\sigma)$ and $\overline{s} = \log(\overline{\sigma})$. The logarithmic normalized successor state then becomes

$$\begin{aligned}\overline{s}' &= s' - \log(||m'||)\\ &= (s - \alpha/4)\mathbb{1}\{||x|| > 1\} + (s + \alpha)\mathbb{1}\{||x|| \leq 1\}\\ &\quad - \log(||x||)\mathbb{1}\{||x|| \leq 1\}\\ &= \underbrace{(s - \alpha/4)}_{\overline{s}'_f}\mathbb{1}\{||x|| > 1\} + \underbrace{(s + \alpha - \log(||x||))}_{\overline{s}'_s}\mathbb{1}\{||x|| \leq 1\}.\end{aligned}$$

We describe the algorithms dynamics through two scenarios. In the case of an unsuccessful offspring, i.e. $\mathbb{1}\{||x|| > 1\}$, the successor state results from a shift by $\alpha/4$ to the left. For a sample that satisfies $\mathbb{1}\{||x|| \leq 1\}$, the successor state is shifted to the right by α plus the logarithmic improvement $-\log(||x||) > 0$. We denote the arising states as $\overline{s}'_f$ (failure) and $\overline{s}'_s$ (success).

For a general $s \in \mathbb{R}$, the inequality for the one-step change of the potential is

$$B \leq -\log(||x||) \cdot \mathbb{1}\{||x|| \leq 1\} + q^T \cdot w(s) - q^T \cdot \hat{\mathbb{E}}[w(\overline{s}') \mid s]. \quad (14)$$

At this point, all the prerequisites have been introduced and the focus will shift towards the linear interpolation scheme and its expected value.

4.2 Linear interpolation

Up until now, an interpolation scheme was assumed but not explicitly stated, as it depends on the choice of algorithm and objective function. This sections introduces a simple linear scheme with the corresponding parameter grid. It aims at deriving a computable expression for $\mathbb{E}[w_i(\overline{s}') \mid s]$.

We want to analyze the one-step progress of the algorithm on a logarithmic scale. Since the states themselves are logarithmic, i.e., $s = \log(\sigma)$, and these states undergo additive changes as outlined above, a uniform grid is a suitable choice. By G we denote the one-dimensional equidistant parameter grid

$$G = \{s_i \mid s_i = s_1 + (i-1) \cdot \Delta, \quad i = 1, \ldots, n\}, \qquad (15)$$

with grid-size $\Delta > 0$. The linear interpolation scheme on G is defined as

$$w_i(s) = \begin{cases} \dfrac{s - s_i}{\Delta} + 1 & \text{if } s \in [s_i - \Delta, s_i],\\ \dfrac{s_i - s}{\Delta} + 1 & \text{if } s \in [s_i, s_i + \Delta],\\ 0 & \text{otherwise.}\end{cases}$$

For any $s \in [s_1, s_n]$ in the vicinity of s_i, specifically $s \in [s_i - \Delta, s_i + \Delta]$, $w_i(s)$ assigns a value relative to the distance of s to s_i. Therefore, for any $s \in [s_1, s_n]$, at most two weights will be larger than zero for $q^T \cdot w(s) = q_i \cdot w_i(s) + q_{i+1} \cdot w_{i+1}(s)$. This behavior is illustrated in Figure 1. Additionally, an extrapolation strategy is required for handling successor states that fall outside the defined interval. To address this, we linearly extend the two boundary weight functions on either end of the grid. A more detailed explanation will follow at the end of this section.

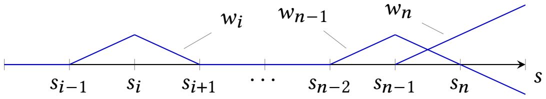

Figure 1: Linear interpolation functions w_i, w_{n-1} and w_n. While w_i represents the standard linear interpolation within the grid, w_{n-1} and w_n are extrapolated to support values of $s > s_n$ when evaluating $q^T w(s)$. Similarly, the first two weights, w_1 and w_2, are extended accordingly to account for $s < s_1$.

As we are interested in the expected value with respect to the logarithmic normalized successor state, we investigate

$$\begin{aligned}E[w_i(\overline{s}') \mid s] &= \mathbb{E}[w_i(\overline{s}'_f \cdot \mathbb{1}\{||x|| > 1\} + \overline{s}'_s \cdot \mathbb{1}\{||x|| \leq 1\}) \mid s]\\ &= \mathbb{E}[w_i(\overline{s}'_f) \cdot \mathbb{1}\{||x|| > 1\} + w_i(\overline{s}'_s) \cdot \mathbb{1}\{||x|| \leq 1\} \mid s]\\ &= w_i(s - \alpha/4)\mathbb{E}[\mathbb{1}\{||x|| > 1\} \mid s]\\ &\qquad + \mathbb{E}[w_i(\overline{s}'_s) \cdot \mathbb{1}\{||x|| \leq 1\} \mid s].\end{aligned}$$

With the success probability $p^{\text{succ}}(s) = \Pr(||x|| \leq 1)$, the expected value is

$$E[w_i(\overline{s}') \mid s] = w_i(s - \alpha/4)(1 - p^{\text{succ}}(s)) + E[w_i(\overline{s}'_s) \cdot \mathbb{1}\{||x|| \leq 1\} \mid s]. \qquad (16)$$

The first part simplifies to the linear interpolation shifted by $\alpha/4$ to the left and the probability of an unsuccessful step happening. Let $\varphi(x; s)$ denote the density of $\mathcal{N}(e_1, e^{2s} I)$. Focusing on the latter term

$$\begin{aligned}&\mathbb{E}[w_i(\overline{s}'_s) \cdot \mathbb{1}\{||x|| \leq 1\} \mid s]\\ &= \int_{\mathbb{R}^d} w_i(\overline{s}'_s) \cdot \mathbb{1}\{||x|| \leq 1\} \cdot \varphi(x; s)\, dx\\ &= \int_{B_d(0,1)} \left(\left(\frac{\overline{s}'_s - s_i}{\Delta} + 1 \right) \mathbb{1}\{\overline{s}'_s \in [s_i - \Delta, s_i]\} \right.\\ &\quad \left. + \left(\frac{s_i - \overline{s}'_s}{\Delta} + 1 \right) \mathbb{1}\{\overline{s}'_s \in [s_i, s_i + \Delta]\} \right) \varphi(x; s)\, dx\end{aligned}$$

we rewrite the indicators as

$$\begin{aligned}&\mathbb{1}\{s_i - \Delta \leq s + \alpha - \log(||x||) \leq s_i\}\\ &= \mathbb{1}\{\exp(s + \alpha - s_i) \leq ||x|| \leq \exp(s + \alpha - s_i + \Delta)\}\end{aligned}$$

and

$$\begin{aligned}&\mathbb{1}\{s_i \leq s + \alpha - \log(||x||) \leq s_i + \Delta\}\\ &= \mathbb{1}\{\exp(s + \alpha - s_i - \Delta) \leq ||x|| \leq \exp(s + \alpha - s_i)\}.\end{aligned}$$

Therefore, both indicators impose further restrictions on the integration volume. To represent the spherical shells formed by the indicators, we introduce the balls

$$\begin{aligned}B_i^{\text{outer}} &= B(0, \exp(s + \alpha - s_i + \Delta)) \setminus B(0, \exp(s + \alpha - s_i)),\\ B_i^{\text{inner}} &= B(0, \exp(s + \alpha - s_i)) \setminus B(0, \exp(s + \alpha - s_i - \Delta)).\end{aligned}$$

These spherical shells represent the following scenario: starting from $s \in \mathbb{R}$, in case of a successful step, it holds $\overline{s}' \in [s_i - \Delta, s_i]$ or

$\bar{s}' \in [s_i, s_i + \Delta]$ if $x \in B_i^{\text{outer}}$ or $x \in B_i^{\text{inner}}$, respectively.
We separate the terms and reduce the volume to

$$\begin{aligned}
&\mathbb{E}[w_i(\bar{s}'_s) \cdot \mathbb{1}\{||x|| \le 1\} \mid s] \\
&= \int_{B_i^{\text{outer}}} \left(\frac{\bar{s}'_s - s_i}{\Delta} + 1 \right) \varphi(x; s)\, dx + \int_{B_i^{\text{inner}}} \left(\frac{s_i - \bar{s}'_s}{\Delta} + 1 \right) \varphi(x; s)\, dx.
\end{aligned} \tag{17}$$

Further rearranging the first term yields

$$\begin{aligned}
&\int_{B_i^{\text{outer}}} \left(\frac{\bar{s}'_s - s_i}{\Delta} + 1 \right) \varphi(x; s)\, dx \\
&= \int_{B_i^{\text{outer}}} \left(\frac{s + \alpha - \log(||x||) - s_i}{\Delta} + 1 \right) \varphi(x; s)\, dx \\
&= \left(\frac{s + \alpha - s_i}{\Delta} + 1 \right) \int_{B_i^{\text{outer}}} \varphi(x; s)\, dx - \int_{B_i^{\text{outer}}} \frac{\log(||x||)}{\Delta} \varphi(x; s)\, dx.
\end{aligned}$$

Essentially, the required numerical integration reduces to a probability of success and a logarithmic progress over a spherical shell. Similar rearrangement can be applied to the second term of (17), leading to a computable expression for the expected value of a single weight vector in the success case:

$$\begin{aligned}
&\mathbb{E}[w_i(\bar{s}'_s) \cdot \mathbb{1}\{||x|| \le 1\} \mid s] \\
&= \left(\frac{s + \alpha - s_i}{\Delta} + 1 \right) \int_{B_i^{\text{outer}}} \varphi(x; s)\, dx - \int_{B_i^{\text{outer}}} \frac{\log(||x||)}{\Delta} \varphi(x; s)\, dx \\
&+ \left(\frac{s_i - s - \alpha}{\Delta} + 1 \right) \int_{B_i^{\text{inner}}} \varphi(x; s)\, dx + \int_{B_i^{\text{inner}}} \frac{\log(||x||)}{\Delta} \varphi(x; s)\, dx.
\end{aligned} \tag{18}$$

The result highlights the mechanism behind the expected value of the linear interpolation scheme in the success case. Starting from some state $s \in G$, a single weight $\mathbb{E}[w_i(\bar{s}'_s) \cdot \mathbb{1}\{||x|| \le 1\} \mid s]$ becomes active, i.e. $E[w_i] > 0$, if the success probability to land in $[s_i - \Delta, s_i + \Delta]$ is positive. This is the case if $s + \alpha < s_i + \Delta$.

To account for successor states that land outside of the bounded region covered by the grid, we establish an extrapolation approach. These situations arise if

$$s - \alpha/4 < s_1 \text{ or } s + \alpha - E[\log(||x||)\mathbb{1}\{||x|| \le 1\} \mid s] > s_n.$$

To handle such cases, we linearly extrapolate the first and last pair of weight functions, w_1, w_2 and w_{n-1}, w_n, as displayed in Figure 1. With respect to $q^T \cdot \mathbb{E}[w(\bar{s}' \mid s)]$, this extends the weight functions and all quantities based thereon from the bounded interval $[s_1, s_n]$ to the unbounded state space $\mathbb{R}$.

For w_n and w_{n-1} we obtain

$$w_n(s) = \begin{cases} \dfrac{s - s_n}{\Delta} + 1 & \text{if } s > s_{n-1}, \\ 0 \quad \text{else}, \end{cases}$$

and

$$w_{n-1}(s) = \begin{cases} \dfrac{s - s_{n-1}}{\Delta} + 1 & \text{if } s \in [s_{n-2}, s_{n-1}], \\ \dfrac{s_{n-1} - s}{\Delta} + 1 & \text{if } s > s_{n-1}, \\ 0 \quad \text{else}. \end{cases}$$

The pair of weights at the left side of the grid, w_1 and w_2, are modified accordingly. As the left bound of the grid can only be crossed with an unsuccessful step, $w_{1/2}(s - \alpha/4)$ can be evaluated directly. Yet, the arising expected values $\mathbb{E}[w_{n/n-1}(\bar{s}'_s)\mathbb{1}\{||x|| \le 1\}]$ need further consideration:

$$\begin{aligned}
&\mathbb{E}[w_n(\bar{s}'_s)\mathbb{1}\{||x|| \le 1\}] \\
&= \int_{B_d(0,1)} \left(\frac{\bar{s}'_s - s_n}{\Delta} \right) \mathbb{1}\{\bar{s}'_s > s_{n-1}\} \varphi(x; s)\, dx \\
&= \int_{||x|| \le e^{s+\alpha-s_{n-1}}} \left(\frac{s + \alpha - \log(||x||) - s_n}{\Delta} + 1 \right) \varphi(x; s)\, dx \\
&= \left(\frac{s + \alpha - s_n}{\Delta} + 1 \right) \int_{B(0,\exp(s+\alpha-s_{n-1}))} \varphi(x; s)\, dx \\
&\quad - \int_{B(0,\exp(s+\alpha-s_{n-1}))} \frac{\log(||x||)}{\Delta} \varphi(x; s)\, dx.
\end{aligned}$$

While other expected weight functions are defined through integration over a spherical shell, the boundary weights account for the non-zero probability that a successor state lands arbitrarily far from s_n by integrating over a ball centered around the origin.

Finally, w_{n-1} yields

$$\begin{aligned}
&\mathbb{E}[w_{n-1}(\bar{s}'_s)\mathbb{1}\{||x|| \le 1\}] \\
&= \left(\frac{s + \alpha - s_{n-1}}{\Delta} + 1 \right) \int_{B_{n-1}^{\text{outer}}} \varphi(x; s)\, dx \\
&\quad - \int_{B_{n-1}^{\text{outer}}} \frac{\log(||x||)}{\Delta} \varphi(x; s)\, dx \\
&\quad + \left(\frac{s_n - s - \alpha}{\Delta} + 1 \right) \int_{B(0,\exp(s+\alpha-s_{n-1}))} \varphi(x; s)\, dx \\
&\quad + \int_{B(0,\exp(s+\alpha-s_{n-1}))} \frac{\log(||x||)}{\Delta} \varphi(x; s)\, dx.
\end{aligned}$$

4.3 Numerical integration

For analysis of the (1+1)-ES, we need approximations of various expected values, in particular of the weights w_i of successor states and the progress $\log(||x||)$, for each grid point s_k. Both functions of the offspring sample x are piecewise smooth, the Gaussian density $\varphi(x; s)$ of x is smooth, and the boundaries between the smooth parts are spheres. Most prominently, the unit sphere separates the bounded success region from the unbounded region of unsuccessful steps. To this end, we also estimate the success probability $P(||x|| \le 1)$, which is the expected value of the indicator function of the unit sphere. The special property of the (1+1)-ES always performing the exact same adaptation for all unsuccessful steps allows us to restrict the integration area of all quantities to the unit ball.

We obtain approximations of the above expectations by means of numerical integration with the midpoint rule applied to rectangles. For each rectangle we compute (term-specific) upper bounds on the deviation of the integrand from its first order Taylor approximation in the midpoint, directly yielding an error bound. Consider the rectangle $R = [l_1, u_1] \times [l_2, u_2]$ with midpoint $c = \frac{1}{2}(l_1 + u_1, l_2 + u_2)$, as well as an integrand $h \in C^2(R)$. Let $h_1(x) = h(c) + (x - c)^T \nabla h(c)$ denote the first order Taylor approximation of h around c, and let b denote an upper bound on the deviation $|h(x) - h_1(x)|$ (usually obtained by bounding second derivatives of h). Then it holds

$$\left| \int_R h(x) dx - (u_1 - l_1)(u_2 - l_2) h(c) \right| \le b(u_1 - l_1)(u_2 - l_2)$$

because the first order term in h_1 cancels out when integrated over the rectangle. If the bound is too large then we subdivide the rectangle with the largest error contribution until the overall error bound falls below the pre-defined threshold $\varepsilon > 0$.

Three cases need special attention.

- The rectangles must be capped at the sphere-shaped boundary. In this case the linear term does not cancel out and must hence be computed explicitly. The problem of integrating a linear function over the intersection of a rectangle and a sphere has a closed form solution (with a number of case distinctions depending on where the circle intersects the rectangle), which we exploit.
- The logarithmic progress $\int_{\|x\|\leq 1} \log(\|x\|)\varphi(x;s)dx$ has a pole at the origin, however, with finite integral. We therefore cut out a small ball around the origin, which is chosen so that the Gaussian density $\varphi(x;s)$ is well approximated by a linear function. Then the integration over the small ball can be performed in closed form with a similar error bound as above.
- Due to the linear interpolation scheme, the weight function is non-smooth at its maximum. We therefore integrate the two smooth parts independently. Each integration region is a ring, i.e., the set difference of two balls centered at the origin.

For the last two cases a similar consideration as for the unit sphere applies to the two spherical boundaries of the integration region, which are handled with the same techniques.

By subdividing a rectangle along both axes into four smaller rectangles the computational effort grows by a factor of four. The second derivatives of h are bounded. Therefore, halving the length scales of $x - c$ has a quadratic effect on

$$\left|(x-c)^T \nabla^2 h(c)(x-c)\right|$$

and on its bound b. This also holds for the boundary rectangles: while the linear term does not vanish, its closed form integration does not incur an increased numerical error. We conclude that the computational effort of each numerical integration is $\Theta(1/\varepsilon)$.

The above consideration focuses on the approximation error ε, which we set to 10^{-6} in our experiments. That precision is far more than sufficient for demonstrating drift. For that setting the approximation error dominates rounding errors caused by floating point arithmetics. However, when increasing the precision further, round-off errors must also be taken into account.

4.4 Linear Program with Approximate Weights

Following up on the considerations outlined in section 3.3, we start from

$$B = \delta_k^L + q_k - q^T \cdot \hat{\mathbb{E}}[w(\bar{s}') \mid s].$$

With $\overline{w}_i = \mathbb{E}[w_i(\bar{s}') \mid s]$, the linear system is derived as

$$\begin{pmatrix}\delta_1^L\\ \vdots\\ \delta_n^L\end{pmatrix} + \begin{pmatrix}\overline{w}_1(s_1)-1 & \overline{w}_2(s_1) & \cdots & \overline{w}_n(s_1)\\ & \vdots & \ddots & \\ \overline{w}_1(s_n) & \overline{w}_2(s_n) & \cdots & \overline{w}_n(s_n)-1\end{pmatrix}\begin{pmatrix}q_1\\ \vdots\\ q_n\end{pmatrix} = \begin{pmatrix}B\\ \vdots\\ B\end{pmatrix}.$$

We (temporarily) introduce the additional constraint $\sum_{k=1}^n q_k = 0$ in order to make the solution unique and obtain the following system of equations:

$$\begin{pmatrix}\overline{w}_1(s_1)-1 & \overline{w}_2(s_1) & \cdots & \overline{w}_n(s_1) & 1\\ & \vdots & \ddots & & \\ \overline{w}_1(s_n) & \overline{w}_2(s_n) & \cdots & \overline{w}_n(s_n)-1 & 1\\ 1 & 1 & \cdots & 1 & 0\end{pmatrix}\begin{pmatrix}q_1\\ \vdots\\ q_n\\ B\end{pmatrix} = \begin{pmatrix}\delta_1^L\\ \vdots\\ \delta_n^L\\ 0\end{pmatrix}.$$

At this point, we rely on numerical integration to obtain approximations for $w_i(s_j)$ and δ_i^L with $i, j \in \{1, \dots, n\}$. The process introduces a componentwise error into the resulting system of linear equations. Let $Wq = \delta^L$ represent the exact system, and let $\hat{W}\hat{q} = \hat{\delta}^L$ denote the perturbed system arising from the numerical approximation. With a slight abuse of notation, we also include the pointwise drift B in q, and extend W accordingly. The size n is adjusted to reflect these additions. By construction of the integration method it holds $|W_{ij} - \hat{W}_{ij}| < \varepsilon$ and $|\delta_i^L - \hat{\delta}_i^L| < \varepsilon$ for all $i, j \in \{1, \dots, n\}$. We are interested in the relative error of our solution $||q - \hat{q}||/||q||$, specifically in terms of its Euclidean norm $||\cdot||_2$. As a starting point, we rely on a classical result from perturbation theory [3, Theorem 7.29]:

Theorem 2. *Assume $Ax = b$ and $\hat{A}\hat{x} = \hat{b}$. If A is nonsingular and $||\hat{A} - A|| < 1/||A^{-1}||$ then it holds*

$$\frac{||\hat{x} - x||}{||x||} \leq \frac{cond(A)}{1 - cond(A)\dfrac{||\hat{A} - A||}{||A||}}\left(\frac{||\hat{b} - b||}{||b||} + \frac{||\hat{A} - A||}{||A||}\right).$$

Since we only have access to $\hat{W}$, $\hat{\delta}^L$ and the componentwise pertubation ε, we upper-bound our relative error further. Exploiting the properties of the Euclidian norm yields $||\hat{W} - W||_2 \leq n \cdot \varepsilon$ and $||W||_2 \geq ||\hat{W}||_2 - n \cdot \varepsilon$. For the vector containing the logarithmic progress, we obtain $||\hat{\delta}^L - \delta^L||_2 \leq \sqrt{n} \cdot \varepsilon$ and $||\delta^L||_2 \geq ||\hat{\delta}^L||_2 - \sqrt{n} \cdot \varepsilon$. Lastly, a bound on the condition number $cond(W) = ||W||_2 \cdot ||W^{-1}||_2$ is derived. Let $\lambda_i(W)$ denote the i-th largest eigenvalue of W, by Weyl's inequality for singular values, it holds $|\lambda_i(W) - \lambda_i(\hat{W})| \leq ||\hat{W} - W||_2$ and for the smallest eigenvalue $\lambda_{\min}(W) \geq \lambda_{\min}(\hat{W}) - ||\hat{W} - W||_2 \geq \lambda_{\min}(\hat{W}) - n \cdot \varepsilon$. This allows us to upper-bound the spectral norm of W^{-1} as

$$||W^{-1}||_2 = \frac{1}{\lambda_{\min}(W)} \leq \frac{1}{\lambda_{\min}(\hat{W}) - n \cdot \varepsilon} \quad \text{if } \lambda_{\min}(\hat{W}) > n \cdot \varepsilon.$$

To be able to verify the prerequisites of the theorem, the same bound is applied as $||\hat{W} - W||_2 < \lambda_{\min}(\hat{W}) - n \cdot \varepsilon \leq 1/||W^{-1}||_2 = \lambda_{\min}(W)$. To show that W is nonsingular, we again rely on Weyl's inequality, by which we know that $|\lambda_i(\hat{W}) - \lambda_i(W)| \leq n \cdot \varepsilon$. Therefore, if $\lambda_i(\hat{W}) > n \cdot \varepsilon$ holds for all $i \in \{1, \dots, n\}$, then W is nonsingular.

Assuming these conditions can be fulfilled and $\lambda_{\min}(\hat{W}) > n \cdot \varepsilon$, we can bound the relative error by

$$\frac{||\hat{q} - q||_2}{||q||_2} \leq \frac{cond(W^*)}{1 - cond(W^*)\dfrac{n \cdot \varepsilon}{||\hat{W}||_2 - n \cdot \varepsilon}} \left(\frac{n \cdot \varepsilon}{||\hat{W}||_2 - n \cdot \varepsilon} + \frac{\sqrt{n} \cdot \varepsilon}{||\hat{\delta}^L||_2 - \sqrt{n} \cdot \varepsilon}\right), \tag{19}$$

with $cond(W^*)$ denoting our bound

$$cond(W^*) = (||\hat{W}||_2 + n \cdot \varepsilon) \cdot \frac{1}{\lambda_{\min}(\hat{W}) - n \cdot \varepsilon}$$

of the condition number.

4.5 Bounding the Amount of Overshooting

In order to obtain a useful runtime bound from Theorem 1 we need to bound the expected overshooting $\beta - \mathbb{E}[X^{(T)}]$. The term describes by how much the last step of the algorithm overshoots the target potential threshold of β. Such excess progress is superfluous for the goal of reaching β, and therefore large overshooting, potentially coming from rare events, can be detrimental for the expected runtime (and its bound).

The amount of overshooting is bounded by the change of the potential $V(\theta)$ in a single step, conditioned on the event that the improvement drives the potential beyond the stopping threshold β. Let $\gamma = V(\theta) - \beta > 0$ denote the gap between current potential and stopping threshold. The following bound is independent of β:

$$\beta - \mathbb{E}[X^{(T)}] \le \sup_{\theta,\gamma>0} \left(V(\theta) - \mathbb{E}\left[V(\theta') \,\middle|\, V(\theta) - V(\theta') > \gamma\right]\right) \quad (20)$$

This term is no easier to control than the drift itself, but we may be satisfied with a less tight bound because it has only an additive effect on the runtime bound, in contrast to the multiplicative effect of the drift.

Plugging the transition function (11) into the potential (4) we see that an unsuccessful offspring can cause a potential difference of at most $m_q \cdot \alpha/4$, where

$$m_q = \max\left\{ \frac{|q_k - q_{k-1}|}{\Delta} \,\middle|\, 1 < k \le n \right\}$$

denotes the maximal slope of Q. These events enter the conditional expectation only for small gaps γ, and more importantly, the amount of overshooting is well controlled. The potential improvement of a successful offspring can be bounded by $(1 + m_q)\big(\log(||m||) - \log(||x||)\big) + m_q\alpha$. It needs closer attention because $\log(||x||)$ is unbounded, and we therefore have to control its effect on the conditional expectation (20).

At this point we can return to the normalized setting with $m = e_1$ and parameter $\bar{s}$. For each fixed tuple of $\gamma > 0$ and $\bar{s}$ there exists a radius $0 < r \le 1$ so that successful offspring samples $x \in B(0, r)$ cause a progress of at least γ. They overshoot by at most $(1 + m_q) \cdot (\log(r) - \log(||x||)) + m_q\alpha$. The expected overshooting of successful steps is therefore bounded by

$$m_q\alpha + (1 + m_q) \int_{B(0,r)} \Big(\log(r) - \log(||x||)\Big)\varphi(x; \bar{s})dx. \quad (21)$$

We note that it holds $\int_{B(0,r)} \log(r) - \log(||x||)dx = \frac{\pi}{2}r^2$. Taking the worst case $r = 1$, it remains to control the (Gaussian) density

$$\varphi(x; \bar{s}) = \frac{1}{2\pi \exp(2\bar{s})} \exp\left(-\frac{1}{2}||x - m||^2 \exp(-2\bar{s})\right).$$

For $\bar{s} \to -\infty$ the density is unbounded in the vicinity of $x = m$, but with a bounded effect on the integrand, which vanishes at that point. We exploit that fact in the following, which requires some technicalities.

For $\bar{s} \ge 0$ we obtain the bound $\varphi(x; \bar{s}) \le \frac{1}{2\pi}$. For $\bar{s} < 0$ we split the integration region into an inner ball $B(0, \rho)$ and an outer ring $R = B(0, 1) \setminus B(0, \rho)$ with $\rho = 1 - 2\sqrt{-\bar{s}}\exp(\bar{s})$. On $B(0, \rho)$ the density attains its maximum at $\rho \cdot e_1$, where it takes the value $\frac{1}{2\pi}$ by construction of ρ. On the ring R the integrand $-\log(||x||)$ is upper bounded by $-\log(\rho) = -\log(1 - 2\sqrt{-\bar{s}}\exp(\bar{s}))$. The function $2\exp(\bar{s})\sqrt{-\bar{s}}$ attains its maximum at $\bar{s} = -\frac{1}{2}$ and we obtain $-\log(\rho) \le -\log(1 - \sqrt{2/e})$, which computes to a value slightly less than two. Integrating that value over a part of a probability density with probability mass bounded by $p^{\text{succ}}(\bar{s}) \le 1/2$ yields a bound of one.

Combining all results, the expected overshooting is bounded by

$$\beta - \mathbb{E}[X^{(T)}] \le m_q\alpha + (1 + m_q)\left(\frac{\pi}{2} + \frac{1}{4}\right). \quad (22)$$

5 Experimental results

This section demonstrates the application of the framework to the (1+1)-ES with 1/5-success rule on the two-dimensional sphere function. We discuss different grid setups and their accuracy, highlight the expected value of the linear interpolation scheme and compare empirical runtimes to those obtained from our approximated drift.

5.1 Setup

The previous section dealt with the numerical difficulties arising from the method developed in Section 2. Yet, variables central to the experimental setup, the grid boundaries s_1 and s_n, along with the grid-size Δ, were mostly untouched. Section 3 highlighted their importance from a theoretical point of view. Here, we introduce various grids and conduct an experimental study on the stability of the simulation and the corresponding drift dependent on the tuple (s_1, s_n, Δ).

We introduce our reference grid with the following parameters: $s_1 = -9$, $s_n = 3$ and $\Delta = 0.1$. We set α to 1/3 in all experiments. Based on our simulations, this grid captures all asymptotic effects with significant buffer, and thus serves as the baseline for assessing sensitivity to reductions in the grid's range and density.

The analysis will focus on several parameters, the constant drift obtained across the grid, the condition number of the resulting (perturbed) matrix of the linear system with a respective error bound, and the general shape of the resulting penalty function.

5.2 Results

We present our main experimental findings through Figure 2 and Table 1. Starting with Figure 2, the penalty term $q^T w(s)$ is displayed for various grid configurations, compared to the reference grid. The setups are deliberately chosen to include cases where stable simulations may not be expected, either because not all effects are fully captured or because the grid is particularly coarse.

Focusing on the baseline, represented by the dotted line, we note that its behaviour aligns with that described by other explicit methods, such as in [1]. The curve can be divided into three regions: two asymptotic regions, where the penalty function grows linearly, and a transition phase in which the step-size is well adapted. We further note the difference in slope for both asymptotic cases, indicating that the step-size adaptation is faster when the step-size is too large, hence the larger slope. Compared to explicit penalty terms of the form $\max\{0, f(\bar{\sigma}), f(1/\bar{\sigma})\}$, for example in [2], the transition zone exhibits a much smoother shape.

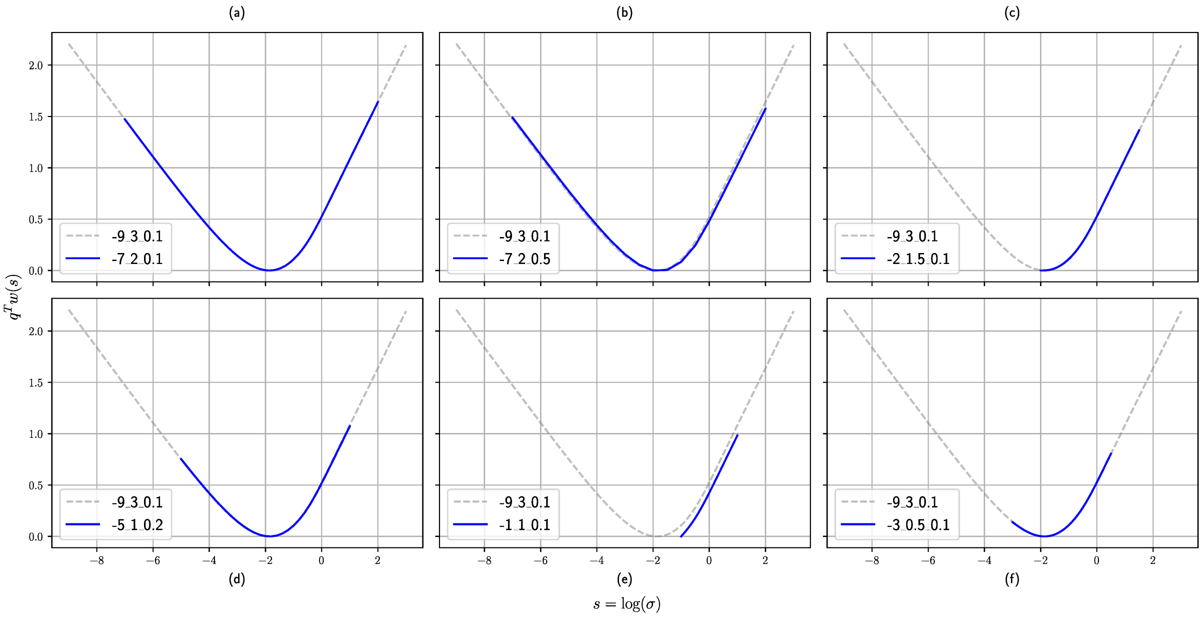

Figure 2: The penalty term $q^T w(s)$ displayed across multiple parameter combinations in comparison to the reference grid with $s_1 = -9$, $s_n = 3$ and $\Delta = 0.1$. The legend depicts the values in the same order: s_1, s_n and Δ.

The optimal asymptotic left and right slopes

$$\lim_{s \to -\infty} Q'(s) \qquad \text{and} \qquad \lim_{s \to +\infty} Q'(s)$$

can be computed explicitly. On the left, the success rate tends to 1/2, while the impact of a success tends to zero. With half of the steps changing the potential by $+\alpha$ and the other half by $-\alpha/4$ we obtain an expected change of s by $\frac{3}{8}\alpha$. The right asymptotic case is governed by the success rate tending to zero, and hence by a near-deterministic change of s by $-\alpha/4 = -\frac{2}{8}\alpha$. In order to achieve a constant drift, the right asymptotic slope must there be $-3/2$ times the left asymptotic slope. For reasonably large grids that capture the asymptotic behaviour, this prediction is conformed to high precision by our experimental results.

Shifting the view to smaller discretized spaces, we observe that almost all plots, with the exception of (e), closely follow the baseline. This is interesting, especially for grids that do not capture the entire spectrum of relevant behavior. The smallest grid, (e), yields a slightly shifted penalty term due to normalization of its minimum value to zero, yet it preserves the overall slope. In terms of coarse grids, only small deviations can be seen for (b), whereas even (d) still follows the baseline.

Table 1 supports the observed findings. The relative error is computed according to equation (19) with $\varepsilon = 10^{-6}$, for which all requirements are fulfilled a posteriori. All parameter combinations yield a similar drift. Comparing (a) and (b), a less dense grid reduces the drift by a small percentage while reducing conditioning of $\hat{W}$ due to the lower number of grid points. A general trend emerges in which a larger and denser grid results in slightly more drift and a higher relative error. Interestingly, (e) yields more drift than (f) despite missing much of the transition phase. Our experimental studies highlight the significance of accurately capturing the right asymptotic region, which may explain the discrepancy between (e) and (f). Furthermore, a grid that only spans the left asymptotic region up to the transition phase yields a matrix $\hat{W}$ that does not fulfill the requirements for a relative error analysis due to its high condition number, and is therefore insufficient for our analysis.

In conclusion, we emphasize the robustness of our approach, even when boundary values are poorly chosen or the grid is relatively coarse. This is particularly important when transitioning to algorithms with larger state spaces, where numerical limitations become a significant constraint.

Plot	s_1	s_n	Δ	Drift	Rel. Error	cond($\hat{W}$)
Ref.	-9	3	0.1	0.046178	0.016	611.37
(a)	-7	2	0.1	0.046174	0.01	393.87
(b)	-7	2	0.5	0.045744	0.003	172.05
(c)	-2	-1.5	0.1	0.046154	0.001	65.28
(d)	-5	1	0.2	0.046	0.0044	257.95
(e)	-1	1	0.1	0.046044	0.0005	45.06
(f)	-3	0.5	0.1	0.045881	0.0053	375.45

Table 1: Grid parameters s_1, s_n and Δ and corresponding results: pointwise drift across the domain, relative error of the solution according to (19) and condition number of the resulting pertubed matrix $\hat{W}$.

5.3 Linear interpolation

Given the observed stability in the experiments, we inspect the underlying expected weight vector $\mathbb{E}[(w_1, \ldots, w_n)^T \mid s]$ of the grid with parameters $s_1 = -7$, $s_n = 2$ and $\Delta = 0.1$, corresponding to Table 1 (a).

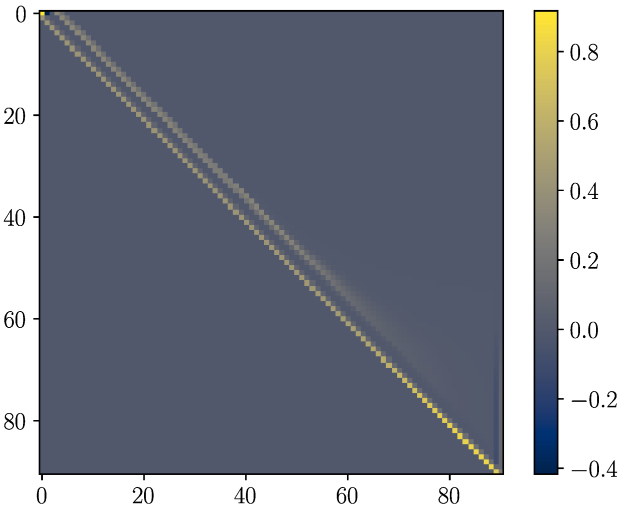

Figure 3: Expected weight vector $\mathbb{E}[(w_1, \ldots, w_n)^T \mid s]$ visualized across the discretized domain with parameters $s_1 = -7$, $s_n = 2$ and $\Delta = 0.1$. Both axes correspond to grid point indices: the x-axis represents the index i of $\mathbb{E}[w_i \mid s_j]$, while the y-axis increases s_j from top to bottom.

It is visualized in Figure 3. Both axes represent index positions within the grid: the x-axis indicates the index i of $\mathbb{E}[w_i(\bar{s}') \mid s_j]$, while the y-axis increases s_j from top to bottom. The diagonal represents the starting state s_j. The difference in the adaptation mechanism of the logarithmic normalized successor state $\bar{s}'$ is clearly visible. Left to the diagonal, at most two weights are active. They interpolate the successor state resulting from unsuccessful offspring. On the right, we observe a different pattern: as s increases, the interpolation spreads across more than two weights, leading to a broader distribution over the successor states. This behavior stems from $\bar{s}'_s = s + \alpha - \log(||x||)$ taking more diverse values for large step size. Additionally, the influence of the success probability is displayed through the increasing probability mass allocated to the interpolation of the failure state.

5.4 Runtime comparison

In this section, we compare the expected runtime resulting from our pointwise drift through the drift theorem with empirical results for the (1+1)-ES with 1/5-success rule on the two-dimensional sphere function. It is important to note that our analysis only yields a uniform drift across all grid points of the chosen bounded domain. At these points, we know it to a certain precision, described by our relative error. However, we have no information about the drift in between grid points or beyond the boundaries of the grid. As such, the derived drift remains an approximation and does not capture the behavior across the entire state space.

As a starting point, Figure 4 yields insights about a typical run of the (1+1)-ES on the two-dimensional sphere function. The figure is based on [2] and while their potential function was explicitly defined for the whole domain, our method relies heavily on extrapolation due to the initial normalized step-sizes being far outside the covered grid range. Nonetheless, the Figure captures the expected behavior, for which the normalized step-size is first adapted into a regime where optimization progress can be made. Then, it remains approximately constant while the algorithm converges linearly.

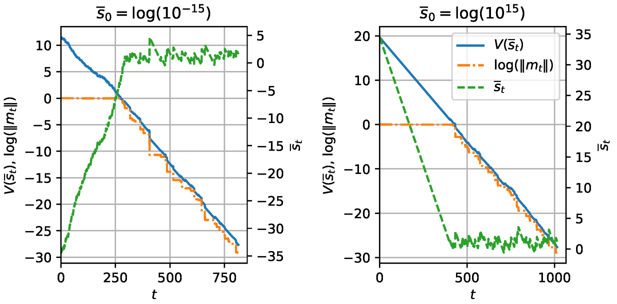

Figure 4: Dynamics of the potential function $V(\bar{s}_t)$ and logarithmic distance to the optimum $\log(||m_t||)$ over function evaluations t for the (1+1)-ES with 1/5-success rule on the two-dimensional sphere function. The potential utilizes a penalty term computed using grid parameters $s_1 = -7$, $s_n = 2$ and $\Delta = 0.1$.

We conclude by conducting an empirical comparison between the averaged runtime of the algorithm and its overshooting with the runtime we obtain through our computed drift. We note again that the drift, as of right now, only serves as an approximation without rigour, as parts of the domain are not covered yet. An actual bound on the drift will likely be smaller to accommodate the remaining obstacles. Still, assuming our drift would be produced across the entire domain, we are interested in the accuracy of our bound. Table 2 provides a comparison between the runtime and overshooting values. The setup consists of a target value $\beta = \log(10^{-12})$ for all simulations. The penalty vector and corresponding drift is computed using a grid with parameters $s_1 = -7$, $s_n = 2$ and $\Delta = 0.1$, with the resulting drift shown in Table 1 (a). The initial mean of the distribution is fixed as $m = e_1$, while the step-size is varied.

A noticeable difference is observed between the empirical overshooting, displayed through its 10th, 50th and 90th percentiles, and the one obtained from inequality (22), which partially explains the difference between the averaged evaluations and our bound from the drift theorem. Nonetheless, our estimates offer a reasonably accurate characterization of the observed behavior.

6 Conclusion

We have presented a novel method for constructing a potential function for drift analysis of evolution strategies. It is suitable for describing the dynamics of a linearly convergent algorithm.

The central technique is to discretize the state space to a grid, and to compute the algorithm dynamics by means of numerical integration. Based on the resulting data an optimized penalty term is constructed for the strategy parameters. The term is extended

$\bar{s}_0$	Mean evals.	$\mathbb{E}[T] \lessapprox$	Est. oversh.	$\beta - \mathbb{E}[X^{(T)}]$
$\log(10^{15})$	1026 ± 37	1091	0.005/0.026/0.122	3.016
$\log(10^{-15})$	851 ± 45	914	0.005/0.026/0.113	3.016
$\log(10^{-1})$	600 ± 36	665	0.005/0.028/0.152	3.016

Table 2: Comparison of empirical runtime and overshooting values with our bounds across three different initial setups. The initial step-size is varied with $m_0 = e_1$. The averaged evaluations are presented with the respective standard deviations, whereas the estimated overshooting is shown through its percentiles 10%/50%/90%. The target β is set to $\log(10^{-12})$ in all experiments.

to the complete state space by means of inter- and extrapolation. By construction, it achieves constant drift on the grid. Constant drift is desirable because an additive drift theorem then provides a perfectly tight runtime bound across all runtime bounds achievable by drift.

We demonstrate the feasibility and the utility of the approach on the most basic evolution strategy, the (1+1)-ES. It provides an ideal testbed for out method because it is extremely well analyzed. In fact, drift was established before by much simpler means. We show that our approach provides an extremely accurate description of the algorithm dynamics, which matches our theoretical understanding and which closely resembles its empirical behavior.

Due to the computer-aided nature of our approach, several limitations arise. In particular, we are unable to derive explicit mathematical expressions for the dependencies on algorithmic parameters, most notably, the problem dimension. Further, the computational complexity restricts the method to low dimensional cases.

While the method already provides useful potential functions, it is not yet fully developed into a proof technique. In future work we will analyze asymptotic effects outside of the grid, and we plan to establish drift in between grid points by means of Lipschitz continuity of the drift. Completing these two steps would enable a fully stringent computer-aided drift-based analysis that yields a tight bound on the convergence rate.

Furthermore, we plan to tackle the analysis of an evolution strategy with covariance matrix adaptation. To the best of our knowledge there is no principled method for constructing a potential function for a variable metric ES suitable for drift analysis. Therefore we believe that our constructive method has the potential to make a significant contribution to the analysis of this important class of optimization algorithms.

References

[1] Youhei Akimoto, Anne Auger, and Tobias Glasmachers. 2018. 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.* 801–808.

[2] Youhei Akimoto, Anne Auger, Tobias Glasmachers, and Daiki Morinaga. 2022. Global linear convergence of evolution strategies on more than smooth strongly convex functions. *SIAM Journal on Optimization* 32, 2 (2022), 1402–1429.

[3] Richard L. Burden, J. Douglas Faires, and Annette M. Burden. 2015. *Numerical Analysis* (9 ed.). Cengage Learning, Boston, MA.

[4] Nathan Buskulic and Carola Doerr. 2019. Maximizing drift is not optimal for solving OneMax. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion.* 425–426.

[5] Benjamin Doerr and Timo Kötzing. 2021. Multiplicative Up-Drift. *Algorithmica* 83, 11 (2021), 3017–3058. doi:10.1007/s00453-020-00775-7

[6] Stephan Frank and Tobias Glasmachers. 2024. A Potential Potential Function for a Variable-Metric Evolution Strategy. In *Conference on Parallel Problem Solving from Nature (PPSN).* Springer, 221–235.

[7] Armand Gissler. 2024. *Linear Convergence of Evolution Strategies with Covariance Matrix Adaptation.* Ph. D. Dissertation. École Polytechnique.

[8] Tobias Glasmachers. 2020. Global Convergence of the (1+1) Evolution Strategy to a Critical Point. *Evolutionary Computation* 28, 1 (2020), 27–53.

[9] Jens Jägersküpper. 2007. Algorithmic analysis of a basic evolutionary algorithm for continuous optimization. *Theoretical Computer Science* 379, 3 (2007), 329–347.

[10] Stefan Kern, Sibylle D Müller, Nikolaus Hansen, Dirk Büche, Jiri Ocenasek, and Petros Koumoutsakos. 2004. Learning probability distributions in continuous evolutionary algorithms–a comparative review. *Natural Computing* 3, 1 (2004), 77–112.

[11] Timo Kötzing. 2024. *Theory of Stochastic Drift.* Technical Report arxiv:2406.14589. arXiv.org.

[12] Per Kristian Lehre. 2012. Drift Analysis (Tutorial). In *Companion to the Genetic and Evolutionary Computation Conference (GECCO 2012).* ACM Press, 1239–1258.

[13] P. K. Lehre and C. Witt. 2021. Tail bounds on hitting times of randomized search heuristics using variable drift analysis. *Combinatorics, Probability and Computing* 30, 4 (2021), 550–569. doi:10.1017/S0963548320000565

[14] Johannes Lengler. 2020. Drift analysis. *Theory of evolutionary computation: Recent developments in discrete optimization* (2020), 89–131.

[15] Daiki Morinaga and Youhei Akimoto. 2019. 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.* 13–24.

[16] Daiki Morinaga, Kazuto Fukuchi, Jun Sakuma, and Youhei Akimoto. 2022. *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.