Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Modelling Evolutionary Algorithms with Stochastic Differential Equations*

Jorge Pérez Heredia jperezheredia1@sheffield.ac.uk
Department of Computer Science, University of Sheffield,
Sheffield S1 4DP, United Kingdom

Abstract

There has been renewed interest in modelling the behaviour of evolutionary algorithms (EAs) by more traditional mathematical objects, such as ordinary differential equations or Markov chains. The advantage is that the analysis becomes greatly facilitated due to the existence of well established methods. However, this typically comes at the cost of disregarding information about the process. Here, we introduce the use of stochastic differential equations (SDEs) for the study of EAs. SDEs can produce simple analytical results for the dynamics of stochastic processes, unlike Markov chains which can produce rigorous but unwieldy expressions about the dynamics. On the other hand, unlike ordinary differential equations (ODEs), they do not discard information about the stochasticity of the process.

We show that these are especially suitable for the analysis of fixed budget scenarios and present analogues of the additive and multiplicative drift theorems from runtime analysis. In addition, we derive a new more general multiplicative drift theorem that also covers non-elitist EAs. This theorem simultaneously allows for positive and negative results, providing information on the algorithm's progress even when the problem cannot be optimised efficiently. Finally, we provide results for some well-known heuristics namely Random Walk (RW), Random Local Search (RLS), the (1+1) EA, the Metropolis algorithm (MA) and the Strong Selection Weak Mutation (SSWM) algorithm.

Keywords

theory, stochastic differential equations, drift, fixed budget, evolutionary algorithms, non-elitism, metropolis, SSWM, random walk, local search, (1+1) EA, modelling.

1 Introduction

Nowadays the theoretical study of randomised heuristics mainly focuses on rigorous runtime analysis, however we can trace back literature about general models of evolutionary computation to Holland (1975). Roughly a decade later, Goldberg (1987) focused on modelling the evolving population of a genetic algorithm (GA). This was extended by Vose (1995) who used a Markov Chain approach and fixed points analysis. In the same spirit of dynamical systems, we can find an analysis for steady state GAs by Wright and Rowe (2001).

However, Schema theory was proven wrong (Altenberg, 1994) and Markov Chain approaches are, in general, intractable (He and Yao, 2003). The remaining mentioned techniques share a common rationale, the stochastic process describing an EA varies according to its expected change. The same idea is used in the so-called ordinary differential equation (ODE) method by Wormald (1995). Here, the EA is considered as a

*An extended abstract of this article with preliminary results was presented at FOGA '17 by Paixão and Pérez Heredia (2017).

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

noisy version of an ODE that tracks the expected rate of change. This method has been applied several times in EC (see e.g. Yin et al., 1995 or Akimoto et al., 2012). Due to this deterministic approach of the expected change, these methods disregard the random component of EAs.

In this paper we follow up on a new method presented by Paixão and Pérez Heredia (2017) based on stochastic differential equations. This approach naturally includes the stochasticity implicit in randomised heuristics through the well established tools from Itô calculus. SDEs have already been used in EC for convergence analysis by Schaul (2012), however the results did not improve the state of the art.

Other modelling attempts such as the application of statistical physics by Prügel-Bennett and Shapiro (1994) yielded highly accurate results when modelling finite populations and fluctuations were also modelled by Prügel-Bennett (1997). However, this method requires that the algorithm's selection operator weights each search point according to the Boltzmann distribution. This is the case for algorithms like Metropolis or SSWM but it will not hold in general. On the other hand, this method and some of the previously mentioned approaches were tackling more general and ambitious problems than what this paper covers.

Stochastic differential equations have proven to be a powerful tool when dealing with stochastic processes (see Øksendal (2003) for many examples). SDEs have accomplished great achievements in physics, starting with the explanation of Brownian motion by Einstein (1905). In finance, specially for Black and Scholes (1973) work on options prices that yielded a Nobel Prize in Economics in 1997. SDEs have also been used in the more related field of Population Genetics (PG) to track the frequency of genes in a population. This approach was used by Kimura (1957) and has ever since been an integral part of the so-called modern synthesis. Many classical results were obtained through this approach, including the steady state distribution of gene frequencies of a finite population under mutation and selection by Kimura (1964), and the probability of fixation of a gene by Kimura (1968). Our work also contributes to the renewed interest of unifying EC and PG (Paixo et al., 2015; Pérez Heredia et al., 2017).

Building on top of the so-called *diffusion approximation*, we derive a SDE that will be used to model the dynamics of EAs. For some scenarios we will be able to solve this equation analytically. We show that this is the case by obtaining equivalent statements to the well-known additive drift by He and Yao (2001) and multiplicative drift by Doerr et al. (2012). Furthermore we will derive a new version of the multiplicative drift, extending its applicability to non-elitist algorithms even when the drift is pushing the algorithm away from the optimum. The work we present here has several aims:

- improving the understating of EA dynamics,
- deriving analytic expressions for the expectation and the variance,
- introducing SDE to the field of EC and
- translation of drift theorems from runtime analysis to fixed budget analysis.

This paper focuses on a fixed budget analysis, first introduced by Jansen and Zarges (2012). In this perspective, the research question is to derive the state of the algorithm after some specific time, this approach gives more information about the process than the classical runtime analysis where the question is to find the time needed to optimise a problem. Despite being introduced recently, fixed budget is gaining importance in the theoretical community (Doerr et al., 2016, 2013; Jansen and Zarges, 2012; Lengler and Spooner, 2015; Nallaperuma et al., 2017a).

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Although we build on top of the same mathematical tool, this paper widely extends the extended abstract presented by Paixão and Pérez Heredia (2017). First, we will consider more elaborated algorithms such as Metropolis and SSWM. And secondly but more important, we present what to our best knowledge is the first fixed budget analysis of non-elitist EAs. Furthermore, our formulation considers any selection strength. This constitutes an important advantage since drift theorems require to bound the process by a positive and monotonic non-increasing distance function (He and Yao, 2001; Doerr et al., 2012; Johannsen, 2010). In the case of non-elitist EAs, this implies increasing the selection strength such that worsening moves are not very likely to be accepted. However in our results the drift can even be negative and yet we will be able to provide the expected behaviour, and the variance.

The disadvantage of the SDE method, unlike runtime analysis or other fixed budget techniques, is that it requires performing two approximations. The first approximation consists of considering the system as a time-continuous process. In the second approximation, we will assume that the change of the process $X_{t+1} - X_t$ follows a distribution without higher statistical moments than the variance. Hence, this method, although is mathematically based, is not completely rigorous. Due to this reason, the last section of the paper presents an extensive validation of these approximations. We will compare the obtained results against rigorous literature for runtime and fixed budget. In addition, we present a graphical comparison against the simulated results, and we will experimentally measure the error of the approximation.

The paper is structured as follows. In the preliminaries section we introduce the studied algorithms, fitness functions, a crash course on SDEs and the diffusion approximation. Then, we derive three drift theorems for fixed budget: additive, multiplicative and non-elitist multiplicative. In the applications section we will apply these theorems for five different algorithms on problems. Finally we present a validation section where we compare our results against the literature and the experimental error.

2 Preliminaries

2.1 Algorithms

In this paper, we consider the maximisation of pseudo-Boolean functions $f : \{0,1\}^n \to \mathbb{R}$. Although we will be studying five different algorithms, they all fit in the family of trajectory-based algorithms. These heuristics evolve a single individual rather than using a population. The two principal operators that will specify the algorithm used within this class are: *mutation* and *selection*. We will consider algorithms with two mutation operators:

- *Local mutations:* Flip a randomly chosen bit from x.
- *Global mutations:* Flip each bit of x independently at random with probability $1/n$.

As mentioned in the introduction, we are interested in algorithms that deviate from elitism. This property is obtained by the action of the selection operator through an acceptance probability $p_{\text{acc}} : \mathbb{R} \to [0,1]$. For the sake of clarity in the analysis, when analysing non-elitist EAs, we will consider only local mutations yielding a simple but general family of algorithms. Within this class of algorithms we can recover well-known heuristics depending on the acceptance probability. In following sections we present results for this general scheme and we will also introduce four widely extended selection operators to compare our results against the literature.

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Algorithm 1: Trajectory Based Algorithm

```
Initialise x ∈ {0,1}^n
repeat
    y ← MUTATE(x)
    Δf ← f(y) − f(x)
    Choose r ∈ [0,1] uniformly at random
    if r ≤ p_acc(Δf) then
        x ← y
until stop;
```

First we will consider one extreme case of non-elitism which is the absence of selection from a Random Walk (RW) process

$$p_{\text{acc}}^{\text{RW}}(\Delta f) = 1 \quad \forall\, \Delta f. \tag{1}$$

On the other side of the spectrum we find elitist algorithms for which the acceptance function is the step function. This is the case for RLS (local mutations) or the (1+1) EA (global mutations).

$$p_{\text{acc}}^{\text{RLS}}(\Delta f) = \begin{cases} 1 & \text{if } \Delta f \geq 0 \\ 0 & \text{if } \Delta f < 0. \end{cases} \tag{2}$$

All the other local search algorithms will be in between these two limit cases. If worsening moves are allowed with an exponentially decreasing probability, but elitism is kept for improving moves we find the so-called Metropolis et al. (1953) algorithm

$$p_{\text{acc}}^{\text{MA}}(\Delta f, \alpha \in \mathbb{R}^+) = \begin{cases} 1 & \text{if } \Delta f \geq 0 \\ e^{\alpha \Delta f} & \text{if } \Delta f < 0, \end{cases} \tag{3}$$

where $\mathbb{R}^+$ denotes the set of positive real numbers. Finally, we will consider the acceptance probability of the Strong Selection Weak Mutation (SSWM) regime. Within this regime, the condition on improving moves is relaxed allowing the rejection of search points with higher fitness (see e.g. Paixão et al., 2017 or Oliveto et al., 2017).

$$p_{\text{acc}}^{\text{SSWM}}(\Delta f, N, \beta) = p_{\text{fix}}(\Delta f, N, \beta) = \frac{1 - e^{-2\beta \Delta f}}{1 - e^{-2N\beta \Delta f}}. \tag{4}$$

This acceptance probability is obtained from a natural evolutionary regime where $N \in \mathbb{N}^+$ is the population size and $\beta \in \mathbb{R}^+$ is a scaling parameter. This probability is normally referred to as the fixation probability, since it represents the probability of a mutant genotype going extinct or fixating in the population (Kimura, 1962).

Table 1: Mutation and selection operator of each of the studied algorithms.

Algorithm 1	*(1+1) EA*	*RLS*	*RW*	*Metropolis*	*SSWM*
Mutation	Global	Local	Local	Local	Local
Selection	$p_{\text{acc}}^{\text{RLS}}$	$p_{\text{acc}}^{\text{RLS}}$	$p_{\text{acc}}^{\text{RW}}$	$p_{\text{acc}}^{\text{MA}}$	$p_{\text{acc}}^{\text{SSWM}}$

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

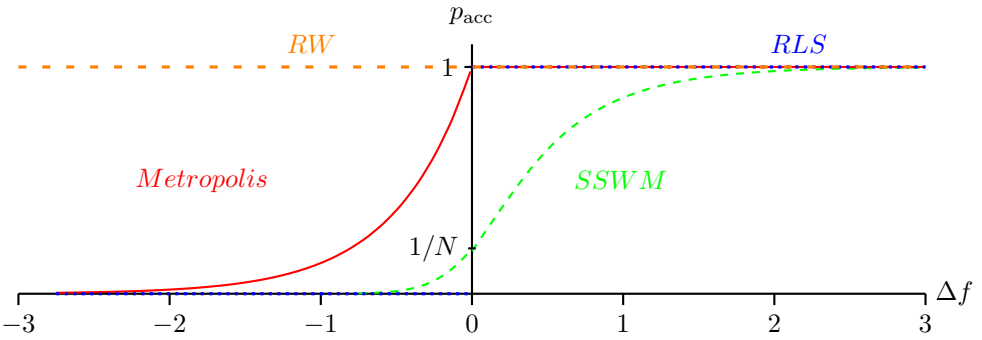

Figure 1: Acceptance probability for RW (orange loosely dashed line), RLS and (1+1) EA (blue dotted line), Metropolis (red solid line) and SSWM (green dashed line).

2.2 Fitness Functions

We will consider two of the most theoretically studied functions. First, the ONEMAX problem represents an easy hill climbing task. The goal is to maximise the number of bit matches with a specific bitstring which is the optimum. Since our algorithms are unbiased, they do not use problem knowledge, without loss of generality we can assume that the optimal search point is the all ones bitstring. Hence, ONEMAX just counts the number of 1-bits of a search point.

Definition 1. *Let* $x \in \{0,1\}^n$, *then* $\text{ONEMAX}(x) = \sum_{i=1}^{n} x_i$.

LEADINGONES is a function that counts the number of 1-bits at the beginning of the bitstring. Using again the landscape metaphor, this problem represents a fitness ridge since bit matches do not always increase the fitness and losing one bit match can yield a huge fitness loss.

Definition 2. *Let* $x \in \{0,1\}^n$, *then* $\text{LEADINGONES}(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$.

2.3 Stochastic Differential Equations

This subsection contains the minimal knowledge of SDEs to be able to understand this paper. For more extended and rigorous details we refer the reader to the textbook by Øksendal (2003). Stochastic differential equations are equations that deal with the behaviour of stochastic processes. We can define these processes as a collection of random variables in $\mathbb{R}$ over time t (i.e. $\{X_t\}_{t\geq 0}$), together with an underlying probability space $(\Omega, \mathcal{F}, P)$ that from now on we will take for granted.

The simplest stochastic process is probably the well-known white noise W_t. Its signal integrated over time will produce another famous process $B_t = \int W_s ds$ known as Brownian motion or Wiener process. Some of the characteristics of these processes are that they have 0 mean and their increments are independent and stationary. Furthermore, all the moments higher than the second are 0 (see e.g. Ross, 1996). We will focus on SDEs of the form

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)W_t dt \;=\; b(t, X_t)dt + \sigma(t, X_t)dB_t.$$

This equation represents a process where the state change dX_t depends on both the time and the current position (which can also depend on the time) through the terms $b(t, X_t)$ and $\sigma(t, X_t)$. These are usually referred to as the drift and diffusion terms or coefficients. We can say that b tracks the expected evolution of the process, note that if

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

$\sigma = 0$ (absence of noise) the drift coefficient simply becomes dX_t/dt. And σ collects the random behaviour by amplifying a white noise or Brownian motion term. Applying the usual integration rules one obtains a simple expression for the state after t iterations.

$$X_t = X_0 + \int_0^t b(s, X_s)ds + \int_0^t \sigma(s, X_s)dB_s. \tag{5}$$

The challenge now is to prove the existence of an integral over an stochastic process $\int gdB_s$ and develop its integration rules. This was done by Itô (1944, 1946, 1950, 1951): he proved the existence of this integral, worked on stochastic integral equations of the type (5) and applied his developments to the already existing field of SDEs. The key idea of Itô's work is a stochastic extension of the Riemann-Stieltjes (1894) integral, which is itself, an extension of the standard Riemann integral for when a function is integrated with respect to another function. The other traditional formulation was given by Stratonovich (1966). In this paper, we will focus on stochastic processes that fit into the following definition of an Itô process (Definition 4.1.1 in Øksendal, 2003).

Definition 3 (1-dimensional Itô process)**.** *A 1-dimensional Itô process is a stochastic process X_t on $(\Omega, \mathcal{F}, P)$ of the form*

$$X_t = X_0 + \int_0^t b(s, X_s)ds + \int_0^t \sigma(s, X_s)dB_s.$$

where dB_t denotes a Brownian motion process, b is absolute integrable in $[0, t]$ and σ is square integrable in $[0, t]$. Alternatively, in differential form $dX_t = b(t, X_t)dt + \sigma(t, X_t)dB_t$.

In the following we will refer as Itô integrals to integrals over a Brownian process $\int gdB_s$. Again, for a formal definition of these integrals and the set of functions $\mathcal{V}(S, T)$ on where they are defined, we refer the reader to Øksendal (2003). These integrals will have the following properties (adapted from Theorem 3.2.1 in Øksendal, 2003).

Theorem 1 (Properties of the Itô integral)**.** *Let $f, g \in \mathcal{V}(0, T)$ and let $0 \leq S < U < T$. Then*

(i) $\int_S^T fdB_t = \int_S^U fdB_t + \int_U^T fdB_t$

(ii) $\int_S^T (cf + g)dB_t = c \cdot \int_S^T fdB_t + \int_S^T gdB_t$ *(c constant)*

(iii) $E\left(\int_S^T fdB_t\right) = 0$

Performing a variable change on a stochastic integral is a bit more involved than for ordinary integrals. The usual rules do not apply and we will have to use the following statement (Theorem 4.1.2 in Øksendal, 2003) which is known as Itô's formula.

Theorem 2 (Itô formula)**.** *Let X_t be a Itô process given by*

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)dB_t.$$

Let $g(t, x)$ be twice continuously differentiable on $x \in \mathbb{R}$ and once in $t \in \mathbb{R}^+$. Then $Y_t = g(t, X_t)$ is again an Itô process, and

$$dY_t = \left(\frac{\partial g}{\partial t} + b\frac{\partial g}{\partial x} + \frac{\sigma^2}{2}\frac{\partial^2 g}{\partial x^2}\right) \cdot dt + \sigma\frac{\partial g}{\partial x}dB_t.$$

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Finally, another useful property specially when computing higher moments, such as the variance, is the Itô Isometry (Corollary 3.1.7 in Øksendal, 2003).

Theorem 3 (Itô Isometry)**.** *For all* $f \in \mathcal{V}(S,T)$

$$E\left(\left(\int_S^T f(X_t,t)dB_t\right)^2\right) = E\left(\int_S^T f(X_t,t)^2 dt\right).$$

2.4 The Diffusion Approximation

We refer to the diffusion approximation as the joint application of a time-continuous and Gaussian approximation. The time evolution of the probability density $p(x,t)$ of any stochastic process X_t can be described by the Chapman-Kolmogorov equation (see e.g. Feller, 1949)

$$p(x,t+\Delta t) = \int \Delta(\delta \mid x) \cdot p(x-\delta,t) \cdot d\delta \tag{6}$$

where $\Delta(\delta \mid x)$ is the transition probability of reaching x from $x-\delta$. Let us consider the left hand term: after performing a first order Taylor expansion around $\Delta t = 0$ we obtain

$$p(x,t+\Delta t) \approx p(x,t) + \Delta t \cdot \frac{\partial p(x,t)}{\partial t}$$

and by Taylor's theorem we know that the error of this approximation will be of order $O(\Delta t)$. Since we are dealing with time-discrete algorithms, the time counter is increased every generation by just one unit, hence $\Delta t = 1$. Similarly, for the right-hand term of (6) we can write the second order Taylor expansion of $\Delta(\delta \mid x)p(x-\delta,t)$ around $x-\delta = x$ yielding

$$\Delta(\delta \mid x)p(x-\delta,t) \approx \Delta(\delta \mid x)p(x,t) - \delta \cdot \frac{\partial}{\partial x}\left(\Delta(\delta \mid x)p(x,t)\right) + \frac{\delta^2}{2} \cdot \frac{\partial^2}{\partial x^2}\left(\Delta(\delta \mid x)p(x,t)\right).$$

Again, by Taylor's theorem we can estimate the error by $O(\delta^2)$. This approximation will give good results when δ is small, i.e., when the system usually moves between close by states in the search space. Introducing both approximations in (6) and noticing that p no longer depends on δ we can write

$$\begin{aligned} p(x,t) + \Delta t \cdot \frac{\partial p(x,t)}{\partial t} \approx\ & p(x,t) \cdot \underbrace{\int \Delta(\delta \mid x) d\delta}_{1} \\ & - \frac{\partial}{\partial x}\left(p(x,t) \cdot \underbrace{\int \delta \cdot \Delta(\delta \mid x) d\delta}_{E(\Delta)}\right) \\ & + \frac{1}{2} \cdot \frac{\partial^2}{\partial x^2}\left(p(x,t) \cdot \underbrace{\int \delta^2 \cdot \Delta(\delta \mid x) d\delta}_{E(\Delta^2)}\right). \end{aligned}$$

As outlined in the equation above, we can identify these integrals with the statistical moments of Δ leading to the well-known diffusion equation

$$\Delta t \cdot \frac{\partial p(x,t)}{\partial t} \approx -\frac{\partial}{\partial x}\left(p(x,t) \cdot \mathrm{E}\left(\Delta\right)\right) + \frac{1}{2} \cdot \frac{\partial}{\partial x^2}\left(p(x,t) \cdot \mathrm{E}\left(\Delta^2\right)\right).$$

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

This equation is known as the Fokker-Planck equation or the Forward Kolmogorov equation (see e.g. Karlin and Taylor, 1981). This is an equation for the time evolution of the probability density $p(x,t)$ of the random variable X_t. In principle, one could solve this equation to obtain a probability distribution at any particular time point. However, this is often impossible in practice. Instead, we deal directly with the SDE associated with the Fokker-Planck equation and take that as a model for EAs. It can be shown (see e.g. Øksendal, 2003), that the associated SDE is an Itô process of the form:

$$dX_t \approx \mathrm{E}\left(\Delta\right) dt + \sqrt{\mathrm{E}\left(\Delta^2\right)} dB_t.$$

This is the central equation for this paper. We will use it to describe the dynamics of EAs by computing its drift coefficient $\mathrm{E}\left(\Delta\right)$ and second moment or diffusion coefficient $\sqrt{\mathrm{E}\left(\Delta^2\right)}$. Finally, we present our model as an hypothesis where it is interesting to note that the first moment $E(\Delta)$ exactly corresponds to the drift as used in runtime analysis.

Hypothesis 1 (The Diffusion Approximation)**.** *The dynamics of any algorithm described by Algorithm 1 with local or global mutations can be approximated by the following Itô process:*

$$\begin{aligned} dX_t &= b(X_t,t)dt + \sigma(X_t,t)dB_t \\ b(X_t,t) &= E\left(X_{t+1} - X_t | X_t\right) \\ \sigma^2(X_t,t) &= E\left(\left(X_{t+1} - X_t\right)^2 | X_t\right), \end{aligned}$$

where B_t is a Brownian motion process.

Are Evolutionary Algorithms Diffusive Processes?

The previous hypothesis would be a rigorous theorem if evolutionary algorithms were a pure diffusion process, but they are not. First of all, we used differential and not difference equations to model a discrete process. This approximation is intrinsic to the chosen method and carries an error of constant size. Secondly, the Gaussian approximation depends on each algorithm and the fitness function. The variable of interest is the transition probability $\Delta(\delta \mid x)$ from equation (6). Which reads as the probability of reaching the state x from the state $x - \delta$ in one iteration. It was shown before that the Gaussian approximation will carry an error of size $O(\delta^2)$. Hence, depending on the evolutionary operators of the EA this approximation will or will not be accurate.

To motivate the use of this approximation for the algorithms from subsection 2.1, we can focus just on the mutation operator. The reason is that selection cannot increase the distance δ between the parent and child solutions. It will only decide if the new individual is accepted or rejected.

The question resides then in how big is the δ produced by mutation. To illustrate this, let us consider a random walk on the Boolean hypercube and its number of ones x. First, with local mutations, the RW will have a constant $\delta = 1$. On the other hand, if global mutations are used δ can be any integer number in $\{0, 1, \ldots, n\}$. Fortunately, the probability of large mutations decays exponentially with their size δ. Actually, the two smallest values of δ (0 and 1) will carry together a probability mass of approximately $2/e$. Hence, all the larger jumps together will only have a probability of $\approx 1 - 2/e$. Since δ is concentrated, the higher order moments should be close to zero, suggesting that the Gaussian approximation is also accurate for global mutations. However, on some fitness problems all the relevant events could belong to the neglected probability mass of $1 - 2/e$. In this paper, we only consider problems (ONEMAX and LEADINGONES) where, as we will see, global mutations are decently Gaussian approximated. Nevertheless, at the end we provide an extensive validation for Hypothesis 1.

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

3 Drift Theorems for Fixed Budget Analysis

The Diffusion Approximation (Hypothesis 1) considers a scenario which in general does not have an analytic solution. This will be dependant on the mathematical expressions of the drift $b(X_t, t)$ and diffusion $\sigma(X_t, t)$ coefficients. In this paper we will consider three simple scenarios where we can obtain equivalent results to drift theorems for runtime analysis.

The methods to solve SDEs are similar to those to solve ODE, but we have to use the Itô formula (Theorem 2) when performing a change of variables. In the following subsections our approach is to use *a priori* knowledge. We will find another process Y_t, described by a simpler function $g(x, t)$ that we know is the solution of the equivalent ODE (SDE with $\sigma = 0$). For example, if our process is of the form $dX_t = X_t dt + X_t dB_t$ our candidate solution will be $Y_t = \ln(X_t)$. When the coefficients are more complex, we will try to find a variable change that leads to an already known SDE.

Before starting the derivation of these theorems, we would like to emphasize that this section is purely rigorous. No approximations are used unlike in the derivation of Hypothesis 1.

3.1 Additive Drift

The simplest case is when the drift $b(X_t, t)$ does not depend on the time t or the current state X_t, i.e., $b(X_t, t) = b$. This corresponds with the additive drift for runtime analysis (He and Yao, 2001).

Theorem 4. *Let X_t be an Itô process of the form*

$$dX_t = bdt + \sigma dB_t$$

where B_t is a Brownian process, $b \in \mathbb{R}$ and $\sigma \in \mathbb{R}_0^+$. Then,

$$\begin{aligned} X_t &= X_0 + bt + \sigma B_t \\ E(X_t) &= E(X_0) + bt \\ Var(X_t) &= Var(X_0) + \sigma^2 t \end{aligned}$$

Proof. This is a trivial case that can be solved by directly integrating

$$\int_0^t dX_s = \int_0^t b ds + \int_0^t \sigma dB_s \tag{7}$$

$$X_t = X_0 + bt + \sigma B_t \quad (B_0 = 0). \tag{8}$$

Taking expectations in equation (7) we can compute the expected value

$$\begin{aligned} \mathrm{E}(X_t) &= \mathrm{E}(X_0) + \mathrm{E}\left(\int_0^t b ds\right) + \mathrm{E}\left(\int_0^t \sigma dB_s\right) \\ &= \mathrm{E}(X_0) + \int_0^t \mathrm{E}(b)\, ds \; = \mathrm{E}(X_0) + bt. \end{aligned} \tag{9}$$

where we have used property (iii) from Theorem 1. Finally, to compute the variance we use the results obtained in (8) and (9) as follows

$$\mathrm{Var}(X_t) = \mathrm{E}\left((X_t - \mathrm{E}(X_t))^2\right) = \mathrm{E}\left((X_0 - \mathrm{E}(X_0))^2 + \sigma^2 B_t^2\right) = \mathrm{Var}(X_0) + \sigma^2 \cdot \mathrm{E}\left(B_t^2\right).$$

Considering a normalised Brownian motion ($\mathrm{E}\left(B_t^2\right) = t$) yields the claimed statement. □

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

As in runtime analysis it is not typically the case that we know exactly the drift's value, but we can bound it within some range. It follows directly that knowing those bounds we will be able to bound the expected progress after t iterations.

Corollary 5. *In the context of Theorem 4, if $b_\ell \leq b \leq b_u$ and $\sigma_\ell \leq \sigma \leq \sigma_u$, with $b_\ell, b_u \in \mathbb{R}$ and $\sigma_\ell, \sigma_u \in \mathbb{R}_0^+$. Then*

$$\begin{aligned} X_0 + b_\ell t + \sigma_l B_t \leq X_t \leq\ & X_0 + b_u t + \sigma_u B_t \\ E(X_0) + b_\ell t \leq E(X_t) \leq\ & E(X_0) + b_u t \\ Var(X_0) + \sigma_\ell^2 t \leq Var(X_t) \leq\ & Var(X_0) + \sigma_u^2 t. \end{aligned}$$

3.2 Multiplicative Drift

Analogous to runtime analysis, the assumption of additive drift is in general too loose since it does not take into account the current state of the process. This is solved by the multiplicative drift theorem from Doerr et al. (2012) when the drift is proportional to the current state X_t.

As discussed in the introduction, SDEs are a well established method in other research fields, allowing us to recycle already existing results. We will make use of a simplified version of the Cox-Ingersoll-Ross (CIR) model (Cox et al., 1985; Maghsoodi, 1996), which describes the evolution of interest rates with the following SDE

$$dX_t = b(\theta - X_t)dt + \sigma\sqrt{X_t}dB_t. \tag{10}$$

The coefficient b specifies the speed (drift) at which the process approaches its expected value θ provoking a mean reversion effect. The noise due to the market dB_t is amplified by a term $\sigma\sqrt{X_t}$ to ensure that there are not negative interest rates (note that this effect is also obtained with any diffusion coefficient that goes to 0 when $X_t = 0$). We will use a simplified version without the mean reversion effect ($\theta = 0$), which resembles an elitist multiplicative drift process.

Theorem 6 (CIR without mean reversion, Cox et al., 1985)**.** *Let $X_t > 0$ be an Itô process of the form*

$$dX_t = -bX_t dt + \sigma\sqrt{X_t}dB_t$$

where B_t is a Brownian process and $b \in \mathbb{R}^+$ and $\sigma \in \mathbb{R}_0^+$. Then,

$$\begin{aligned} E(X_t) &= E(X_0) \cdot e^{-bt} \\ Var(X_t) &= E(X_0)\frac{\sigma^2}{b} \cdot e^{-bt}\left(1 - e^{-bt}\right) \end{aligned}$$

Proof. The results for the expectation and variance of the original CIR process (10) are (equation (19) in Cox et al., 1985)

$$\begin{aligned} \mathrm{E}(X_t) &= \mathrm{E}(X_0) \cdot e^{-bt} + \theta\left(1 - e^{-t}\right) \\ \mathrm{Var}(X_t) &= \mathrm{E}(X_0)\frac{\sigma^2}{b} \cdot e^{-bt}\left(1 - e^{-bt}\right) + \theta\frac{\sigma^2}{2b}\left(1 - e^{-t}\right)^2 \end{aligned}$$

Our statement is just the special case when $\theta = 0$. □

Once more, if we can only compute bounds rather than exact values it follows straightforward the next corollary. Notice that the result for the expectation is very similar to a previous fixed-budget multiplicative drift by Lengler and Spooner (2015) (Theorem 1).

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Corollary 7. *In the context of Theorem 6, if $b_\ell \leq b \leq b_u$ and $\sigma_\ell \leq \sigma \leq \sigma_u$ with $b_\ell, b_u \in \mathbb{R}^+$ and $\sigma_\ell, \sigma_u \in \mathbb{R}_0^+$. Then,*

$$E(X_0) \cdot e^{-b_\ell t} \geq E(X_t) \geq E(X_0) \cdot e^{-b_u t}$$
$$Var(X_t) \geq E(X_0) \frac{\sigma_\ell^2}{b_\ell} e^{-b_\ell t} \left(1 - e^{-b_\ell t}\right)$$
$$Var(X_t) \leq E(X_0) \frac{\sigma_u^2}{b_u} e^{-b_u t} \left(1 - e^{-b_u t}\right)$$

3.3 Non-elitist Multiplicative Drift

Drift theorems for the runtime share a common condition - the drift must be bounded by a monotonic function towards the optimum (He and Yao, 2001; Doerr et al., 2012; Johannsen, 2010). This limitation is also shared by previous fixed budget drift theorems (Lengler and Spooner, 2015 and Theorem 6). In the following statement we relax this condition - the process also follows a monotonic behaviour in expectation, but towards the stationary distribution (which can be far away from the optimum). In fact, as we will see in the experimental section, even when the process starts near the optimum it will follow an exponential decay towards its equilibrium state, moving away from the optimum.

Although we have named it *Non-Elitist Multiplicative Drift* this theorem can also be applied for elitist algorithms. This will be the limit case when $b = \alpha = 0$ and $\beta < 0$ where we recover Theorem 6.

Theorem 8. *Let $X_t \geq 0$ be an Itô process of the form*

$$dX_t = a(b - X_t)dt + \sqrt{\alpha - \beta X_t} dB_t$$

with $a \in \mathbb{R}^+$, $b, \alpha \in \mathbb{R}_0^+$, $\beta \in \mathbb{R}$, $\alpha - \beta X_t \geq 0$ for all X_t, $\alpha - b\beta \geq 0$ and B_t is a Brownian process. Then,

$$X_t = X_0 e^{-at} + b\left(1 - e^{-at}\right) + \int_0^t e^{-a(t-s)} \sqrt{\alpha - \beta X_s} dB_s$$
$$E(X_t \mid X_0) = X_0 e^{-at} + b\left(1 - e^{-at}\right)$$
$$Var(X_t \mid X_0) = \frac{\alpha}{2a}(1 - e^{-2at}) - \frac{\beta}{a} X_0 e^{-at}(1 - e^{-at}) - \frac{b\beta}{2a}\left(1 - e^{-at}\right)^2$$

Proof. Let us use the integrating factor e^{at} to perform the variable change $Y_t = e^{at} X_t$, then by applying Itô formula (Theorem 2) we obtain

$$\begin{aligned} dY_t &= \left(\frac{\partial(e^{at}X_t)}{\partial t} + a(b - X_t)\frac{\partial(e^{at}X_t)}{\partial X_t} + \frac{\alpha - \beta X_t}{2}\frac{\partial^2(e^{at}X_t)}{\partial X_t^2} \right) \cdot dt \\ &\quad + \sqrt{\alpha - \beta X_t}\frac{\partial(e^{at}X_t)}{\partial X_t} dB_t \\ &= \left(ae^{at}X_t + a(b - X_t)e^{at}\right) \cdot dt + \sqrt{\alpha - \beta X_t} e^{at} dB_t \\ &= abe^{at}dt + \sqrt{\alpha - \beta X_t} e^{at} dB_t. \end{aligned}$$

Applying the usual integration rules leads to

$$Y_t = Y_0 + ab \int_0^t e^{as} ds + \int_0^t \sqrt{\alpha - \beta X_s} e^{as} dB_s.$$

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Let us now compute just the first integral and also revert the variable change. Then,

$$e^{at}X_t = X_0 + b\left(e^{at}-1\right) + \int_0^t \sqrt{\alpha - \beta X_s}e^{as}dB_s.$$

Multiplying both sides by e^{-at} leads to the first claimed result in the theorem. For the second result we take expectations leading to

$$\mathrm{E}\left(X_t\right) = \mathrm{E}\left(X_0\right)e^{-at} + b\left(1-e^{-at}\right) + \mathrm{E}\left(\int_0^t e^{-a(t-s)}\sqrt{\alpha-\beta X_s}dB_s\right).$$

Fortunately, expectations of integrals over a Brownian motion average out (see property (iii) from Theorem 1) yielding the second result of the theorem. Finally, to compute the variance we will start from the formula $\mathrm{Var}\left(X_t\right) = \mathrm{E}\left(\left(X_t - \mathrm{E}\left(X_t\right)\right)^2\right)$ which after introducing the two previous results yields

$$\mathrm{Var}\left(X_t \mid X_0\right) = \mathrm{E}\left(\left(\int_0^t \sqrt{\alpha-\beta X_s}e^{-a(t-s)}dB_s\right)^2\right)$$

applying Itô isometry (Theorem 3) we obtain

$$\mathrm{Var}\left(X_t \mid X_0\right) = \int_0^t \left(\alpha - \beta\mathrm{E}\left(X_s\right)\right)e^{-2a(t-s)}ds = \alpha e^{-2at}\int_0^t e^{2as}ds - \beta e^{-2at}\int_0^t \mathrm{E}\left(X_s\right)e^{2as}ds$$

introducing the result for the expectation leads to

$$\begin{aligned}\mathrm{Var}\left(X_t \mid X_0\right) &= \alpha e^{-2at}\int_0^t e^{2as}ds - \beta e^{-2at}\int_0^t \left(X_0e^{-as} + b\left(1-e^{-as}\right)\right)e^{2as}ds \\ &= \alpha e^{-2at}\int_0^t e^{2as}ds - \beta e^{-2at}X_0\int_0^t e^{as}ds - b\beta e^{-2at}\int_0^t \left(e^{2as}-e^{as}\right)ds\end{aligned}$$

computing the integrals completes the proof

$$\begin{aligned}\mathrm{Var}\left(X_t \mid X_0\right) &= \alpha e^{-2at}\cdot\frac{e^{2at}-1}{2a} - \beta e^{-2at}X_0\cdot\frac{e^{at}-1}{a} - b\beta e^{-2at}\cdot\frac{\left(e^{at}-1\right)^2}{2a} \\ &= \frac{\alpha}{2a}(1-e^{-2at}) - \frac{\beta}{a}X_0e^{-at}(1-e^{-at}) - \frac{b\beta}{2a}\left(1-e^{-at}\right)^2.\square\end{aligned}$$

Once more, if exact values for the drift and diffusion coefficients are hard to compute, but can be bounded we can make use of the following corollary.

Corollary 9. *In the context of Theorem 8, if* $a_\ell(b_\ell - X_t) \leq a(b-X_t) \leq a_u(b_u - X_t) \in R^+$ *for all* X_t*, and* $\alpha_\ell - \beta_\ell X_t \leq \alpha - \beta X_t \leq \alpha_u - \beta_u X_t \in R_0^+$ *for all* X_t*. Then,*

$$\begin{aligned}E\left(X_t \mid X_0\right) &\leq X_0e^{-a_ut} + b_u\left(1-e^{-a_ut}\right) \\ E\left(X_t \mid X_0\right) &\geq X_0e^{-a_\ell t} + b_\ell\left(1-e^{-a_\ell t}\right) \\ Var\left(X_t \mid X_0\right) &\leq \frac{\alpha_u}{2a_u}(1-e^{-2a_ut}) - \frac{\beta_u}{a_u}X_0e^{-a_ut}(1-e^{-a_ut}) - \frac{b_u\beta_u}{2a_u}\left(1-e^{-a_ut}\right)^2 \\ Var\left(X_t \mid X_0\right) &\geq \frac{\alpha_\ell}{2a_\ell}(1-e^{-2a_\ell t}) - \frac{\beta_\ell}{a_\ell}X_0e^{-a_\ell t}(1-e^{-a_\ell t}) - \frac{b_\ell\beta_\ell}{2a_\ell}\left(1-e^{-a_\ell t}\right)^2.\end{aligned}$$

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

4 Applications

Although the results derived in the previous section are rigorous, its application for evolutionary algorithms is just an approximation (see subsection 2.4). In this section we will take for granted that the diffusion approximation holds and we will relax our statements to the category of *application* rather than *theorem*. It will be finally in section 5 where we will extensively validate this assumption (Hypothesis 1).

4.1 Elitist Algorithms on LEADINGONES

First, we start with a study case for the additive drift (Theorem 4). A good problem candidate, although not ideal, is the LEADINGONES function. Many algorithms like RLS or the (1+1) EA have an *almost* additive drift on this function, however at the end of the optimisation process the drift depends on the state. The reason this fitness function is not ideal for an additive drift approach are the fitness boundary conditions ($0 \leq x \leq n$). Our formulation of additive drift does not consider boundaries on the process. SDEs can cope with these conditions but this will highly increase the complexity and extension of the paper deviating from our introductory interests. To circumvent this problem, we will add a clause in the application's statement excluding our results after the optimum is reached. This approach has already been used in the literature as in Nallaperuma et al. (2017a).

Application 10. *Under the Diffusion Approximation, after t iterations, RLS on* LEADINGONES *will find the optimum or reach an expected value and variance of:*

$$E(X_0) + \frac{1}{n}t \leq E(X_t) \leq E(X_0) + \frac{2}{n}t$$
$$E(X_0) + \frac{1}{n}t \leq Var(X_t) \leq Var(X_0) + \frac{4}{n}t$$

where X_0 is the number of leading ones of the initial search point.

Proof. Let us denote by X_t the number of leading ones in the bitstring at time t. And assume that Hypothesis 1 is a good approximation for the dynamics of X_t. As explained in the introduction, the drift and diffusion coefficients will be the first and second moments of the rate of change of the process. Or mathematically speaking

$$b = \mathrm{E}(X_{t+1} - X_t \mid X_t) = \sum_x x \cdot p(X_{t+1} - X_t = x \mid X_t)$$
$$\sigma^2 = \mathrm{E}\left((X_{t+1} - X_t)^2 \mid X_t\right) = \sum_x x^2 \cdot p((X_{t+1} - X_t)^2 = x \mid X_t).$$

Recall that RLS flips exactly one bit per iteration, this bit can either be: (A) one of the X_t leading ones bits, (B) the first 0-bit or (C) one of the remaining $n - 1 - X_t$ bits (free riders). Since RLS is an elitist algorithm, we can neglect events of type A (fitness decrease). In the case of events of type C, although the algorithm will accept the movement towards a point of equal fitness, it will not yield a contribution to the drift since $x = 0$. Hence, we focus just on event B and using the law of total expectation we obtain

$$b(X_t) = \sum_{x=1}^{n-X_t} x \cdot p(X_{t+1} - X_t = x \mid X_t, B)p(B).$$

Note that flipping the first 0-bit (probability $1/n$) automatically increases the fitness by 1, however it could even be the case that the remaining $n-1-X_t$ bits (free riders) were

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

already set to 1 and the problem is immediately optimised. As described before, events of type C are accepted and free riders bits are being flipped no matter its value, i.e, they are uniformly at random distributed (Lemma 1 in Lehre and Witt, 2012). Hence, the number of free riders set to 1 follows a geometric distribution with parameter $1/2$. Therefore,

$$b(X_t) = \frac{1}{n} \cdot \left(\sum_{x=1}^{n-X_t-1} x \cdot \frac{1}{2} \left(1 - \frac{1}{2}\right)^{x-1} + (n - X_t) 2^{-(n-X_t)} \right).$$

The reason for the term $(n - X_t)2^{-(n-X_t)}$ is that the problem size n is finite. Then, the event where all the free riders were already set to 1 absorbs the remaining probability mass. Computing the sum yields the already known result for the drift of RLS on LEADINGONES

$$b(X_t) = \frac{1}{n} \cdot \left(2 - 2^{-(n-1-X_t)}\right).$$

This result can be understood as the probability of flipping the first 0-bit, times the drift impact of such mutation. We now bound the drift by bounding the impact in the range $[1, 2)$ yielding

$$b_l = \frac{1}{n}, \quad b_u = \frac{1}{n} \cdot 2. \tag{11}$$

To compute the diffusion coefficient σ we use the same rationale but raising the impact to the power of 2, therefore

$$\sigma_l^2 = \frac{1}{n} \quad \sigma_u^2 = \frac{1}{n} \cdot 2^2$$

These coefficient bounds fit in our additive drift description (Corollary 5), using this corollary with the previously computed coefficients directly leads to the claimed results. □

The application to the (1+1) EA is equivalent but we have to deal with global mutations, which makes computing the exact value for the drift and diffusion coefficients harder.

Application 11. *Under the Diffusion Approximation, after t iterations, the (1+1) EA on* LEADINGONES *will find the optimum or reach an expected value and variance of:*

$$E(X_0) + \frac{1}{en}t \leq E(X_t) \leq E(X_0) + \frac{2}{n}t$$
$$Var(X_0) + \frac{1}{en}t \leq Var(X_t) \leq Var(X_0) + \frac{4}{n}t$$

where X_0 is the number of leading ones of the initial search point.

Proof. Let us denote by X_t the number of leading ones in the bitstring at time t. And assume that Hypothesis 1 is a good approximation for the dynamics of X_t.

As in the previous application, the drift can be expressed with two terms: the probability of having a positive drift and its impact. The impact term remains identical, but the mutational term is different since we have to ensure that all the leading 1-bits of the current solution are preserved.

$$b(X_t) = \frac{1}{n} \left(1 - \frac{1}{n}\right)^{X_t} \cdot \left(2 - 2^{-(n-1-X_t)}\right).$$

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

To obtain the bounds for the drift we will consider the extreme cases $X_t = 0$ and $X_t = n - 1$ for the mutational term and bound the impact in the range $[1, 2)$ yielding

$$b \leq \frac{1}{n} \cdot 2 = b_u \tag{12}$$

$$b \geq \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-1} \cdot 1 \geq \frac{1}{en} = b_l \tag{13}$$

For the diffusion coefficient we can recycle the previous probabilities but we have to raise the impact to the power of 2 to obtain the second moment of the process

$$\sigma_l^2 = \frac{1}{en} \cdot 1^2 \leq \sigma^2 \leq \frac{1}{n} \cdot 2^2 = \sigma_u^2$$

Finally, introducing these calculations in Corollary 5 proves the result. □

4.2 Elitist Algorithms on ONEMAX

Secondly we apply the multiplicative drift for the same algorithms but for the ONEMAX problem, which is ideal for this drift theorem. Unlike in LEADINGONES, the boundary conditions are not a problem since our multiplicative drift bounds can not push the system out of the fitness interval $[0, n]$. Notice that the drift is always increasing the number of ones and vanishes when approaching the optimum.

Application 12. *Under the Diffusion Approximation, the expected fitness reached by RLS in* ONEMAX *after t iterations is*

$$E(X_t) = n\left(1 - e^{-\frac{t}{n}}\right) + E(X_0) \cdot e^{-\frac{t}{n}}$$

$$Var(X_t) = (n - E(X_0)) \cdot e^{-\frac{t}{n}} \left(1 - e^{-\frac{t}{n}}\right)$$

where $E(X_0)$ is the expected number of ones of the initial search point.

Proof. Let us denote by Z_t the number of ones in the bitstring at time t. And assume that Hypothesis 1 is a good approximation for the dynamics of Z_t. The drift coefficient, as always, will be the first moment (expectation) of the process. The diffusion coefficient will be the second moment.

$$b(Z_t) = \mathrm{E}(Z_{t+1} - Z_t \mid Z_t) = \sum_z z \cdot p(Z_{t+1} - Z_t = z \mid Z_t)$$

$$\sigma^2(Z_t) = \mathrm{E}\left((Z_{t+1} - Z_t)^2 \mid Z_t\right) = \sum_z z^2 \cdot p((Z_{t+1} - Z_t)^2 = z \mid Z_t)$$

In the ONEMAX problem at state Z_t there are Z_t bit-flips out of n that lead to an improvement, with each of these events reducing the number of zeros by 1, therefore

$$b(Z_t) = -\frac{Z_t}{n}.$$

For the diffusion coefficient we just have to repeat the calculations raising to 2 the impact of each event $(-1)^2$, yielding

$$\sigma^2(Z_t) = (-1)^2 \frac{Z_t}{n} = \frac{Z_t}{n}.$$

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Therefore the approximated SDE for this process is of the form $dZ_t = b(Z_t)dt + \sigma\left(\sqrt{Z_t}\right)dB_t$, which fits in our multiplicative drift (Theorem 6). Applying this theorem with $b = -1/n$ and $\sigma = 1/\sqrt{n}$ we obtain the following results

$$\begin{aligned}\mathrm{E}\left(Z_t\right) &= \mathrm{E}\left(Z_0\right)\cdot e^{-\frac{t}{n}}\\ \mathrm{Var}\left(Z_t\right) &= \mathrm{E}\left(Z_0\right)\cdot e^{-\frac{t}{n}}\left(1-e^{-\frac{t}{n}}\right)\end{aligned}$$

Finally translating to the number of ones $X_t = n - Z_t$ leads to the theorem's statement. □

The next application is regarding the (1+1) EA, again in this case we have computed bounds on the coefficients rather than exact values due to the difficulty to obtain exact results with global mutations.

Application 13. *Under the Diffusion Approximation, the expected fitness reached by the (1+1) EA in* ONEMAX *after t iterations is bounded by*

$$\begin{aligned}E\left(X_t\right) &\geq n\left(1-e^{-\frac{t}{en}}\right)+E\left(X_0\right)\cdot e^{-\frac{t}{en}}\\ E\left(X_t\right) &\leq n\left(1-e^{-\frac{3.1t}{n}}\right)+E\left(X_0\right)\cdot e^{-\frac{3.1t}{n}}\\ Var\left(X_t\right) &\geq \left(n-E\left(X_0\right)\right)\cdot e^{-\frac{t}{en}}\left(1-e^{-\frac{t}{en}}\right)\\ Var\left(X_t\right) &\leq \left(n-E\left(X_0\right)\right)\cdot e^{-3.1\frac{t}{n}}\left(1-e^{-3.1\frac{t}{n}}\right)\end{aligned}$$

where $E\left(X_0\right)$ is the expected number of ones of the initial search point.

Proof. Let us denote by Z_t the number of ones in the bitstring at time t. And assume that Hypothesis 1 is a good approximation for the dynamics of Z_t. The drift and diffusion coefficients (first and second statistic moments of the process) can be expressed as

$$b(Z_t) = \sum_{j=1}^{Z_t} -j\cdot \mathrm{mut}(Z_t, Z_t-j), \quad \sigma^2(Z_t) = \sum_{j=1}^{Z_t}(-j)^2\cdot \mathrm{mut}(Z_t, Z_t-j),$$

where $\mathrm{mut}(Z_t, Z_t-j)$ is the probability of mutation moving the process from Z_t to $Z_t - j$. This is obtained by k 1-bits flipped into 0-bits and vice-versa for $k+j$ bits. Then,

$$\mathrm{mut}(Z_t, Z_t-j) = \sum_{k=0}^{n}\binom{n-Z_t}{k}\binom{Z_t}{k+j}\left(\frac{1}{n}\right)^{j+2k}\left(1-\frac{1}{n}\right)^{2n-j-2k}.$$

The drift coefficient can be lower bounded using only the case when $j = 1$ leading

$$b(Z_t) \geq -\frac{Z_t}{n}\left(1-\frac{1}{n}\right)^{n-1} \geq -\frac{Z_t}{ne}.$$

For the upper bound, we make use of Lemma 3 from Paixão et al. (2017) to upper bound $\mathrm{mut}(Z_t, Z_t+j)$ by $\left(\frac{Z_t}{n}\right)^j\cdot\left(1-\frac{1}{n}\right)^{n-j}\cdot\frac{1.14}{j!}$ yielding

$$b(Z_t) = -\sum_{j=1}^{Z_t}\left(\frac{Z_t}{n}\right)^j\cdot\left(1-\frac{1}{n}\right)^{n-j}\cdot\frac{1.14}{j!}\cdot j$$

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

$$\leq -1.14 \cdot \frac{Z_t}{n} \cdot \sum_{j=1}^{\infty} \frac{j}{j!} = -1.14 \cdot \frac{Z_t}{n} \cdot e \leq -3.1 \cdot \frac{Z_t}{n}.$$

Analogous calculations lead to the following bounds on the diffusion coefficient

$$\frac{Z_t}{en} \leq \sigma^2(Z_t) \leq 3.1 \cdot \frac{Z_t}{n}.$$

Calling Corollary 7 with $-1/en \leq b \leq -3.1/n$ and $1/en \leq \sigma^2 \leq 3.1/n$ leads to

$$\begin{aligned} \mathrm{E}\left(Z_0\right) \cdot e^{-\frac{1}{en}t} \leq \mathrm{E}\left(Z_t\right) \leq\ & \mathrm{E}\left(Z_0\right) \cdot e^{-\frac{3.1}{n}t} \\ \mathrm{Var}\left(Z_t\right) \geq\ & \mathrm{E}\left(Z_0\right) \cdot e^{-\frac{t}{en}} \left(1 - e^{-\frac{t}{en}}\right) \\ \mathrm{Var}\left(Z_t\right) \leq\ & \mathrm{E}\left(Z_0\right) \cdot e^{-3.1\frac{t}{n}} \left(1 - e^{-3.1\frac{t}{n}}\right). \end{aligned}$$

Finally translating to the number of ones $X_t = n - Z_t$ leads to the claimed results. □

4.3 Non-Elitist Algorithms on ONEMAX

In this subsection we will consider algorithms with local mutation but different selection operators. We keep general the acceptance probability, with the only assumption that it is monotonically non-decreasing. Afterwards, in the validation section we will plug in the expressions for a RW, MA and SSWM.

Lemma 14. *Consider any Algorithm 1 with a monotonically non-decreasing acceptance function $p_{\mathrm{acc}}(\Delta f)$ on* ONEMAX. *Then,*

$$\begin{aligned} E\left(X_{t+1} - X_t \mid X_t\right) &= \frac{p^{\uparrow} + p^{\downarrow}}{n} \cdot \left(n \frac{p^{\uparrow}}{p^{\uparrow} + p^{\downarrow}} - X_t\right) \\ E\left(\left(X_{t+1} - X_t\right)^2 \mid X_t\right) &= p^{\uparrow} - \frac{X_t}{n}\left(p^{\uparrow} - p^{\downarrow}\right). \end{aligned}$$

where X_t denotes the number of 1*-bits in the search point at time t, $p^{\uparrow} = p_{\mathrm{acc}}(\Delta f = 1)$ and $p^{\downarrow} = p_{\mathrm{acc}}(\Delta f = -1)$.*

Proof. Let us start from $\mathrm{E}\left(X_{t+1} - X_t \mid X_t\right) = \sum_x x \cdot p(X_{t+1} - X_t = x \mid X_t)$. Now we use the same arguments as for the analysis of RLS on ONEMAX (see the proof of Application 12) but considering the acceptance of deleterious mutations. Then,

$$\begin{aligned} \mathrm{E}\left(X_{t+1} - X_t \mid X_t\right) &= \frac{n - X_t}{n} p^{\uparrow} - \frac{X_t}{n} p^{\downarrow} = p^{\uparrow} - \frac{X_t}{n}\left(p^{\uparrow} + p^{\downarrow}\right) \\ &= \frac{p^{\uparrow} + p^{\downarrow}}{n} \cdot \left(n \frac{p^{\uparrow}}{p^{\uparrow} + p^{\downarrow}} - X_t\right), \end{aligned}$$

where we have multiplied and divided by $\left(p^{\uparrow} + p^{\downarrow}\right)/n$. For the diffusion coefficient, we use the definition of the second statistical moment $\mathrm{E}\left(\left(X_{t+1} - X_t\right)^2 \mid X_t\right) = \sum_x x^2 \cdot p(X_{t+1} - X_t = x \mid X_t)$. Using the same rationale as for the drift coefficient but noticing that the second moment can only change by $(\pm 1)^2$ yields

$$\mathrm{E}\left(\left(X_{t+1} - X_t\right)^2 \mid X_t\right) = (+1)^2 \cdot \frac{n - X_t}{n} p^{\uparrow} + (-1)^2 \cdot \frac{X_t}{n} p^{\downarrow} = p^{\uparrow} - \frac{X_t}{n}\left(p^{\uparrow} - p^{\downarrow}\right).$$

□

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Once the coefficients are known, the following application derives the fixed budget results by using the non-elitist multiplicative drift (Theorem 8).

Application 15. *Consider any Algorithm 1 with a monotonically non-decreasing acceptance function $p_{\text{acc}}(\Delta f)$ on* ONEMAX*. Under the Diffusion Approximation the following statements hold*

$$E\left(X_t \mid X_0\right) = X_0 e^{-\frac{p^\uparrow + p^\downarrow}{n} t} + n\frac{p^\uparrow}{p^\uparrow + p^\downarrow}\left(1 - e^{-\frac{p^\uparrow + p^\downarrow}{n} t}\right)$$

$$Var\left(X_t \mid X_0\right) = \frac{np^\uparrow}{2(p^\uparrow + p^\downarrow)}\left(1 - e^{-2t\frac{p^\uparrow + p^\downarrow}{n}}\right) - \frac{p^\uparrow - p^\downarrow}{p^\uparrow + p^\downarrow} X_0 e^{-\frac{p^\uparrow + p^\downarrow}{n} t}\left(1 - e^{-\frac{p^\uparrow + p^\downarrow}{n} t}\right) - \frac{p^\uparrow(p^\uparrow - p^\downarrow)}{2n}\left(1 - e^{-\frac{p^\uparrow + p^\downarrow}{n}{}_l t}\right)^2,$$

where X_t denotes the number of 1*-bits in the search point at time t, $p^\uparrow = p_{\text{acc}}(\Delta f = 1)$ and $p^\downarrow = p_{\text{acc}}(\Delta f = -1)$.*

Proof. Assume that Hypothesis 1 is a good approximation for the dynamics of X_t. Then, the statement is proven after a direct application of Theorem 8 with the coefficients from Lemma 14.

$$a = \frac{p^\uparrow + p^\downarrow}{n} \qquad b = n\frac{p^\uparrow}{p^\uparrow + p^\downarrow}$$
$$\alpha = p^\uparrow \qquad \beta = \frac{p^\uparrow - p^\downarrow}{n}. \tag{14}$$

□

5 Validation of the Diffusion Approximation

The previous section showed several case studies of the application of the three rigorous drift theorems for fixed budget analysis. However this method was the result of an approximation which we were assuming to be accurate (Hypothesis 1). Finally, we provide an extensive validation to this assumption. We will tackle the accuracy of the method from several angles including experiments but also comparing the results against those obtained by rigorous methods.

5.1 Comparison with the Literature

We now compare the results obtained in subsections 4.1 and 4.2 with some of the fixed budget literature. For the comparison on the LEADINGONES problem, we use the paper that presented the first fixed budget analysis (Jansen and Zarges, 2012). In Table 2, we observe a high agreement for the upper bounds estimations (even an exact match for the RLS). Both approaches obtained the same growth term of $2t/n$ and the discrepancies for the (1+1) EA are present only on small terms. However, on the lower bounds the discrepancy is bigger, our method only obtained a growth term of t/n since we had to pessimistically use the smallest drift value (when $X_t = n - 1$) to be able to apply the additive drift (recall the proof of Application 10). On the other hand, the results from the literature only hold for time budget values which are sufficiently smaller that the expected optimisation, whereas our results hold until the optimum is found.

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Table 2: Comparison of results for LEADINGONES. The *literature* column is obtained from Theorems 7 and 8 in Jansen and Zarges (2012). The results for RLS hold for $t = (1-\beta)n^2$ with $1/2 + c < \beta < 1$. In the case of the (1+1) EA, the result holds for $t = \frac{(1-\beta)n^2}{\alpha(n)}$ with $1/2 + c < \beta < 1$ and $\alpha(n) = w(1)$. The *diffusion approximation* column corresponds with Applications 10 and 11 and hold for $t < \min\{t \geq 0 \mid X_t = n\}$. Random initialisation was considered in all the cases: $\mathrm{E}(X_0) = 1 - 2^{-n}$.

Algorithm	$\mathrm{E}(X_t)$	Literature	Diffusion Approximation
RLS	$\geq$	$1 + 2t/n - 2^{-\Omega((1-\beta)n)}$	$1 + t/n - 2^{-n}$
	$\leq$	$1 + 2t/n - 2^{-n}$	$1 + 2t/n - 2^{-n}$
(1+1) EA	$\geq$	$1 + 2t/n - o(t/n)$	$1 + t/(en) - 2^{-n}$
	$\leq$	$1 + 2t/n - o(t/n)$	$1 + 2t/n - 2^{-n}$

In Table 3 we consider the ONEMAX problem. Again, for RLS we compare against Jansen and Zarges (2012), we observe how both results (first two rows) can be recovered from our method noticing that $e^{-t/n} = (e^{-1/n})^t \approx (1 - 1/n)^t$, which is a good approximation for large n. In the case of the (1+1) EA we were able to obtain exactly the same lower bound as Lengler and Spooner (2015). The upper bounds are also in high agreement, however the constant inside the exponential term is not as tight as in the literature.

Table 3: Comparison of results for ONEMAX. The *literature* column is obtained from Theorems 4 and 5 in Jansen and Zarges (2012) and Theorem 5 in Lengler and Spooner (2015). The *diffusion approximation* column corresponds with Applications 12 and 13. Random initialisation was considered in all the cases apart from the first row ($X_0 = 0^n$).

Algorithm	$\mathrm{E}(X_t)$	Literature	Diffusion Approximation
RLS	$=$	$n\left(1 - (1 - 1/n)^t\right)$	$n\left(1 - e^{-t/n}\right)$
	$=$	$n - n/2\,(1 - 1/n)^t$	$n - n/2e^{-t/n}$
(1+1) EA	$\geq$	$n\left(1 - \frac{1}{2}e^{-t/(en)}\right)$	$n\left(1 - \frac{1}{2}e^{-t/(en)}\right)$
	$\leq$	$n\left(1 - \frac{1}{2}e^{-t/n}\right) - O(t/n)$	$n\left(1 - \frac{1}{2}e^{-(3.1t)/n}\right)$

Finally, since to our best knowledge, there are no previous fixed budget analysis of non-elitist algorithm we could not perform this sanity check for RW, MA and SSWM. Nonetheless, the following subsections circumvents this problem by recurring to the concept of stationary distribution.

5.2 Stationary Distribution

One feature of non-elitist EAs is that although they can sample the optimum, they might leave it and move to a search point of lower fitness. Actually, in the long run the algorithm (under certain conditions) will reach the so-called *stationary distribution* $\pi(x)$ for when the probabilities of leaving each state cancel out with the probabilities of reaching that state (see e.g. Ross, 1996)

$$\pi(x) \cdot P(x \to y) = \pi(y) \cdot P(y \to x) \quad \forall x, y. \tag{15}$$

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Once the process reaches this limit state all the points in the search space will have a non-zero probability of being found when measuring the system. However a smart algorithm will assign higher probability masses to points of higher fitness.

It is straightforward to notice that elitist algorithms do not have a stationary distribution. This is due to the fact that these algorithms always accept moves towards points of higher fitness but never accept moves towards points of lower fitness. This breaks the ergodicity required for a Markov Chain to have a stationary distribution. Hence, this sanity check for our method can only be applied for a RW, MA and SSWM.

If we consider a RW with local mutations, it will follow the mutational bias towards bitstrings with $n/2$ zeros. Then, after enough time depending on the initialisation, the expected number of ones will be $n/2$. In the case of MA and SSWM, we find in the literature that they share the following stationary distribution (see Metropolis et al., 1953; Sella and Hirsh, 2005 or for a proof Nallaperuma et al., 2017b).

$$\pi(x) = \frac{e^{\gamma f(x)}}{Z} \tag{16}$$

where $\gamma = \alpha$ for Metropolis, $\gamma = 2\beta(N-1)$ for SSWM and Z is the normalising constant.

Theorem 16. *Consider Metropolis and SSWM on* ONEMAX *once they have reached the stationary distribution. Then for any initialisation, the expected number of ones will be* $\mathrm{E}(X_t) = \frac{n}{1+e^{-\gamma}}$*, where* $\gamma = \alpha$ *for Metropolis and* $\gamma = 2\beta(N-1)$ *for SSWM.*

Proof. In the case of ONEMAX we can express $\mathrm{E}(X_t)$ with the probabilities of each bit i being in each of the two possible values $p_i(1)$ and $p_i(0)$,

$$\mathrm{E}(X_t) = \sum_{i=1}^{n} \frac{p_i(1)}{p_i(1) + p_i(0)} = n \cdot \frac{p(1)}{p(1) + p(0)},$$

where in the last equality we have used the fact that each bit is optimised independently, thus we can apply linearity of expectations and consider that for any bit $p_i(1) = p(1)$ and $p_i(0) = p(0)$. Finally, introducing the probabilities at equilibrium from equation (16) yields

$$\mathrm{E}(X_t) = n \cdot \frac{e^{\gamma}}{e^{\gamma} + 1} = \frac{n}{1 + e^{-\gamma}}.\square \tag{17}$$

As discussed earlier, the long term (limit for $t \to \infty$) expectation of the non-elitist multiplicative drift tends to the b term from the drift coefficient which according to Equation (14) has a value of $b = (np^{\uparrow})/(p^{\uparrow} + p^{\downarrow})$. We can observe in Table 4 how this value matches exactly the real expected fitness at equilibrium $\mathrm{E}(X_t)$ showing the strength of the SDE approach for this scenario.

Table 4: Expected value at equilibrium $\mathrm{E}(X_t)$ from Theorem 16 and long term expectation b from Application 15 on ONEMAX. With $p^{\uparrow} = p_{\mathrm{acc}}(1)$ and $p^{\downarrow} = p_{\mathrm{acc}}(-1)$.

	RW	**Metropolis**	**SSWM**
$p^{\uparrow}$	1	1	$\frac{1-e^{-2\beta}}{1-e^{-2N\beta}}$
$p^{\downarrow}$	1	$e^{-\alpha}$	$\frac{e^{2\beta}-1}{e^{2N\beta}-1}$
b	$n/2$	$\frac{n}{1+e^{-\alpha}}$	$\frac{n}{1+e^{-2\beta(N-1)}}$
$\mathrm{E}(X_t)$	$n/2$	$\frac{n}{1+e^{-\alpha}}$	$\frac{n}{1+e^{-2\beta(N-1)}}$

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

5.3 Reconciling Fixed Budget with Runtime Analysis

As argued by Jansen and Zarges (2012), the fixed budget perspective provides more insight on the optimisation process. Runtime analysis focuses on the expected first hitting time of the optimum, this time is expressed as a function of the problem size $\mathrm{E}(T) = f(n)$. On the other hand, fixed budget considers the expected fitness obtained after t iterations. Hence, the result will also depend on the time $\mathrm{E}(X_t) = f(n,t)$.

However, both approaches should be in concordance. If we know the first hitting time $T(X)$ of any specific fitness value X, it looks natural to compute the inverse function to obtain the equivalent fixed budget result. As shown by Doerr et al. (2013) this is the case for deterministic algorithms, but it will not be true in general for randomised algorithms. The authors circumvent this problem by using sharp bounds on the probability of deviating from the expected times. Finally, they developed a method to recycle expected optimisation times to derive fixed budget results.

In this section, we consider the inverse approach i.e. deriving optimisation times from fixed budget results. Again, inverting the function $\mathrm{E}(X_t) = f(n,t)$ for the time will not give, in general, the correct answer. However, we do not aim for a rigorous translation tool as achieved by Doerr et al. (2013). Our purpose is to obtain some time measurement that can be compared with the runtime literature (up to the use of Landau's notation).

Additive Drift

Before jumping into conclusions for the studied algorithms, let us consider an additive process. This is, a collection of stochastic variables in time $X_{t\geq 0} \in \mathbb{R}$ whose expected change remains constant $\mathrm{E}(X_t - X_{t-1} \mid X_t) = b$. When b is positive, the expectation $\mathrm{E}(X_t)$ is constantly being increased and $\mathrm{E}(X_t)$ always reaches any finite target value x (given enough, but finite time). Let us recall that this will not be the case for EAs reaching the optimum x_{opt} of a fitness function, here $X_t \leq x_{\mathrm{opt}}$ and therefore $\mathrm{Prob}(\mathrm{X_t} = \mathrm{x_{opt}}) = 1$ is required in order for $\mathrm{E}(X_t) = x_{\mathrm{opt}}$. We will consider first a pure additive process and afterwards we will relate it with RLS and the (1+1) EA optimising LeadingOnes.

We mentioned before that computing the inverse function of $\mathrm{E}(X_t)$ for the time t will not yield, in general, the correct result. However, this approach is correct for an additive drift process. Here, we can solve $\mathrm{E}(X_t) = f(n,t)$ for the time, obtaining a function of the form $t = f(\mathrm{E}(X), n)$. Finally, if we introduce any target value for $\mathrm{E}(X)$, it will yield the time when the expectation reaches the target value.

Theorem 17. *Consider a stochastic process $X_{t\geq 0} \in \mathbb{R}$ that follows an Additive Drift i.e. $E(X_t - X_{t-1} \mid X_t) = b$ for some constant $b > 0$. The time $\tau(x)$ is defined as the time when the expectation $E(X_t)$ reaches a value of x and is given by*

$$\tau(x) = \frac{x - E(X_0)}{b}.$$

Proof. By using the additive drift condition, we can compute $\mathrm{E}(X_t \mid X_0)$ as follows

$$\mathrm{E}(X_t \mid X_0) = X_0 + \sum_{i=0}^{t} \mathrm{E}(X_{i+1} - X_i \mid X_i) = X_0 + \sum_{i=0}^{t} b = X_0 + bt.$$

Using the law of total expectation, we obtain $\mathrm{E}(X_t) = \mathrm{E}(X_0) + bt$. Introducing $\mathrm{E}(X_t) = x$ and $t = \tau$ leads to $x = \mathrm{E}(X_0) + b\tau$, which solving for τ yields the claimed statement. □

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

It is straightforward to notice that this theorem is not applicable for optimisation algorithms. If X_t describes the fitness of the EA, we observe that the boundary condition $X_t \leq x_{\text{opt}}$ breaks the additive requirement: $\mathrm{E}\left(X_t - X_{t-1} \mid X_t = x_{\text{opt}}\right) \leq 0 \neq b$. As we mentioned earlier, we do not aim for a rigorous translation tool. However, as shown in the subsection 4.1 we can approximate the optimisation process of some algorithms on LEADINGONES with additive drift processes.

In Applications 10 and 11, we found additive drift bounds for the real drift of RLS and the (1+1) EA on LEADINGONES respectively. By introducing $x = n$ (the optimal fitness of LEADINGONES) in Theorem 17, we will compare, in a non-rigorous way, the $\tau(n)$-times for these processes with the runtimes from the literature. The following table presents these results showing a high agreement, up to smaller terms, of the SDE method with the literature. Since we used bounds for the drift coefficient b, the following table includes and upper τ_u and a lower τ_ℓ bound for the time $\tau(n)$.

Table 5: Expected optimisation times $\mathrm{E}(T)$ from the literature (see e.g. Oliveto and Yao, 2011) and bounds on $\tau(n)$ from Theorem 17. The bounds on the drift $b_{u,\ell}$ are taken from Equation (11) (RLS) and Equations (13) and (12) (lower and upper bound of the (1+1) EA respectively). Initialisation at 0^n.

LEADINGONES	**RLS**	**(1+1) EA**
b_ℓ	$1/n$	$1/en$
b_u	$2/n$	$2/n$
$\mathrm{E}(X_0)$	0	0
$\tau_\ell(n)$	$n^2/2$	$n^2/2$
$\tau_u(n)$	n^2	en^2
$\mathrm{E}(T)$	$\Theta(n^2)$	$\Theta(n^2)$

Multiplicative Drift

In the case of the multiplicative drift (Theorems 6 and 8) if we were to perform the same sanity check we would obtain an infinite time. This is due to the exponential decay of $\mathrm{E}(X_t)$ through the term e^{-t}. However, we can circumvent this problem by recurring to the concept of half-life from exponential decaying radioactive processes (see e.g. Krane, 1987). The half-life is defined as the time taken until half of the atoms from a radionuclide have followed a decay process. In the same spirit, the following definition introduces the notion of δ-life.

Definition 4 (δ-life)**.** *Consider a stochastic process $X_t > 0$ whose expectation follows an exponential decay i.e. $dE(X_t)/dt = -aE(X_t)$ for $a \in \mathbb{R}^+$. The δ-life namely τ_δ is defined as the time when $E(X_t)/E(X_0) = \delta$ and has a value of*

$$\tau_\delta = \frac{\ln(1/\delta)}{a}.$$

As used in subsection 4, we will focus on the number of 1-bits on linear functions, then we establish our satisfiability criterion as $\delta = 1/n$. This implies that in the worst case, where the system starts at a distance n from the long term expectation we will be satisfied when there is only a difference of one 1-bit with the optimal solution.

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Theorem 18. *The $1/n$-life of the Non-Elitist Multiplicative Drift (Theorem 8) is given by*

$$\tau_{1/n} \leq \frac{n \ln n}{p^{\uparrow}}.$$

Proof. The result for the expectation from Theorem 8 can be rewritten as $\mathrm{E}(X_t) - b = (\mathrm{E}(X_0) - b) \cdot e^{-at}$. It is straightforward to see that the process $\mathrm{E}(X_t) - b$ follows an exponential decay, then using Definition 4 for $\delta = 1/n$ leads to

$$\tau_{1/n} = \frac{\ln n}{a},$$

introducing the a coefficient from equation (14) completes the proof.

$$\tau_{1/n} = \frac{n \ln n}{p^{\uparrow} + p^{\downarrow}} \leq \frac{n \ln n}{p^{\uparrow}}.\square$$

The following table includes the $1/n$-lives ($\tau_{1/n}$) derived from the SDE method with the above expression and expected optimisation times $\mathrm{E}(T)$ from the literature. Before making the comparison let us remember that optimisation times refer to hitting the optimum whereas the $1/n$-life refers to approaching equilibrium. If the parameters of the algorithm allow the stationary distribution, to be close enough to the optimum this will be a fair comparison. This is the reason why the results for the RW highly differ in the table, a RW can not optimise ONEMAX in polynomial time but can reach its stationary distribution. On the other hand the results for the three other algorithms are in high concordance with the literature (up to smaller terms).

Table 6: $\tau_{1/n}$ from Theorem 18 and expected optimisation times $\mathrm{E}(T)$ on ONEMAX (RLS - e.g. Oliveto and Yao (2011), RW - Garnier et al. (1999), Metropolis - Jansen and Wegener (2007) and SSWM - Pérez Heredia et al., 2017). Where $p^{\uparrow} = p_{\mathrm{acc}}(1)$, $\gamma = \alpha$ for Metropolis and $\gamma = 2\beta(N-1)$ for SSWM.

ONEMAX	**RLS**	**RW**	**Metropolis**	**SSWM**
$p^{\uparrow}$	1	1	1	$\frac{1-e^{-2\beta}}{1-e^{-2N\beta}}$
$\tau_{1/n}$	$n \ln n$	$n \ln n$	$n \ln n$	$\frac{n \ln n}{p^{\uparrow}}$
$\mathrm{E}(T)$	$n \ln n + O(n)$	$\Omega(2^n)$	$n \ln n + O(n)$	$\frac{n \ln n}{p^{\uparrow}} + O(n)$
γ	-	-	$\geq \ln n$	$\geq \ln n$

5.4 Simulations

Another sanity-check that we have considered to validate Hypothesis 1 is to perform numerical simulations. We built several graphical representations where the experimental results are represented by: solid lines (expectation) and a shadowed region (one standard deviation around the expectation). On the other hand, theoretical results for the expectation are represented by dashed lines. For aesthetics reasons we have not included the variance estimations in the figures. This would have implied plotting two extra shadowed zones (upper and lower bounds estimations) per algorithm.

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Firstly, the left graph from Figure 2 shows the application of the additive drift for LEADINGONES. We can observe that for RLS this approach is very precise until the last part of the optimisation process. Whereas for the (1+1) EA there is a large gap, nevertheless the method properly represents the evolution of the (1+1) EA's fitness on this problem. In our opinion, the discrepancy gap is mainly due to two reasons:

- choosing an additive drift approach for a process whose drift varies depending on the state and
- pessimistic estimations for the additive constant b (see Theorem 4 and Applications 10 and 11).

This deviation from additivity can be seen at the end of the simulation when the experimental lines start curving, whereas the theoretical lines remain straight until they hit the optimum.

Secondly, the right graph from Figure 2 compares RLS and the (1+1) EA on the ONEMAX problem. The comparison between experiments and theory for RLS is excellent, even better than in LEADINGONES due to the absence of the boundary condition. Again for the (1+1) EA the theory collects the exponential decrease of the process. However there is still a gap which in our opinion is mainly due to equivalent reasons:

- using a multiplicative drift when the drift is not exactly multiplicative and
- pessimistic estimations for the multiplicative constant b (see Theorem 6 and Application 13).

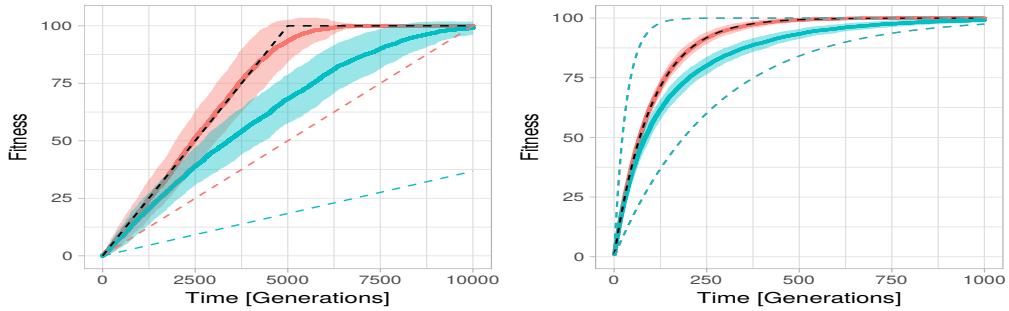

Figure 2: Comparison of results for LEADINGONES (left) and ONEMAX (right). Solid lines represent the experimental results (RLS - red and (1+1) EA- blue). Dashed lines represent the theoretical results (RLS - black and (1+1) EA- blue). For LEADINGONES, the theoretical upper bound is shared for both algorithms and the lower bound for RLS is shown in red. Initialisation at 0^n. Problem size $n = 100$. Time budget 10000 (LEADINGONES) and 1000 (ONEMAX). Results were averaged over 100 runs and the shadowed zones include one standard deviation around the expectation.

The following figures show the application of the non-elitist drift for ONEMAX. The left graph from Figure 3 shows an excellent match between theoretical and experimental results for all the algorithms, describing the dynamics of these processes even when they are not able to optimise the problem. With the same accuracy the right graph of Figure 3 shows a case study where two algorithms (Metropolis and RW) started in between their stationary expected value and the optimum but due to their weak selection strength they move away the optimum but towards the equilibrium's value.

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

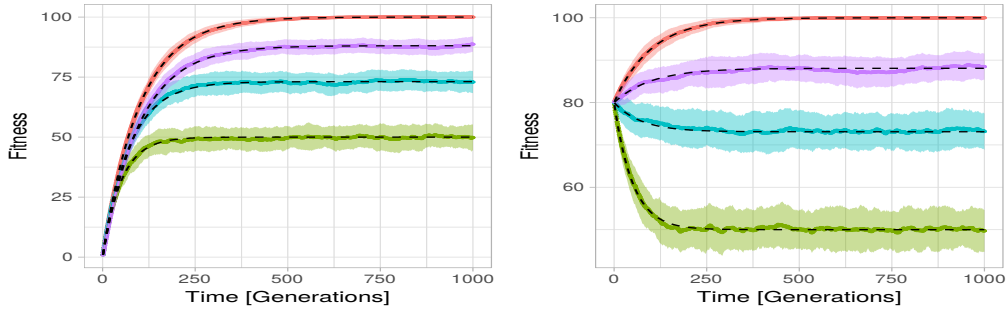

Figure 3: Theoretical (dashed black lines) and experimental results (RLS - red, RW - green, Metropolis - blue and SSWM - pink) on ONEMAX. Initialisation at 0^n (left) and $0^{20}1^{80}$ (right). Problem size $n = 100$. Time budget 1000. Algorithms' parameters: $\alpha = 1$ for Metropolis and $(\beta, N) = (1, 2)$ for SSWM. Results were averaged over 100 runs and the shadowed zone includes one standard deviation around the expectation.

We would like to point out that our applications were just study cases to motivate the use of SDEs as an analysis tool for EAs. If we were aiming for precision we could have recycled already known and better bounds for the drift such us $b(Z_t) \leq -\frac{Z_t}{en}\left(1 + \frac{16Z_t}{n}\right)$ from Doerr et al. (2011), but then it would become a variable drift process (Johannsen, 2010).

5.5 Experimental Error

The previous sanity-check was based on a visual representation of experimental and theoretical results. However, graphical representations can sometimes be misleading and we have to be sure that the results obtained were not just an aesthetic effect. In order to objectively validate the SDE method (Hypothesis 1) we perform an experimental measurement of the error. We will use the widely used concepts of mean absolute percentage error (MAPE) and root mean squared error (RMSE) (see e.g. Hyndman and Koehler, 2006).

Definition 5. *Let $Y_{i=1,2,\ldots,n}$ be the vector of observed values and $\overline{Y}_{i=1,2,\ldots,n}$ the vector of predicted values. Then, the mean absolute percentage error (MAPE) and the root mean squared error (RMSE) are given by*

$$\mathrm{MAPE}(Y_i) = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\overline{Y_i} - Y_i}{Y_i}\right|, \qquad \mathrm{RMSE}(Y_i) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\overline{Y_i} - Y_i\right)^2}.$$

The advantage of the MAPE is that it takes into consideration the magnitude of the observed values. It produces a result in the form of a percentage that is interpreted as the magnitude of the deviation with reality. However, for small values of the observed values this measurement is not appropriate due to the singularity at $Y_i = 0$. This is a problem for measuring the error of the variance due to the deterministic initialisation used in the experiments and the concentration of the independent runs towards the same fitness value at the end of the simulation. For this reason, we have decided to use the MAPE for the expectations and the RMSE for the variance.

The following tables show agreement with Figures 2 and 3. The results for all the algorithms with local mutations, i.e., all but the (1+1) EA, are excellent on ONEMAX. With MAPE of the expectation being even smaller than 1%. Only for the (1+1) EA we can see a significant difference, due to the reasons explained in the previous section.

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Table 7: Experimental errors for the expectation and variance of the five studied algorithms on ONEMAX. The observed values are obtained from experiments. The predicted values are those derived in applications 12, 13 and 15. Algorithms' parameters: $\alpha = 1$ for Metropolis and $(\beta, N) = (1, 2)$ for SSWM. Initialisation at 0^n. Problem size $n = 100$. Time budget $100$0.

ONEMAX	MAPE(E (X_t))	RMSE(Var (X_t))
RLS	0.15%	0.41
(1+1) EA lower bound	16.34%	0.71
(1+1) EA upper bound	22.18%	1.30
RW	0.88%	0.42
MA	0.62%	0.44
SSWM	0.45%	0.37

Finally, the comparison against LEADINGONES (Table 8) is not as successful as in the previous case. Here, the errors are quite large as shown in the left graph of Figure 2. But again, we think that the reasons for the discrepancy are not intrinsic to using SDEs, they are due to the chosen drift technique.

Table 8: Experimental errors for the expectation and variance of RLS and the (1+1) EA on LEADINGONES. The observed values are obtained from experiments. The predicted values are those derived in applications 10 and 11. Initialisation at 0^n. Problem size $n = 100$. Time budget 10000.

LEADINGONES	MAPE(E (X_t))	RMSE(Var (X_t))
RLS lower bound	47.90%	7.82
RLS upper bound	3.39%	7.73
(1+1) EA lower bound	72.27%	8.03
(1+1) EA upper bound	23.74%	7.93

6 Conclusions

For the last decade rigorous runtime analysis has become immensely popular, in part due to the emergence of the so-called "drift theorems" that inform about the time to reach a particular state or set of states. However, these results are not directly translatable for fixed budget scenarios which ask about the opposite question: how much improvement can be obtained in a fixed number of fitness evaluations. This problem is relevant in practice: in real problems it is usually impossible to know whether the optimum has been found. In this case, it is arguably more useful to know how much improvement an algorithm can achieve in a fixed number of steps.

Here, we have introduced the use of SDEs to the study of EAs which seem particularly suited to obtain results for fixed budget scenarios. Even though SDEs are approximations of the dynamics of EAs, they can produce analytical insight which is not possible with other tools, such as Markov chains. At the same time, SDEs do not

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

discard the stochasticity of the dynamics and allow for the estimation of the variance of relevant quantities, such as the state at which the algorithm will find itself after t iterations.

Here we made a simple preliminary exploration of the potential of these tools for the study of EAs. We did not make use of other techniques for the study of SDEs that allow for greater insight into the dynamics of a stochastic process. For example, the Fokker-Planck equation associated with a SDE can be used to track the full probability distribution of the process. Even though this is, in many instances, impossible to solve analytically it can reveal what are the crucial parameters of a stochastic process. Furthermore, it is often solvable for the steady state, allowing for the characterization of the stationary distribution of the process. In fact, these techniques are commonly used in the related field of population genetics where they have been used to provide foundational results. The results we presented here are in themselves interesting, especially when applied to fixed budget analysis, but they also hold the promise that SDEs can become a standard tool in EC.

On the other hand, it is fair to say that previous modelling attempts (those mentioned in the introduction) were aiming for more general and ambitious problems than what this paper has covered. This is why we believe that the natural direction for future work will be considering more complex functions or more elaborated heuristics. But the main challenge would be a rigorous translation of the SDE method presented here.

7 Acknowledgments

The author would like to thank Dirk Sudholt and Tiago Paixão for valuable input and discussions on this manuscript. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 618091 (SAGE).

References

Akimoto, Y., Auger, A., and Hansen, N. (2012). Convergence of the continuous time trajectories of isotropic evolution strategies on monotonic $\mathcal{C}^2$-composite functions. In *Parallel Problem Solving from Nature - PPSN XII: 12th International Conference, Taormina, Italy, September 1-5, 2012, Proceedings, Part I*, pages 42–51. Springer Berlin Heidelberg.

Altenberg, L. (1994). The schema theorem and price's theorem. In *Proceedings of the Third Workshop on Foundations of Genetic Algorithms. Estes Park, Colorado, USA, July 31 - August 2 1994*, pages 23–49.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654.

Cox, J. C., Ingersoll, J. E., and Ross, S. A. (1985). A Theory of the Term Structure of Interest Rates. *Econometrica*, 53(2):385–407.

Doerr, B., Doerr, C., and Yang, J. (2016). Optimal parameter choices via precise black-box analysis. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, pages 1123–1130, New York, NY, USA. ACM.

Doerr, B., Fouz, M., and Witt, C. (2011). Sharp bounds by probability-generating functions and variable drift. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pages 2083–2090, New York, NY, USA. ACM.

Doerr, B., Jansen, T., Witt, C., and Zarges, C. (2013). A method to derive fixed budget results from expected optimisation times. In *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation (GECCO '13)*, pages 1581–1588. ACM.

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Doerr, B., Johannsen, D., and Winzen, C. (2012). Multiplicative drift analysis. *Algorithmica*, 64(4):673–697.

Einstein, A. (1905). Über die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen. *Annalen der Physik*, 322(8):549–560.

Feller, W. (1949). On the theory of stochastic processes, with particular reference to applications. In *Proceedings of the [First] Berkeley Symposium on Mathematical Statistics and Probability*, pages 403–432, Berkeley, Calif. University of California Press.

Garnier, J., Kallel, L., and Schoenauer, M. (1999). Rigorous hitting times for binary mutations.

Goldberg, D. (1987). Simple genetic algorithms and the minimal, deceptive problem. In Davis, L., editor, *Genetic algorithms and simulated annealing*, pages 74–88.

He, J. and Yao, X. (2001). Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 127:57–85.

He, J. and Yao, X. (2003). Towards an analytic framework for analysing the computation time of evolutionary algorithms. *Artificial Intelligence*, 145(1):59 – 97.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA. Reprint edition 1992 (originally published in 1975).

Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679 – 688.

Itô, K. (1944). Stochastic integral. In *Proceedings of the Imperial Academy*.

Itô, K. (1946). On a stochastic integral equation. In *Proceedings of the Japan Academy*.

Itô, K. (1950). Stochastic differential equations in a differentiable manifold. *Nagoya Mathematical Journal*.

Itô, K. (1951). Diffusion processes and their sample paths. *Nagoya Mathematical Journal*.

Jansen, T. and Wegener, I. (2007). A comparison of simulated annealing with a simple evolutionary algorithm on pseudo-boolean functions of unitation. *Theor. Comput. Sci.*, 386(1-2):73–93.

Jansen, T. and Zarges, C. (2012). Fixed budget computations: a different perspective on run time analysis. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '12)*, pages 1325–1332. ACM.

Johannsen, D. (2010). *Random Combinatorial Structures and Randomized Search Heuristics*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany and the Max-Planck-Institut für Informatik.

Karlin, S. and Taylor, H. M. (1981). *A Second Course in Stochastic Processes*. Academic Press, New York, 1 edition edition.

Kimura, M. (1957). Some Problems of Stochastic Processes in Genetics. *The Annals of Mathematical Statistics*, 28(4):882–901.

Kimura, M. (1962). On the Probability of Fixation of Mutant Genes in a Population. *Genetics*, 47(6):713–719.

Kimura, M. (1964). Diffusion Models in Population Genetics. *Journal of Applied Probabilitiy*, 1(2):177–232.

Kimura, M. (1968). Genetic Variability Maintained in a Finite Population Due to Mutational Production of Neutral and Nearly Neutral Isoalleles. *Genetics Research*, 11(03):247–270.

Krane, K. S. (1987). *Introductory nuclear physics*. Wiley, New York.

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Lehre, P. K. and Witt, C. (2012). Black-box search by unbiased variation. *Algorithmica*, 64(4):623–642.

Lengler, J. and Spooner, N. (2015). Fixed budget performance of the (1+1) EA on linear functions. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, FOGA '15, pages 52–61, New York, NY, USA. ACM.

Maghsoodi, Y. (1996). Solution of the Extended Cir Term Structure and Bond Option Valuation. *Mathematical Finance*, 6(1):89–109.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Nallaperuma, S., Neumann, F., and Sudholt, D. (2017a). Expected fitness gains of randomized search heuristics for the traveling salesperson problem. *Evolutionary Computation*, pages 1–33. To appear.

Nallaperuma, S., Oliveto, P. S., Heredia, J. P., and Sudholt, D. (2017b). When is it beneficial to reject improvements? In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 1391–1398, New York, NY, USA. ACM.

Øksendal, B. (2003). *Stochastic Differential Equations: An Introduction with Applications*. University of Michigan Press.

Oliveto, P. S., Paixão, T., Pérez Heredia, J., Sudholt, D., and Trubenová, B. (2017). How to escape local optima in black box optimisation: When non-elitism outperforms elitism. *Algorithmica*. To appear.

Oliveto, P. S. and Yao, X. (2011). Runtime analysis of evolutionary algorithms for discrete optimization. In *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. World Scientific Publishing Co., Inc.

Paixão, T. and Pérez Heredia, J. (2017). An application of stochastic differential equations to evolutionary algorithms. In *Proceedings of the 14th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*, FOGA '17, pages 3–11, New York, NY, USA. ACM.

Paixão, T., Pérez Heredia, J., Sudholt, D., and Trubenová, B. (2017). Towards a runtime comparison of natural and artificial evolution. *Algorithmica*, 78(2):681–713.

Paixo, T., Badkobeh, G., Barton, N., r, D., Dang, D.-C., Friedrich, T., Lehre, P. K., Sudholt, D., Sutton, A. M., and Trubenov, B. (2015). Toward a unifying framework for evolutionary processes. *Journal of Theoretical Biology*, 383:28 – 43.

Pérez Heredia, J., Trubenová, B., Sudholt, D., and Paixão, T. (2017). Selection limits to adaptive walks on correlated landscapes. *Genetics*, 205(2):803–825.

Prügel-Bennett, A. (1997). Modelling evolving populations. *Journal of Theoretical Biology*, 185(1):81–95.

Prügel-Bennett, A. and Shapiro, J. L. (1994). Analysis of genetic algorithms using statistical mechanics. *Physical Review Letters*, 72(9):1305.

Ross, S. (1996). *Stochastic Processes*. John Wiley & Sons.

Schaul, T. (2012). Natural evolution strategies converge on sphere functions. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, pages 329–336, New York, NY, USA. ACM.

Sella, G. and Hirsh, A. E. (2005). The application of statistical physics to evolutionary biology. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):95419546.

Evolutionary Computation Just Accepted MS.
https://doi.org/10.1162/evco_a_00216

Stieltjes, T. J. (1894). Recherches sur les fractions continues. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, pages 68–71.

Stratonovich, R. (1966). A new representation for stochastic integrals and equations. *SIAM Journal on Control*.

Vose, M. D. (1995). Modeling simple genetic algorithms. *Evol. Comput.*, 3(4):453–472.

Wormald, N. C. (1995). Differential equations for random processes and random graphs. *Ann. Appl. Probab.*, 5(4):1217–1235.

Wright, A. H. and Rowe, J. E. (2001). Continuous dynamical system models of steady-state genetic algorithms. In Martin, W. N. and Spears, W. M., editors, *Foundations of Genetic Algorithms 6*, pages 209 – 225. Morgan Kaufmann, San Francisco.

Yin, G., Rudolph, G., and Schwefel, H.-P. (1995). Analyzing (1,λ) evolution strategy via stochastic approximation methods. *Evol. Comput*, pages 473–489.