Theory of Stochastic Drift

A Guided Tour

Timo Kötzing

Compiled on September 26, 2024

Abstract. In studying randomized search heuristics, a frequent quantity of interest is the first time a (real-valued) stochastic process obtains (or passes) a certain value. Commonly, the processes under investigation show a bias towards this goal, the *stochastic drift*. Turning an iteration-wise expected bias into a first time of obtaining a value is the main result of *drift theorems*. This thesis gives an introduction into the theory of stochastic drift, providing examples and reviewing the main drift theorems available. Furthermore, the thesis explains how these methods can be applied in a variety of contexts, including those where drift theorems seem a counter intuitive choice. Later sections examine related methods and approaches.

This document is available as HTML[1]; furthermore, a copy was uploaded to arXiv[2].

Contents

Preface 4

1 What is Stochastic Drift? 5
- 1.1 A Guide to this Document 6

2 A Gentle Introduction to Classic Drift Theorems 7
- 2.1 The Additive and the Multiplicative Drift Theorems 7
- 2.2 Some Simple Applications 9
- 2.3 More Complex Problems 12
- 2.4 Classic Results for Evolutionary Algorithms 13

3 The Art of Potential Functions 15
- 3.1 A Simple Heuristic for Choosing Potential Functions 15
- 3.2 Potential Functions for Two-Part Drift 17
- 3.3 $(1 + 1)$ EA on Linear Functions 17
- 3.4 Designing a Potential Function via Step-Wise Differences 19
- 3.5 Further Potential Functions 23
- 3.6 Conclusion 23

4 Going Nowhere: Drift Without Drift 24
- 4.1 Unbiased Random Walks 24
- 4.2 Analysis of Concrete Unbiased Random Walks 25

5 The Zoo: A Tour of Drift Theorems 29
- 5.1 Additive Drift 29
- 5.2 Additive Drift: Concentration 32
- 5.3 Multiplicative Drift 34
- 5.4 Variable Drift 36
- 5.5 Negative Drift 37
- 5.6 Finite State Spaces 38
- 5.7 Headwind Drift 39
- 5.8 Multiplicative Up-Drift 40
- 5.9 Wormald's Method 42

1 https://hpi.de/friedrich/docs/scripts/22_AllDrift/index.html
2 https://arxiv.org/abs/2406.14589

6 No Going Back: The Fitness Level Method (FLM) 44
6.1 Applications 47

7 A Different Perspective: Fixed Budget Optimization 49
7.1 The Additive Case 49
7.2 Variable Fixed Budget Drift 50
7.3 Applications to OneMax and LeadingOnes 53
7.4 Bibliographic Remarks 53

8 Drift as an Average: A Closer Look on the Conditioning of Drift 55
8.1 Drift as an Average 55
8.2 The Conditioning of Drift 56
8.3 Reaching the Target 59

9 Notation 60
9.1 Algorithms 60
9.2 Notation 60
9.3 Lemmas 62

10 Acknowledgments 65

References 66

Preface

This document gives an introduction to the theory of stochastic drift, as developed by the community researching the theory of randomized search heuristics. For researchers new to the area (but with some basic familiarity with probability theory and random processes), the early sections provide a gentle introduction into the main theorems and sample applications. Later sections give more specialized theorems for particular applications. Seasoned researchers might turn directly to later sections, browsing the list of drift theorems for many settings which provides further pointers to the literature, as well as remarks on details of the techniques and their relation to similar approaches.

Furthermore, this document serves as "a summarized and systematic presentation of the candidate's own work" in partial fulfillment of the requirements for *Habilitation* at the Digital Engineering Faculty of the University of Potsdam, Germany.

1 What is Stochastic Drift?

Suppose that you win a million dollars in a lottery and that you start spending your winnings. You observe that you spend on average 10.000 dollars per day. How long will your lottery winnings last? Intuitively, you would divide the million you won by 10.000 and estimate that your winnings would last for 100 days. But that feels like confusing a random process with a deterministic one. Well, yes, but the good news are: There is a theorem that tells us that 100 days is the mathematically precise answer, even when the process is randomized. Even better, if you gain money on some days (say, by playing in a casino) but still, in expectation, your balance goes down by 10.000 per day, the conclusion still holds. There can even be dependencies between the earnings and spendings of different days (say, on a day after a big earning, you gamble higher amounts than otherwise). The theorem which shows that this is the case is called the *additive drift theorem* (see Theorem 2.1 [Additive Drift, Upper Bound]). The term *drift* refers to the difference between two successive values of the process, and the term *additive* refers to the requirement that the drift is, in expectation, bounded by an additive constant.

A similar setting to that of the process described above is the well-known *coupon collector* process, defined as follows. Suppose you want to collect coupons until you have one of each of n different colors. Each day you get one coupon the color which is chosen uniformly at random and independent of the other days; in particular, you may gain coupons of a color which you already received before. How long does it take until you have a complete set of at least one coupon of each color? For the analysis, note that in the first iteration you get a new color of coupon with certainty (since you do not have any coupons yet). This changes over time: once you already collected exactly half of the colors, the probability of gaining a new one is only one in two. Once you have already gained 90 percent, it is down to one in ten, and so on. Or, flipped around: if you are only half the way from your goal, you only have half the chance of making progress, and if you are 10 percent away from the goal, you make 10 percent of the progress. This is a multiplicative expected progress (the progress is a multiple of the current state of the process) and the *multiplicative drift theorem* (see Theorem 2.5 [Multiplicative Drift]) can be used to analyze exactly this setting. The theorem also holds when the number of coupons gained in each iteration is random (for example, if I gain every color of coupon with a random chance of some value p in each iteration), and it even holds when there is a possibility of losing coupons. Furthermore, it also gives an upper concentration bound.

As these two examples show, we analyze the expected progress of a single step of the given random process in order to find the first time the random process reaches a target state (the so-called "first-hitting time"). This brings us to the following description of drift theory:

> **Drift theory** is a collection of theorems to turn iteration-wise expected gains into expected first-hitting times.

The first drift theorem, the *additive* drift theorem, was introduced by He and Yao [HY01], based on an intricate theorem by Hajek [Haj82]. He and Yao applied their theorem in the context of analyzing randomized search heuristics (RSHs), such as evolutionary algorithms (EAs), which work by the principle of variation (mutating solutions by random changes) and selection (accepting improvements and rejecting worsenings). Drift theory gained a lot of traction in the EA theory community after the *multiplicative* drift theorem was introduced by Doerr, Johannsen, and Winzen [DJW10]. Their proof used additive drift, but a proof not relying on Hajek's result was given shortly after by Doerr and Goldberg [DG10]. Since then, drift theory has been the dominant method for formally analyzing RSHs, easing their analysis significantly over analyses not arguing via drift. For example, the main result of Droste [Dro04] on noisy optimization, spanning an entire paper, was reproven in a more general form by Giessen and Kötzing [GK16] on a single page.

Drift theorems find application in the analysis of a plethora of different settings, ranging from randomized optimization over approximation algorithms to further stochastic processes (see Section 2 [A Gentle Introduction to Classic Drift Theorems] and the second part of Section 4 [Going Nowhere: Drift Without Drift]). Key to the applicability is to model the problem as a search for the time until a random process reaches a target state. However, in spite of the versatility of drift theorems, there are only very few results outside of the theory of RSHs applying drift theory [BLM+20, GLR20, KU18]. Given the versatility of the approach within the area of randomized optimization, it is likely that a higher visibility of these theorems could benefit further research communities.

1.1 A Guide to this Document

This document presents an overview over drift theory. What theorems are available? How can they be applied? What pitfalls abound when using drift theory? Concretely, the contents of this document are as follows.

Section 2 [A Gentle Introduction to Classic Drift Theorems] presents the two already mentioned drift theorems (additive and multiplicative) formally. These are by far the two most important drift theorems and the section includes many examples of their use from a diverse range of settings.

Section 3 [The Art of Potential Functions] discusses the most important technique of making drift theorems applicable: With potential functions, random processes can be mapped to fulfill the requirements of drift theorems, and this section discusses heuristics of how to do this.

One main example for how potential functions can be used to make a process exhibit drift is the analysis of *unbiased* random walks. These walks have, by definition, a drift of 0, but nonetheless drift theory can be used to analyze such processes. This is detailed in Section 4 [Going Nowhere: Drift Without Drift].

> For researchers new to the area, Sections 2 to 4 give a brief but well-rounded introduction to the field. Further sections provide deeper material, extending the applicability and discussing the finer points of drift theory.

Section 5 [The Zoo: A Tour of Drift Theorems] provides a long list of available drift theorems, including the famous *variable* drift theorem, as well as many other drift theorems tailored to various settings of drift. Some example applications and discussions on the relation between the different theorems give an overview of the currently available drift theorems.

A special case of drift is exhibited by monotone processes (processes that cannot go back). This is a specific branch of analysis which was developed independently of the other drift theorems; we discuss the corresponding theorems in Section 6 [No Going Back: The Fitness Level Method (FLM)].

While classic drift theorems give statements about how long it takes for a process to reach a certain state, the dual question is to ask what state to expect after a given number of iterations. Also this area has theorems similar to drift theorems, and we discuss them in Section 7 [A Different Perspective: Fixed Budget Optimization].

In Section 8 [Drift as an Average: A Closer Look on the Conditioning of Drift] we consider the very technical side of drift theorems. We contrast and discuss different ways in stating the drift theorems and point to pitfalls in applying drift theorems without checking all conditions.

Finally, Section 9 [Notation] introduces some notation used in this work, before the author gives acknowledgments in Section 10 [Acknowledgments].

2 A Gentle Introduction to Classic Drift Theorems

In this section we present two classic drift theorems, the additive and the multiplicative drift theorem (see Section 2.1 [The Additive and the Multiplicative Drift Theorems]). These two theorems are the basis for most analyses made by drift theory, and many more advanced drift theorems are variations of these two core examples. We give a number of instructive examples for how and when the theorems are helpful in Section 2.2 [Some Simple Applications], followed by two more complex examples in Section 2.3 [More Complex Problems]; at the end of this section, in Section 2.4 [Classic Results for Evolutionary Algorithms], we provide two classic applications from the theory of randomized search heuristics. Note that, for this section, we refrain from diving into the technically most powerful statements and present simpler versions of these theorems; stronger versions can be found in Section 5 [The Zoo: A Tour of Drift Theorems].

2.1 The Additive and the Multiplicative Drift Theorems

We state the two most commonly used drift theorems. The first drift theorem is the additive drift theorem, which requires a uniform bound on the expected change of a random process. It is due to [HY01, HY04]. The very general version given here is due to [KK19], where also an instructive proof can be found. We give another proof on Theorem 5.1 [Additive Drift, Upper Bound, Time Condition].

Theorem 2.1: Additive Drift, Upper Bound

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable process over $\mathbb{R}$, and let $T = \inf\{t \in \mathbb{N} \mid X_t \le 0\}$. Furthermore, suppose the following two conditions hold (non-negativity, drift).

(NN) For all $t \le T$, $X_t \ge 0$.

(D) There is a $\delta > 0$ such that, for all $t < T$, it holds that $\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \ge \delta$.

Then
$$\mathrm{E}[T] \le \frac{\mathrm{E}[X_0]}{\delta}.$$

The condition (D) is the *drift condition*, this is where we require the additive progress towards the target state 0. Note that we require to have drift for all possible histories $X_0, \ldots, X_t$ of the process. In many applications, we have a Markov chain, which implies that conditioning on the history is equivalent on conditioning on X_t only. See Section 8 [Drift as an Average: A Closer Look on the Conditioning of Drift] for a detailed discussion on what to condition on.

The condition (NN) requires *non-negativity* of the process. We cannot allow the process to assume smaller values than the target 0 as demonstrated by the following example.

Example 2.2: Additive Drift and Processes Reaching Negative Numbers — Suppose our process starts with $X_0 = 5$ and, in each iteration deterministically, the process decreases by 2. Then the expected time (in fact, the deterministic time) until $X_t \le 0$ is exactly 3. If we want to apply Theorem 2.1 [Additive Drift, Upper Bound] we use that the expected gain is 2, so the conclusion suggests an expected time of 2.5. This incorrect conclusions comes from the disregard for (NN).
We can amplify this effect with the following example. Let $(X_t)_{t\in\mathbb{N}}$ be a random process with $X_0 = 1$ and, for all $t \in \mathbb{N}$, with probability $1 - 1/n$, $X_{t+1} = X_t$ and otherwise $X_{t+1} = -n + 1$; the expected time until $X_t \le 0$ is n (since it follows a geometric distribution with probability $1/n$), while the expected gain is 1, for which the additive drift theorem would suggest an expected time of 1.

Note that there are also additive drift theorems that remove the condition (NN) and instead incorporate an additional term in the conclusion, see Theorem 5.3 [Additive Drift, Upper Bound with Overshooting].

The additive drift theorem also allows for a corresponding lower bound as follows [HY01, HY04, KK19]. In Theorem 3 of [KST11], this theorem was used to show a lower bound to derive an asymptotically tight run time analysis of an evolutionary algorithm. Another application can be found in Theorem 4 of [FKL+17, FKL+20].

Theorem 2.3: Additive Drift, Lower Bound

Let $(X_t)_{t \in \mathbb{N}}$ be an integrable process over $\mathbb{R}$, and let $T = \inf\{t \in \mathbb{N} \mid X_t \leq 0\}$. Furthermore, suppose the following conditions (bounded steps, drift).

(B) There is a $c > 0$ such that, for all $t < T$, it holds that $\mathrm{E}[|X_t - X_{t+1}| \mid X_0, \ldots, X_t] \leq c$.

(D) There is a $\delta > 0$ such that, for all $t < T$, it holds that $\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \leq \delta$.

Then
$$\mathrm{E}[T] \geq \frac{\mathrm{E}[X_0]}{\delta}.$$

For this lower bound we need to require (B), a bounded expected step size. This is to avoid counterexamples like the following process.

Example 2.4: Additive Drift and Unbounded Step Size — Let $(X_t)_{t \in \mathbb{N}}$ with $X_0 = 1$ and, for all t, with probability $1/2$, $X_{t+1} = 0$ and otherwise $X_{t+1} = 2X_t - 2\delta$. This process exhibits a drift of δ, suggesting an expected time of $1/\delta$, but the true time until $X_t \leq 0$ is again geometrically distributed, this time with probability $1/2$, giving an expected time of 2.

In order to apply an additive drift theorem, one has to find a single constant δ bounding drift uniformly. However, for processes where large parts of the state space exhibit a drift very differnt from this uniform bound, stronger results can be obtained by using a drift theorem which allows for a different drift in different states of the process.

The multiplicative drift theorem covers the case where the drift is proportional to the current value of the process. It is due to [DJW10], with tail bounds given in [DG10].

Theorem 2.5: Multiplicative Drift

Let $(X_t)_{t \in \mathbb{N}}$ be an integrable process over $\{0, 1\} \cup S$, where $S \subset \mathbb{R}_{>1}$, and let $T = \inf\{t \in \mathbb{N} \mid X_t \leq 0\}$. Assume that there is a $\delta \in \mathbb{R}_+$ such that, for all $s \in S \cup \{1\}$ and all $t < T$, it holds that
$$\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq \delta X_t.$$

Then
$$\mathrm{E}[T] \leq \frac{1 + \ln \mathrm{E}[X_0]}{\delta}.$$

Further, for all $k > 0$ and $s \in S \cup \{1\}$ with $\Pr[X_0 \leq s] > 0$, it holds that
$$\Pr\left[T > \frac{k + \ln s}{\delta} \mid X_0 \leq s\right] \leq \mathrm{e}^{-k}.$$

The condition (D) gives a bound dependent on the history, specifically dependent on the "current" value of the process. Intuitively it requires that, if the process has a current value of $X_t = s$, then the drift is at least δs. In fact, the multiplicative drift theorem is frequently stated with a condition of $X_t = s$ instead of $X_0, \ldots, X_t$, and the upper bound is written as δs instead of δX_t. See Section 8 [Drift as an Average: A Closer Look on the Conditioning of Drift] for a detailed discussion on the different ways to write a drift theorem.

These drift theorems cover a lot of applications; the remainder of this section gives a range of use cases. Most scientists consider the drift theorems stated above first before turning to other drift theorems (see Section 5 [The Zoo: A Tour of Drift Theorems] for a list and discussion of such alternatives). An incomplete list of some applications of these basic theorems, regarding the analysis of evolutionary algorithms, is as follows.

- Theorem 15 of [KSNO12] uses it for a simple $O(n \log n)$ bound.
- Similarly easy arguments are given in Theorems 14 and 17 of [DDK15].
- A number of applications is given in [FKL+17, FKL+20].
- Lemma 2 of [KM12] uses the concentration bound of Theorem 2.5 [Multiplicative Drift].
- So does Theorem 9 of [FKKS15a] (see also Theorem 9 of [FKKS17]) for the analysis of the cGA.
- The application in Theorem 9 of [FK13] is a bit more intricate argument for an upper bound via multiplicative drift.
- Similarly in Theorems 8 and 15 of [FKKS15b] for the analysis of an ant colony optimization (ACO) algorithm optimizing noisy OneMax.

We note that, in our applications of the drift theorems in the following, we do not show that the random processes under consideration are integrable, since this is easily observed from the context that they are defined in.

2.2 Some Simple Applications

We will start by looking at the process from Section 1 [What is Stochastic Drift?] about collecting coupons, a classic process analyzed in many text books on random processes. We start with a (suboptimal) analysis via additive drift.

Theorem 2.6: Coupon Collector with Additive Drift

Suppose we want to collect at least one of each color of $n \in \mathbb{N}_{\geq 1}$ coupons. Each round, we are given one coupon chosen uniformly at random from the n colors. Then, in expectation, we have to collect for at most n^2 rounds.

Proof. Let X_t be the number of coupons missing after t rounds and let T be the random variable describing the first time such that $X_t = 0$. Furthermore, let $t < T$. The probability of making progress (of 1) with coupon $t+1$ is at least X_t/n. In the worst case, when only one color is missing, this is still $1/n$. Thus, $\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] = X_t/n \geq 1/n$. Since we start with $X_0 = n$ missing colors, an application of Theorem 2.1 [Additive Drift, Upper Bound] gives the desired upper bound of n^2 rounds, using $\delta = 1/n$. ■

The analysis with additive drift completely disregards the very high probability of finding new colors while still a lot of colors are missing. Thus, the analysis with multiplicative drift gives a much better bound, as the following theorem shows. In a sense, the multiplicative drift theorem is a generalization of the classic analysis of the coupon collector process; or, vice versa, the analysis of the coupon collector process follows directly from the multiplicative drift theorem.

Theorem 2.7: Coupon Collector with Multiplicative Drift

Suppose we want to collect at least one of each color of $n \in \mathbb{N}_{\geq 1}$ coupons. Each round, we are given one coupon chosen uniformly at random from the n colors. Then, in expectation, we have to collect for at most $n(1 + \ln n)$ rounds. Furthermore, for all $k \in \mathbb{R}_{>0}$, overshooting this time by kn has a probability of at most $e^{-(k+1)}$.

Proof. Let X_t be the number of coupons missing after t rounds and let T be the random variable describing the first time such that $X_t = 0$. Furthermore, let $t < T$. The probability of making progress (of 1) with coupon $t + 1$ is X_t/n. Thus, $E[X_t - X_{t+1} \mid X_0, \ldots, X_t] = X_t/n$. An application of Theorem 2.5 [Multiplicative Drift] gives the desired result. ∎

A lower bound can be derived with an appropriate lower bounding multiplicative drift theorem (see Theorem 5.12 [Coupon Collector, Lower Bound]). Since the process is monotone, both an upper and a lower bound can be derived with the *fitness level method*, see Theorem 6.6 [Coupon Collector, Lower Bound via Fitness Levels].

Using an analogous proof as in Theorem 2.7 [Coupon Collector with Multiplicative Drift], one can directly analyze a *generalized version* of the coupon collector process as follows.

Theorem 2.8: Generalized Coupon Collector

Suppose we want to collect at least one of each color of $n \in \mathbb{N}_{\geq 1}$ coupons. For each color of coupon and each round, we get this color of coupon with probability at least $p \in (0, 1]$. Then, in expectation, we have to wait for at most $(1 + \ln n)/p$ rounds. Furthermore, for all $k \in \mathbb{R}_{>0}$, overshooting this time by k/p has a probability of at most $e^{-(k+1)}$.

Proof. Let X_t be the number of coupons missing after t rounds and let T be the random variable describing the first time such that $X_t = 0$. Furthermore, let let $t < T$. The expected progress is $E[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq pX_t$, since the expected number of missing coupons that we get in the next iteration is pX_t. An application of Theorem 2.5 [Multiplicative Drift] gives the desired result. ∎

Note that this generalized version does not make any assumptions on how many coupons we get per iteration, or whether these indicator random variables are in any way correlated.

We now turn to the well-known geometric distribution. The typical computation for its expectation involves modifying infinite sums. Using drift, the computation is rather simple. Furthermore, our analysis allows for processes where the probability of success changes over time and depends on the history, but a uniform bound on this probability is known.

Theorem 2.9: Geometric Distribution

Let $(X_t)_{t \in \mathbb{N}}$ be some random process where, in each iteration, a *success* event happens with some probability, possibly dependent on the history of the process; we let $S(X_0, \ldots, X_t)$ denote the success event. Then the following estimates hold for all $p \in (0, 1]$.

(1) If, for each t, $\Pr[S(X_0, \ldots, X_t) \mid X_0, \ldots, X_t] \leq p$, then the expected time until any success event happened is at least $1/p$.

(2) If, for each t, $\Pr[S(X_0, \ldots, X_t) \mid X_0, \ldots, X_t] \geq p$, then the expected time until any success event happened is at most $1/p$.

(3) If, for each t, $\Pr[S(X_0, \ldots, X_t) \mid X_0, \ldots, X_t] = p$, then the expected time until any success event

happened is exactly $1/p$.

Proof. We start with the first statement. For all $t \in \mathbb{N}$, let X_t be 0 if a success event has happened within the first t iterations, and 1 otherwise. We let T be the random variable describing the first time that a success event happened. Furthermore, let $t < T$. Then $E[X_t - X_{t+1} \mid X_0, \ldots, X_t] \le p$. Thus, Theorem 2.3 [Additive Drift, Lower Bound] give us the corresponding bounds of at least $1/p$ iterations until the first success event. Analogously, we get the second statement from Theorem 2.1 [Additive Drift, Upper Bound]. The third statement is the conjunction of the first two. ∎

Next we consider a sequence of fair coin tosses. Known as the Gambler's Fallacy is the believe that a sequence of "heads" makes the occurrence of "tails" more likely. Quite in contrast to this, for any given $k \in \mathbb{N}$ there will be an occurrence of k "heads" in a row if the coin is tossed sufficiently often. In the following theorem we derive exactly how long we have to wait in expectation for such an event to happen. In the proof we apply the additive drift theorem not going down towards 0, but going up to a value of k. Since the additive drift is symmetrical, we can use it in either direction equally.

Theorem 2.10: Winning Streaks

Let $k \in \mathbb{N}$ be given. Consider flipping a fair coin indefinitely. Then the expected number of coin flips until the first time that *heads* comes up k *times in a row* is (exactly) $f(k) = 2^{k+1} - 2$.

Proof. For all $t \in \mathbb{N}$, let R_t be the length of the current streak of heads after t iterations ($R_t = 0$ if in iteration t we got tails, as well as before any coin flip at $t = 0$). In the following computation, we will condition on a value for the current search point, which is equivalent to conditioning on the history since our process is a discrete Markov chain (see Section 8 [Drift as an Average: A Closer Look on the Conditioning of Drift] for details). Let $X_t = f(R_t)$ be our process for which we aim to show drift. Let $i \in \mathbb{N}$ be given. If our current streak of heads is i, then in the next iteration one of two things happens: either we lose all progress, falling to a 0 heads, or we now have a streak of $i+1$ heads. Each happens with probability $1/2$, so we have

$$\begin{aligned} E[X_{t+1} - X_t \mid X_t = f(i)] &= E[X_{t+1} \mid X_t = f(i)] - f(i) \\ &= \frac{1}{2} f(i+1) + \frac{1}{2} f(0) - f(i) \\ &= \left(2^{i+2}/2 - 2/2\right) + 0/2 - (2^{i+1} - 2) \\ &= 1. \end{aligned}$$

Thus, using Theorem 2.1 [Additive Drift, Upper Bound] and Theorem 2.3 [Additive Drift, Lower Bound] together, going up instead of down, we get an expected number of iterations of $f(k) = 2^{k+1} - 2$ to reach a streak of k heads. ∎

Note that the potential function in the last proof, as in many places where potential functions are used, is not intuitive, so let us discuss where this potential function comes from. We decide we want to set up for additive drift, since the additive drift theorem gives both lower and upper bounds. Since any potential function that gives an additive drift can be normalized to give an additive drift of 1, we search for a potential function that gives a drift of exactly 1. From the two possible outcomes of the coin flipping process in each iteration, we now get the condition of $f(i+1)/2 - f(i) = 1$ for the potential f. In this case, this is a straightforward and easy to solve recurrence relation, so that with the (arbitrary) setting of $f(0) = 0$ we get the desired formula for f.

For a more in-depth discussion of potential functions and their use for the application of drift theorem, see Section 3 [The Art of Potential Functions].

2.3 More Complex Problems

In contrast to the previous applications of drift theorems, the following examples consider processes that are not Markovian. This is no problem for the drift theorems, but the user now has to make sure that all bounds hold regardless of the history, not just with respect to the current value of the process.

Our next example is a randomized algorithm for finding, in expectation, a 2-approximation of the classical vertex cover problem. For an undirected graph (V, E), a subset $C \subseteq V$ such that, for all $\{u, v\} \in E$, u or v is in C is called a *vertex cover*. By Theorem 2.1 [Additive Drift, Upper Bound], we easily bound the expected size of the vertex cover that the algorithm constructs.

Theorem 2.11: Vertex Cover Approximation

Given an undirected graph, iteratively choose an uncovered edge and add uniformly at random an endpoint to the cover. Then, in expectation, the resulting cover is a 2-approximation of an optimal vertex cover of the given graph.

Proof. Let a graph G be given. Furthermore, fix a minimum vertex cover C. For all t, let D_t be the set of vertices chosen by the algorithm after t iterations. Let X_t be 0 if D_t is a vertex cover, and otherwise let X_t be the number of vertices of C that are not in D_t. Clearly, the algorithm terminates exactly when $X_t = 0$. Furthermore, in each step and regardless of the history, the algorithm selects a vertex from C with probability at least 1/2, since, for every edge of G, at least one of the endpoints is in C. We get $\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq \frac{1}{2}$. Hence, using Theorem 2.1 [Additive Drift, Upper Bound], we get that the algorithm terminates in expectation after choosing $2|C|$ vertices. ■

The next example considers a simple randomized sorting algorithm. This and similar sorting algorithms were considered by Scharnow, Tinnefeld, and Wegener [STW04] (before the advent of drift theory). The analysis via the multiplicative drift theorem is short, easy and intuitive.

Theorem 2.12: Random Sorting

Consider the sorting algorithm which, given an input array A of length $n \in \mathbb{N}_{\geq 1}$, iteratively chooses two different positions of the array uniformly at random and swaps them if and only if they are out order. Then the algorithm obtains a sorted array after $\Theta(n^2 \log n)$ iterations in expectation.

Proof. For all $i, j \in [n]$ with $i < j$, an ordered pair (i, j) is called an *inversion* if and only if $A[i] > A[j]$. Note that the maximum number of inversions is $\binom{n}{2}$. Let X_t be the number of inversions after $t \in \mathbb{N}$ iterations, and let A_t denote the array after that iteration. If the algorithm chooses a pair which is not an inversion, nothing changes. If the algorithm chooses an inversion (i, j), then this inversion is removed; for any other inversion, only indices $k \in [i..j]$ are relevant. If $A_t[k] < A_t[j]$ $(< A_t[i])$, then (i, k) is an inversion before and after the swap, while (k, j) is neither an inversion before nor after the swap; similarly for $A_t[k] > A_t[i]$ $(> A_t[j])$. Finally, if $A_t[j] < A_t[k] < A_t[i]$, then (i, k) and (k, j) are inversions before the swap but are not afterwards. Overall, this shows that the number of inversions goes down by at least 1 whenever the algorithm chooses an inversion for swapping, regardless of the history.

Let t be such that A_t is not sorted. Since the probability of the algorithm choosing an inversion is $X_t/\binom{n}{2}$, we get $\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq X_t/\binom{n}{2}$. An application of Theorem 2.5 [Multiplicative Drift] gives the desired upper bound.

Regarding the lower bound, consider the array A which is almost sorted but the first and second element are swapped, the third and fourth, and so on. Then the algorithm effectively performs a coupon collector process on $n/2$ coupons, where each has a probability of $1/\binom{n}{2}$ to be collected. This takes an expected time of $\Omega(n^2 \log n)$ with a proof analogous to that of Theorem 5.12 [Coupon Collector, Lower Bound]. ■

2.4 Classic Results for Evolutionary Algorithms

The basic evolutionary algorithm (EA) we want to analyze is the $(1+1)$ EA; it proceeds as follows (see also Section 9.1 [Algorithms]).

Algorithm 1: The $(1+1)$ EA

```
1 Sample x ∈ {0,1}^n uniformly at random
2 for i = 1 to ∞ do
3     y ← mutate(x)
4     if f(y) ≥ f(x) then x ← y
```

The $(1+1)$ EA minimizing a function $f: \{0,1\}^n \to \mathbb{R}$. Mutation flips each bit independently with probability $1/n$.

The algorithm is set up to maximize the given function f; by turning around the inequality in line 4, we get the analogous algorithm for minimization.

The two cardinal test functions that are used to analyze the performance of evolutionary algorithms are OneMax and LeadingOnes:

- OneMax is a function $\{0,1\}^n \to \mathbb{R}$ mapping any bit string to the number of 1s in the bit string.
- LeadingOnes is a function $\{0,1\}^n \to \mathbb{R}$ mapping any bit string to the number of 1s *before the first* 0 (if any) in the bit string (the number of leading 1s).

See also Definition 9.1 [Test Functions].

Theorem 2.13: $(1+1)$ EA on OneMax

Consider the $(1+1)$ EA maximizing the fitness function OneMax. Then the expected time for the algorithm to find the global optimum 1^n is $O(n \log n)$ iterations.

Proof. For all t, let X_t be the Hamming distance to the optimum of the current individual after t iterations. We want to use the multiplicative drift theorem and estimate the drift as follows. If the currently best search point has a Hamming distance of s, then, for each bit i of the s missing positions, the event of flipping position i and no other when producing offspring will result in an accepted offspring with a distance of 1 less to the optimum. These events are disjoint (since only one bit flips) and each has a probability of $1/n \cdot (1-1/n)^{n-1} \geq 1/(en)$. Since the $(1+1)$ EA does not accept worsenings, no other event can contribute negatively to the drift, so we can pessimistically assume a contribution of 0 to the drift in all other cases. Thus, we get

$$\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq X_t/(en).$$

An application of Theorem 2.5 [Multiplicative Drift] gives the desired upper bound since $X_0 \leq n$. ■

Theorem 2.14: $(1+1)$ EA on LeadingOnes

Consider the $(1+1)$ EA maximizing the fitness function LeadingOnes. Then the expected time for the algorithm to find the optimum is $O(n^2)$ iterations.

Proof. For all t, let X_t be the number of leading ones of the current individual after t iterations (i.e. the fitness). We want to use the additive drift theorem and estimate drift as follows. Improving the fitness of the current individual requires flipping its first 0 and none of the previous positions. There are at most $n-1$ previous positions, so the probability is at least $1/n \cdot (1-1/n)^{n-1} \geq 1/(en)$. An improvement is an improvement by at least 1. Since the $(1+1)$ EA does not accept worsenings, no event can contribute negatively to the drift, so we can pessimistically assume a contribution of 0. Thus, we get

$$\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq 1/(en).$$

An application of Theorem 2.1 [Additive Drift, Upper Bound] gives the desired upper bound since $X_0 \geq 0$ and the target is at n. ■

Note that much more precise bounds are known for the optimization time of the $(1+1)$ EA on LeadingOnes, see Theorem 6.7 [Run Time of $(1+1)$ EA on LeadingOnes].

3 The Art of Potential Functions

Drift theorems can be applied to random processes on $\mathbb{R}$. For the analysis of randomized algorithms, this typically means that one has to map the state of the algorithm to a real number, so that the resulting process will be a process on $\mathbb{R}$. Such a mapping is called *potential function* and we already saw multiple in Section 2 [A Gentle Introduction to Classic Drift Theorems]. Especially in the proof of Theorem 2.10 [Winning Streaks] we saw that sometimes unintuitive potential functions can lead to very strong results. In fact, one could say that the art of applying drift theorems is in choosing the right potential function. Later in this work, for example in the proof of Theorem 4.1 [Unbiased Random Walk on the Line], we see yet more intricate potential functions. In this section, we want to discuss a few cardinal examples.

3.1 A Simple Heuristic for Choosing Potential Functions

There is a very important rule of thumb to designing potential functions: **better search points** should have **better potential**. The following example showcases this.

Example 3.1: Better Search Points with Better Potential — Let $(X_t)_{t\in\mathbb{N}}$ be a Markov chain with $X_0 = 2$ and, for all t, if $X_t = 2$ then, with probability $1 - 1/100$, $X_{t+1} = 0$ and otherwise $X_{t+1} = 1$; if $X_t = 1$ then $X_{t+1} = 0$ with probability $1/100$ and $X_{t+1} = 1$ otherwise. Additive drift provides an upper bound of an expected 200 iterations to reach 0 (since the lowest drift of $1/100$ is encountered in state 1 and we start in state 2).
This process is very much misleading in that state 1 sounds like it is closer than 2 to the target of 0, when actually it is not. Consider the potential function $f(0) = 0$, $f(1) = 100$ and $f(2) = 2$. Now both states 1 and 2 have an drift of exactly 1 towards the target 0, and we start in a state with potential 2, so we get an expected time of 2 to reach 0 from the additive drift theorem.

From this example we see that what seems "natural" (because some process on the reals presents itself) might not be the best for drift. In fact, as we will see later in this section, a potential can turn drift away from the optimum into drift towards the optimum.

The example can be generalized to arbitrary processes: on time-homogeneous Markov chains, the best potential for getting a tight bound with the additive drift theorem is the potential which assigns each state the time until finding the target from starting in that state. The next theorem from [HY04] makes this formal.

Theorem 3.2: Expected Time as Potential

Let $\mathcal{X}$ be some state space and let $(X_t)_{t\in\mathbb{N}}$ be a time-homogeneous Markov chain on $\mathcal{X}$ and let $O \subseteq \mathcal{X}$ be a set of targets. For any $x \in \mathcal{X}$, let $T(x)$ be the random variable describing the number of steps until reaching an element in O (for the first time) when starting in x, and suppose that all such $T(x)$ have finite expectation. We define

$$g\colon \mathcal{X} \to \mathbb{R}, x \mapsto \mathrm{E}[T(x)].$$

Then, for all t with $X_t \notin O$,

$$\mathrm{E}[g(X_t) - g(X_{t+1}) \mid t < T(X_0)] = 1.$$

Proof. Since $(X_t)_{t\in\mathbb{N}}$ is a time-homogeneous Markov-chain, let an operator θ be given such that, for all $t \in \mathbb{N}$, $X_{t+1} = \theta(X_t)$. For all $i \in \mathbb{N}$, we use θ^i to denote the i-times self-composition of θ. In particular, for all $t \in \mathbb{N}$, we have

$$T(X_t) = \min_{i\in\mathbb{N}} \theta^i(X_t) \in O = \min_{i\in\mathbb{N}} X_{t+i} \in O.$$

For all $t \in \mathbb{R}$, conditional on $t < T(X_0)$ (which implies $X_t \notin O$) we thus have

$$T(X_{t+1}) = \min_{i \in \mathbb{N}} X_{t+i+1} \in O = \left(\min_{i \in \mathbb{N}} X_{t+i} \in O\right) - 1 = T(X_t) - 1.$$

In particular,

$$\begin{aligned} \mathrm{E}[g(X_t) - g(X_{t+1}) \mid t < T(X_0))] &= \mathrm{E}[\mathrm{E}[T(X_t)] - \mathrm{E}[T(X_{t+1})] \mid t < T(X_0)] \\ &= \mathrm{E}[\mathrm{E}[T(X_t)] - \mathrm{E}[T(X_t) - 1] \mid t < T(X_0)] \\ &= 1. \end{aligned}$$

This shows the claim. ■

The theorem is interesting for understanding what a good potential should be; in order to apply a drift theorem it is, however, completely useless: We could now use upper and lower additive drift theorems and the proven drift of 1 to derive an expected time of $\mathrm{E}[g(X_0)]$ to find an element of O when starting in X_0. But $g(X_0)$ is defined to be the expected time to find an element from O when starting in X_0, so we arrived where we started.

Note that, in general, after mapping a Markov chain with a potential function, the resulting process is not necessarily a Markov chain anymore. This is not a problem at all, since a well-formulated drift theorem does not need the requirement of the process being a Markov chain, see Section 8 [Drift as an Average: A Closer Look on the Conditioning of Drift] for a discussion.

When considering optimization algorithms, it is sometimes easy to show that the *distance to the optimum* or directly the *fitness* decreases in expectation in each step. This means that this would make for a good potential function; but sometimes this expected change is either too weak, too hard to analyze or even negative. In this case, more inventive potential functions are sought, which is the concern of the remainder of this section.

Before we dive into developing concrete potential functions, we discuss normalizing potential functions. This will later make one decision very easy for us.

Theorem 3.3: Normalizing Additive Drift

Let $\mathcal{X}$ be some state space and let $(X_t)_{t \in \mathbb{N}}$ be a random process on $\mathcal{X}$. Let $c \in \mathbb{R}_{>0}$ and let $g \colon \mathcal{X} \to \mathbb{R}$ be any potential function such that

$$\mathrm{E}[g(X_t) - g(X_{t+1}) \mid g(X_0), \ldots, g(X_t)] \geq c.$$

Then there is a potential function $\overline{g} \colon \mathcal{X} \to \mathbb{R}$ such that

$$\mathrm{E}[\overline{g}(X_t) - \overline{g}(X_{t+1}) \mid \overline{g}(X_0), \ldots, \overline{g}(X_t)] \geq 1,$$

and $\mathrm{E}[\overline{g}(X_0)] = \mathrm{E}[g(X_0)]/c$.

Proof. We choose $\overline{g} = g/c$. ■

The theorem shows that, whenever there is any potential function at all amenable to analysis by additive drift, there is one with an drift of 1. Note that normalization has no impact on the resulting time bound, since the starting value is scaled correspondingly (and now equals the time bound derived by the additive drift theorem).

3.2 Potential Functions for Two-Part Drift

In our first example, we want to "glue together" two drift regimes.

> **Example 3.4: Gluing Together Fitness Functions** — Let $(X_t)_{t\in\mathbb{N}}$ be a discrete integrable process on $[0,n]$ with $X_0 = n$ and let $k \in [0..n]$. Let T be the first time t such that $X_t = 0$. Suppose that we have
>
> $$\mathrm{E}[X_t - X_{t+1} \mid X_t \geq k] \geq 2$$
>
> and
>
> $$\mathrm{E}[X_t - X_{t+1} \mid 0 < X_t < k] \geq 1.$$
>
> In other words: while the potential is high, we get a drift of 2, for small values only a drift of 1. Further assume that, if $X_t < k$, then also $X_{t+1} < k$. We want to show that
>
> $$\mathrm{E}[T] \leq \frac{n+k}{2}.$$
>
> In order to make use of the stronger drift for large values of the process, we choose the potential function
>
> $$g\colon \mathbb{R} \to \mathbb{R}, x \mapsto \begin{cases} x, & \text{if } x < k; \\ (x+k)/2, & \text{otherwise.} \end{cases}$$
>
> We have $g(X_0) = (n+k)/2$, so it is sufficient to show a drift of at least 1 to get our desired bound. This holds trivially for $X_t < k$, heavily relying on the fact that this implies $X_{t+1} < k$. For the following reasoning, note that, for all $x \in \mathbb{R}$, $g(x) \leq (x+k)/2$. For $X_t \geq k$, we see
>
> $$\begin{aligned} \mathrm{E}[g(X_t) - g(X_{t+1}) \mid X_t \geq k] &\geq \mathrm{E}[(X_t+k)/2 - (X_{t+1}+k)/2 \mid X_t \geq k] \\ &= \mathrm{E}[X_t - X_{t+1} \mid X_t \geq k]/2 \\ &\geq 1. \end{aligned}$$
>
> This shows a drift of 1 in potential and thus gives the desired bound using Theorem 2.1 [Additive Drift, Upper Bound].

Note that the potential function g used in the proof above is concave, so in the main derivation in the proof we could have reasoned with Jensen's Inequality. Thus, this approach generalizes to other concave potential functions.

3.3 $(1+1)$ EA on Linear Functions

One of the most famous examples of an analysis with potential functions and drift theory is the analysis of the $(1+1)$ EA (see Section 9.1 [Algorithms]) on linear functions. In fact, the paper introducing the multiplicative drift analysis [DJW12] used this drift theorem with a suitable potential function to show an upper bound of $(1+o(1))\ 1.39e\ n\ln(n)$ on arbitrary linear functions. This bound was later improved to $(1 \pm o(1))\ e\ n\ln(n)$, including a matching lower bound, in [Wit13], with a more intricate potential function that crucially depended on the concrete linear function.

Here we give a simple proof from [DJW12], showcasing the use of potential functions which achieves a bound of $(1+o(1))\ 4e\ n\ln(n)$. We will further restrict the linear functions to have no duplicate weights, avoiding to treat this edge case.

Theorem 3.5: (1 + 1) EA on Linear Functions, no duplicated weights

Let f be any linear function without duplicate weights. The expected time until the $(1+1)$ EA on f samples the optimum for the first time is $(1+o(1))\ 4e\ n\ln(n)$.

Proof. Let $w_1, \ldots, w_n$ be the weights of f. We note that the $(1+1)$ EA is *unbiased*, that is, "reordering" the bit positions and swapping the "meaning" of 0 and 1, leads to analogous behavior of the algorithm [LW12]. Thus, we can assume, without loss of generality, that the weights are ordered decreasingly and are positive, that is,

$$w_1 > w_2 > \ldots > w_n > 0.$$

Now we define our potential function as follows. Let $g\colon \{0,1\}^n \to \mathbb{R}$ be such that, for all $x \in \{0,1\}^n$,

$$g(x) = \sum_{i=1}^{n} (2 - i/n)(1 - x_i).$$

This potential essentially awards a potential weight between 1 and 2 to any incorrectly set bit, where higher potential weight for a bit position corresponds with higher weight of this bit position in the objective function. We call $2 - i/n$ also the *potential of bit i*.

For each $t \geq 0$, let X_t be the current best bit string found by the $(1+1)$ EA after t iterations. We want to show that there is multiplicative drift in $(g(X_t))_{t\in\mathbb{N}}$. To that end, fix $t \in \mathbb{N}$. Let $I = \{i \leq n \mid x_i = 0\}$ be the set of positions where the current best bit string has a 0. We define a number of events that we want to distinguish; these events will be a partition of the entire event space at iteration t.

- For each $i \in I$, let A_i be the event that bit i is the only 0-bit which is flipped by mutation and the final offspring is accepted.
- Let C be the event that at least 2 of the 0 bits are flipped.
- Let D be the event that no 0 bit is flipped.

Let $\Delta = g(X_t) - g(X_{t+1})$ be the drift. We can now get the following breakup of the drift by the law of total expectation.

$$\begin{aligned} \mathrm{E}[\Delta \mid g(X_t)] = \ & \mathrm{E}[\Delta \mid g(X_t), C] \Pr[C \mid g(X_t)] \\ & + \mathrm{E}[\Delta \mid g(X_t), D] \Pr[D \mid g(X_t)] \\ & + \sum_{i \in I} \mathrm{E}[\Delta \mid g(X_t), A_i] \Pr[A_i \mid g(X_t)]. \end{aligned}$$

If no 0 bit flips (event D), then any flip of a 1 bit will result in worse offspring, which will be discarded; thus, $\mathrm{E}[\Delta \mid g(X_t), D] = 0$.

Now consider event C. At least two 0 bits flip, leading to an increase in potential of at least 2. There are at most n many 1 bits, each flipping with a probability of $1/n$ and the potential associated with that bit is strictly less than 2. Thus we lose (in expectation) strictly less than a potential of 2 from flipping 1 bits, but we gain a potential of at least 2 from flipping 0 bits. Furthermore, the result might be accepted or not, and if it is not accepted, then $\Delta = 0$. Note that, the more 1 bits are flipped, the less likely the offspring is accepted, so there is a negative correlation between number of 1 bits flipped and the probability of acceptance. This shows $\mathrm{E}[\Delta \mid g(X_t), C] \geq 0$.

Now we consider the events A_i. Note that, for all $i \in I$, $P(A_i) \geq (1 - 1/n)^{n-1}/n \geq 1/en$, since the event that bit i flips and no other is a subevent of A_i. Not flipping $n-1$ bits has a probability of $(1-1/n)^{n-1}$, and flipping a specific bit has a probability of $1/n$. Crucially, it is impossible that any bit with position $< i$ flips and the offspring is accepted, since it has a (strictly!) higher weight than bit i (and bit i is the only 0 bit that flips).

Overall, we have the following.

$$\begin{aligned}
\mathrm{E}[\Delta \mid g(X_t)] &\geq \sum_{i \in I} \mathrm{E}[\Delta \mid g(X_t), A_i] \Pr[A_i \mid g(X_t)] \\
&\geq \sum_{i \in I} \frac{1}{en} \mathrm{E}[\Delta \mid g(X_t), A_i] \\
&\geq \frac{1}{en} \sum_{i \in I} \left[\left(2 - \frac{i}{n}\right) - \sum_{j=i+1}^{n} \frac{1}{n}\left(2 - \frac{j}{n}\right) \right] \\
&= \frac{1}{en} \sum_{i \in I} \left[\left(2 - \frac{i}{n}\right) - \frac{1}{n}\left(2(n-i) - \frac{\sum_{j=i+1}^{n} j}{n}\right) \right] \\
&= \frac{1}{en} \sum_{i \in I} \left[2 - \frac{i}{n} - \frac{2(n-i)}{n} + \frac{\sum_{j=i+1}^{n} j}{n^2} \right] \\
&= \frac{1}{en} \sum_{i \in I} \left[\frac{-i + 2i}{n} + \frac{n(n+1) - (i+1)i}{2n^2} \right] \\
&\geq \frac{1}{en} \sum_{i \in I} \left[\frac{i}{n} + \frac{n(n+1) - (i+1)n}{2n^2} \right] \\
&= \frac{1}{en} \sum_{i \in I} \left[\frac{i + n/2 - i/2}{n} \right] \\
&= \frac{1}{en} \sum_{i \in I} \left[\frac{1}{2} + \frac{i}{2n} \right] \\
&\geq \frac{1}{en} \sum_{i \in I} \frac{1}{2} \\
&= \frac{|I|}{2en}.
\end{aligned}$$

Since we have $|I| \geq g(X_t)/2$, we get a multiplicative drift with drift constant $\delta = 4en$. Using $g(X_0) \leq 2n$, we can apply Theorem 2.5 [Multiplicative Drift] to get

$$\mathrm{E}[T] \leq (1 + o(1))\, 4e\, n \ln(n).$$

■

3.4 Designing a Potential Function via Step-Wise Differences

In this section we give a method for finding a suitable potential function by defining the potential differences of "neighboring" states. Note that this method was used to find the proof given in Theorem 2.10 [Winning Streaks] and is also the basis of the work in Section 5.6 [Finite State Spaces], with details in [KK18]. Furthermore, a version of this method for overcoming negative drift was given in [GK14, GK16]. Early work using such an approach can be found in [DJW00].

We consider the optimization of a test function which looks like OneMax for most of the search space, but around the optimum is a plateau of constant fitness. This is a fitness function defined as follows, for a given parameter $k \in \mathbb{N}$ (for $x \in \{0,1\}^n$ we use $|x|_1$ to denote the number of 1s in x).

$$\text{PLATEAU}_k \colon \{0,1\}^n \to \{0,1\}^n, x \mapsto \begin{cases} |x|_1, & \text{if } |x|_1 \le n-k \text{ or } |x|_1 = n; \\ n-k, & \text{otherwise.} \end{cases}$$

This function is maximized by the bit string 1^n. All bit strings with a distance between 1 and k to the optimum have identical fitness, so there is no guiding signal towards the optimum on that so-called plateau.

We want to study the random search heuristics $(1+1)$ EA and RLS on the plateau function (see Section 9.1 [Algorithms]).

For the $(1+1)$ EA, we can analyze the performance as follows. Within $O(n \log n)$ iterations the algorithm will have found a search point on the plateau, that is, at distance at most k to the optimum (this follows analogously to the analysis of $(1+1)$ EA on OneMax). From now on at most k bits will be incorrect, and correcting exactly those k bits and no others has a probability of

$$\left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \ge \frac{1}{en^k}.$$

Thus, for $k \ge 2$, the total expected optimization time will be $O(n^k)$. This reasoning disregards the analysis of the random walk performed by the algorithm on the plateau.

The search heuristic Random Local Search (RLS) exchanges the mutation operator of the $(1+1)$ EA for an operator which flips exactly one bit. The analysis of the $(1+1)$ EA above explicitly makes use of large steps which are not performed by RLS, so a different analysis is required. We now have to understand the random walk on the plateau as an essential part to finding the optimization time, and analyzing it with drift theory provides a nice example of the power of potential functions. In fact, it is somewhat surprising that, also for this fitness function, an analysis with drift theory can find a good bound on the expected optimization time: Using the fitness function as the potential function, the drift for "inner" points on the plateau (where all neighbors are also points on the plateau) is 0.

We want to develop a potential function that is 0 at the optimum. We aim for an drift for the RLS optimizing Plateau of at least 1, given that Theorem 3.3 [Normalizing Additive Drift] shows that we can always find a normalized drift function. For reasons of symmetry, all bit strings at the same distance to the optimum should have the same potential, so we now wonder what should be the potential of a bit string with exactly d many 0s (so d is the Hamming distance to the optimum). Let us simplify and consider first the case of $d = 1$.

On the plateau, any change is accepted by RLS. Thus, if there is only one incorrect bit, RLS will correct it with probability $1/n$ and otherwise lose a different bit with probability $1 - 1/n$. If we think about the potential difference between $d = 0$ and $d = 1$ as $a(0)$, and the potential difference between $d = 1$ and $d = 2$ as $a(1)$, then the expected gain in potential is given by

$$\frac{1}{n} \cdot a(0) - \left(1 - \frac{1}{n}\right) \cdot a(1) = \frac{a(0) - (n-1)a(1)}{n}.$$

We want this quantity to be a least 1, so, for fixed $a(0)$, we get $a(1) \le (a(0) - n)/(n-1)$. This is a rather complex term, but note that for $a(0) \ge 2n$, we can choose $a(1) = a(0)/(2n)$, a much simpler term.

Turning to the general case of arbitrary d, we get an drift of

$$\frac{d}{n} \cdot a(d-1) - \left(1 - \frac{d}{n}\right) \cdot a(d) = \frac{d \cdot a(d-1) - (n-d)a(d)}{n}.$$

Thus, if again $a(d-1) \ge 2n$, we could work with $a(d) = a(d-1)/(2n)$. Inductively, we now have $a(d) = a(0)/(2n)^d$.

Note that we need this to hold for d starting at $d = 0$ up until $d = k - 1$, since on the plateau we cannot go outward from being exactly k away. We can now choose $a(0)$ to suit all requirements. Concretely, for all $d \in [0..k-1]$ we need

$$\frac{a(0)}{(2n)^d} = a(d) \geq 2n,$$

so $a(0) \geq (2n)^{d+1}$. This restriction is strongest for $d = k - 1$, leading to $a(0) \geq (2n)^k$.

After we established the size of the different *gaps* between different states of the algorithm, we now define the potential function (denoting the number of 0s in a bit string x as $|x|_0$) as

$$g\colon \{0,1\}^n \to \mathbb{R}_{\geq 0}, x \mapsto \sum_{i=0}^{|x|_0-1} a(i).$$

This way of defining a potential function as a sum of “gaps” has a the advantage that the difference in potential of similar search points is easy to compute.

We now get to the final proof, where we will also need to worry about the “easy” part of the search space (again “gluing together” the drift regimes as in Section 3.2 [Potential Functions for Two-Part Drift]). Note that we simplify the terms $a(i)$ by changing the base from $2n$ to n; while the $2n$ suggested itself during proof discovery, the final computations can do without.

Theorem 3.6: RLS on Plateau, upper bound

Let $k \geq 2$. The expected time for RLS to optimize PLATEAU is $O(n^k)$.

Proof. Let $(X_t)_{t\in\mathbb{N}}$ be the current search point of RLS after t iterations. For all $d \in \mathbb{N}$ we define $a(d) = n^{k-d}$ and $g_0 = \sum_{i=0}^{k-1} a(i)$. Now we define a potential function g as

$$g\colon \{0,1\}^n \to \mathbb{R}_{\geq 0}, x \mapsto \begin{cases} \sum_{i=0}^{|x|_0-1} a(i), & \text{if } |x|_0 \leq k; \\ g_0 + (|x|_0 - k)n, & \text{if } |x|_0 > k. \end{cases}$$

Intuitively, we artificially distort the gaps where the plateau does not provide a fitness signal and use unit gap sizes in the easy part. Note that this is suboptimal for the easy part, but the impact on the overall bound of the theorem will only be in lower order terms, since the time to cross the plateau dominates.

Let now t be given and let $x = X_t$ and $x' = X_{t+1}$. We are interested in bounding $\mathrm{E}[g(x) - g(x')]$. We will use the law of total expectation and make a case distinction on $|x|_0$.

First, let $d < k$ be given and consider an iteration of RLS on a bit string with exactly d many 0s (where it flips exactly 1 bit). RLS either gains a potential of $a(d-1)$, with probability d/n, or loses a potential of $a(d)$, otherwise. We have

$$\begin{aligned} \mathrm{E}[g(x) - g(x') \mid |x|_0 = d] &= \frac{d}{n}a(d-1) - \frac{n-d}{n}a(d) \\ &= \frac{d}{n} \cdot n^{k-d+1} - \frac{n-d}{n} \cdot n^{k-d} \\ &= (dn - n + d) \cdot n^{k-d-1}. \end{aligned}$$

For $d = 1$ this equals $n^{k-2} \geq 1$; for $d > 1$ this is at least $(d-1)n \cdot n^{k-d-1} \geq n^{k-d}$. Since $d \leq k$, this value is at least 1.

We now consider the case of $d = k$. Note that, in this case, we cannot lose potential, as the selection of RLS discards any strictly worse search point. Thus, we have

$$\begin{aligned} \mathrm{E}[g(x) - g(x') \mid |x|_0 = k] &= \frac{k}{n} a(k-1) \\ &= \frac{k}{n} \cdot n^{k-k+1} \\ &= k \geq 1. \end{aligned}$$

Finally, we consider the case of $d > k$. Also in this case we cannot lose potential; we have

$$\begin{aligned} \mathrm{E}[g(x) - g(x') \mid |x|_0 = d] &= \frac{d}{n} \cdot n \\ &= d \geq 1. \end{aligned}$$

Thus, Theorem 2.1 [Additive Drift, Upper Bound] gives an upper bound on the expected optimization time of the maximal potential value of $g_0 + n(n-k) \leq n \cdot 2n^k + n^2 = O(n^k)$. ■

For k constant, we can use drift theory with a similar potential function to find a matching lower bound. Note that this a particular strength of the additive drift theorem: it comes with a matching lower bound without additional requirements to the random process (such as concentration in each step).

Theorem 3.7: RLS on Plateau, lower bound

Let $k \geq 2$ be constant. The expected time for RLS to optimize Plateau is $\Omega(n^k)$.

Proof. Let $(X_t)_{t\in\mathbb{N}}$ be the current search point of RLS after t iterations. For all $d \in \mathbb{N}$ we define $a(d) = ((n-k)/k)^{k-d}$ and $g_0 = \sum_{i=0}^{k-1} a(i)$. Now we define a potential function g as

$$g\colon \{0,1\}^n \to \mathbb{R}_{\geq 0}, x \mapsto \begin{cases} \sum_{i=0}^{|x|_0 - 1} a(i), & \text{if } |x|_0 \leq k; \\ g_0, & \text{if } |x|_0 > k. \end{cases}$$

Intuitively, we ignore the run time outside of the plateau, since it does not contribute to the asymptotic bound.

Let now t be given and let $x = X_t$ and $x' = X_{t+1}$. We are interested in bounding $\mathrm{E}[g(x) - g(x')]$, this time we want an upper bound. We will again use the law of total expectation and make a case distinction on $|x|_0$.

First, let $d < k$ be given and let the current bit string x have $|x|_0 = d$. Since RLS will flip exactly 1 bit, we either gain a potential of $a(d-1)$ (with probability d/n) or lose a potential of $a(d)$, otherwise. From $d \leq k$ we see $d(n-k) \leq k(n-d)$ and thus

$$\frac{d}{n} \frac{n-k}{k} \leq \frac{n-d}{n}.$$

Now we can derive

$$\begin{aligned} \mathrm{E}[g(x) - g(x') \mid |x|_0 = d] &= \frac{d}{n} a(d-1) - \frac{n-d}{n} a(d) \\ &= \frac{d}{n} \cdot ((n-k)/k)^{k-d+1} - \frac{n-d}{n} \cdot ((n-k)/k)^{k-d} \\ &\leq \frac{n-d}{n} \cdot ((n-k)/k)^{k-d} - \frac{n-d}{n} \cdot ((n-k)/k)^{k-d} \\ &= 0. \end{aligned}$$

An upper bound of 0 might seem surprising, but in this area of the search space the distortion by the potential function is large enough to arrive at negative drift.

We now consider the case of $d = k$. In this case, we cannot lose potential. We have

$$\begin{aligned} \mathrm{E}[g(x) - g(x') \mid |x|_0 = k] &= \frac{k}{n} a(k-1) \\ &= \frac{k}{n} \cdot ((n-k)/k)^{k-k+1} \\ &= (n-k)/n \leq 1. \end{aligned}$$

Finally, we consider the case of $d > k$. Since in this case all neighboring search points have the same potential, we again have a drift of 0.

$$\mathrm{E}[g(x) - g(x') \mid |x|_0 = d] = 0.$$

The initial potential is g_0 with a probability of at least some constant c. Thus, Theorem 2.3 [Additive Drift, Lower Bound] gives a lower bound on the expected optimization time of the initial potential value of $cg_0 \geq c(n/k)^k = \Omega(n^k)$. ■

As can be seen from the proof, the lower bound extends to super-constant k as $\Omega((n/k)^k)$. Note that both this lower bound and the upper bound of $O(n^k)$ are no longer optimal for super-constant k. In particular, the extreme case of $k = n$ is known as the Needle function (see [GKS99] for the first analysis on Needle).

3.5 Further Potential Functions

The literature knows many more example applications of potential functions in order to allow for the applications of drift theory. For example, in [FKN⁺23] a clever potential function is used to incorporate a state of the algorithm into the general progress of the algorithm towards the goal. In Section 4.2 of [DKLL17], the potential function essentially has two parts to allow for a unified drift argument, rather than arguing over two phases.

Further interesting potential functions can be found in [DDK18]. One function incorporates speed (a self-adjusting parameter) of the algorithm and the distance to the optimum into a single potential. Another combines the distances in different dimensions in a suitably scaled way to arrive at a useful potential function.

3.6 Conclusion

While building a potential function is more of an art than a science, there are heuristics which can help.

- As we saw in Example 3.1 [Better Search Points with Better Potential], states that are "closer" to the target should have potential "closer" to that of the target; in fact, as shown by Theorem 3.2 [Expected Time as Potential], the most accurate potential assigns each search point the "distance" to the target.
- Drift might be different in different parts of the search space; in this case, we can use potential functions to "glue together" these parts, as showcased by Example 3.4 [Gluing Together Fitness Functions].
- In Section 3.4 [Designing a Potential Function via Step-Wise Differences] we saw one way of iteratively building a potential function by comparing "neighboring" states.

4 Going Nowhere: Drift Without Drift

We encounter a surprisingly easy application of drift theory in the absence of drift. An example of a random process which does not exhibit any expected change (drift) is the *Gambler's Ruin* process. By considering a transformation of the process (essentially: squaring it) the process now exhibits drift in the order of its variance, which is then amenable to analysis with drift theory.

4.1 Unbiased Random Walks

We start by deriving two general corollaries, before we draw conclusions for specific random walks. The first, Theorem 4.1 [Unbiased Random Walk on the Line], concerns completely unbiased random walks. The second, Theorem 4.2 [Unbiased Random Walk on the Line, One Barrier], gives the situation for random walks with one barrier.

Theorem 4.1: Unbiased Random Walk on the Line

Let $n \in \mathbb{N}$, let $(X_t)_{t \in \mathbb{N}}$ be an integrable random process over $[0, n]$, and let $T = \inf\{t \in \mathbb{N} \mid X_t \in \{0, n\}\}$. Suppose that there is a $\delta \in \mathbb{R}_+$ such that, for all $t < T$, we have the following conditions (variance, drift).

(Var) $\mathrm{Var}[X_{t+1} - X_t \mid X_0, \ldots, X_t] = \delta$;

(D) $\mathrm{E}[X_{t+1} - X_t \mid X_0, \ldots, X_t] = 0$.

Then $\mathrm{E}[T] = \frac{\mathrm{E}[X_0(n-X_0)]}{\delta}$.

Proof. We consider the process $Y_t = X_t(n - X_t)$. Note that T is the first time $t \in \mathbb{N}$ such that $Y_t = 0$. In the following, we condition on $X_0, \ldots, X_t$, a filtration that $Y_0, \ldots, Y_t$ is adapted to; this allows us to apply our drift theorems by Theorem 8.6 [Conditioning on Filtration vs. History vs. Events] while giving information not only about the value of Y_t, but also about X_t. Furthermore, for all $s \in [1..n-1]$, we have

$$\begin{aligned}
\mathrm{E}[Y_t - Y_{t+1} \mid X_0, \ldots, X_t] &= \mathrm{E}\left[X_{t+1}^2 - X_t^2 \mid X_0, \ldots, X_t\right] - n\,\mathrm{E}[X_{t+1} - X_t \mid X_0, \ldots, X_t] \\
&= \mathrm{E}\left[X_{t+1}^2 \mid X_0, \ldots, X_t\right] - X_t^2 = \mathrm{Var}[X_{t+1} \mid X_0, \ldots, X_t] \\
&= \mathrm{Var}[X_{t+1} - X_t \mid X_0, \ldots, X_t] = \delta.
\end{aligned}$$

Thus, we have a drift of δ towards 0. Since $Y_0 = X_0(n - X_0)$, the theorem follows from an application of Theorem 2.1 [Additive Drift, Upper Bound]. ∎

Since the proof is based on the additive drift theorem, a lower bound of δ on the variance is enough for an upper bound on the expected first-hitting time and vice versa.

Theorem 4.2: Unbiased Random Walk on the Line, One Barrier

Let $n \in \mathbb{N}$, let $(X_t)_{t \in \mathbb{N}}$ be an integrable random process over $[0, n]$, and let $T = \inf\{t \in \mathbb{N} \mid X_t = n\}$. Suppose that there is a $\delta \in \mathbb{R}_+$ such that, for all $t < T$, we have the following conditions (variance, drift).

(Var) $\mathrm{Var}[X_{t+1} - X_t \mid X_0, \ldots, X_t] \geq \delta$;

(D) $\mathrm{E}[X_{t+1} - X_t \mid X_0, \ldots, X_t] \geq 0$.

Then $\mathrm{E}[T] \leq \frac{n^2 - \mathrm{E}\left[X_0^2\right]}{\delta}$.

Proof. We consider the process $Y_t = n^2 - X_t^2$. Note that T is the first time such that $Y_t = 0$. Further note that, from (D) we get, for all $t < T$,

$$\mathrm{E}[X_{t+1} \mid X_0, \dots, X_t]^2 \geq X_t^2. \tag{$*$}$$

As in the previous proof, we condition on $X_0, \dots, X_t$ and implicitly use Theorem 8.6 [Conditioning on Filtration vs. History vs. Events]. We now have

$$\begin{aligned}
\mathrm{E}[Y_t - Y_{t+1} \mid X_0, \dots, X_t] &= \mathrm{E}\left[X_{t+1}^2 - X_t^2 \mid X_0, \dots, X_t\right] \\
&= \mathrm{E}\left[X_{t+1}^2 \mid X_0, \dots, X_t\right] - X_t^2 \\
&\underset{(*)}{\geq} \mathrm{E}\left[X_{t+1}^2 \mid X_0, \dots, X_t\right] - \mathrm{E}[X_{t+1} \mid X_0, \dots, X_t]^2 \\
&= \mathrm{Var}[X_{t+1} \mid X_0, \dots, X_t] \\
&= \mathrm{Var}[X_{t+1} - X_t \mid X_0, \dots, X_t] \\
&\geq \delta.
\end{aligned}$$

Thus, we have a drift of at least δ towards 0. Since $Y_0 = n^2 - X_0^2$, the theorem follows from an application of Theorem 2.1 [Additive Drift, Upper Bound]. ■

Note that in neither of the two preceding theorems is the process allowed to overshoot the target. Using an additive drift theorem that allows for overshooting, like Theorem 5.3 [Additive Drift, Upper Bound with Overshooting], one can derive corresponding extensions of the above two theorems with essentially the same proof. We note that Theorem 4.2 [Unbiased Random Walk on the Line, One Barrier] is tight with the following example.

> **Example 4.3: Fair Random Walk** — Let $(X_t)_{t\in\mathbb{N}}$ be the time-homogeneous Markov-chain on $[0..n]$, where, for all $t \in \mathbb{N}$,
>
> (1) for all $i \in [1..n-1]$, $\Pr[X_{t+1} = i+1 \mid X_t = i] = 1/2 = \Pr[X_{t+1} = i-1 \mid X_t = i]$;
>
> (2) the state 0 is reflective, that is, $\Pr[X_{t+1} = 1 \mid X_t = 0] = 1$; and
>
> (3) the state n is absorbing.
>
> We transform X into the fair random walk $(Y_t)_{t\in\mathbb{N}}$ on $[0..2n]$, where the states 0 and $2n$ are both absorbing, such that, for all $t \in \mathbb{N}$, it holds that $X_t = |Y_t - n|$.
> Informally, we mirror X at 0 and then shift it by n. Whenever this new process is at n, it goes to either $n-1$ or $n+1$, each with probability $1/2$, which results exactly in Y. Note that $T = \inf\{t \in \mathbb{N} \mid Y_t \in \{0, 2n\}\} = \inf\{t \in \mathbb{N} \mid X_t = n\}$. Applying Theorem 4.1 [Unbiased Random Walk on the Line] to Y yields $\mathrm{E}[T] = \mathrm{E}[Y_0(2n - Y_0)]$. Since $X_0 \leq n$, it holds that $Y_0 = n - X_0$. Substituting this back into the equation for $\mathrm{E}[T]$ yields $\mathrm{E}[T] = \mathrm{E}[(n - X_0)(n + X_0)] = n^2 - \mathrm{E}\left[X_0^2\right]$, which is exactly the bound of Theorem 4.2 [Unbiased Random Walk on the Line, One Barrier].

4.2 Analysis of Concrete Unbiased Random Walks

In this section we see several domains in which we apply our theorems about unbiased random walks. The Gambler's Ruin in Theorem 4.4 [Gambler's Ruin] is the most straightforward application of Theorem 4.1 [Unbiased Random Walk on the Line]. A more intricate application is given in Theorem 4.5 [The Recolour Algorithm], where it is used to bound the expected run time for an algorithm to find a certain coloring of a graph.

Regarding Theorem 4.2 [Unbiased Random Walk on the Line, One Barrier], in Theorem 4.6 [Random 2-SAT] we use it to derive an upper bound on the time for an algorithm to find a satisfying assignment for a 2-SAT formula.

We start with the *gambler's ruin*, a random walk on the line. It starts at n, going either one step to the left or one step to the right, each with probability $1/2$, modeling winning or losing a fair coin toss to either win or lose a coin. The question of how long it takes to either be broke (0 coins left) or double the starting number of coins is the simplest setting of an unbiased random walk. This process also goes by many other names, such as *drunkard's walk*, *random walk on a line*, or *one-dimensional random walk*.

Theorem 4.4: Gambler's Ruin

Suppose we start with $n \in \mathbb{N}$ coins and, in each iteration, uniformly at random either gain a coin or lose a coin. Then, after an expected number of exactly n^2 iterations, we are either broke or have reached a total of $2n$ coins.

Proof. For all $t \in \mathbb{N}$, let X_t be the random sequence of the number of coins after t iterations. We have

$$\mathrm{E}[X_{t+1} - X_t \mid X_0, \ldots, X_t] = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot (-1) = 0.$$

Furthermore, we have

$$\mathrm{Var}[X_{t+1} - X_t \mid X_0, \ldots, X_t] = \frac{1}{2} \cdot 1^2 + \frac{1}{2} \cdot (-1)^2 = 1.$$

Thus, we can now apply Theorem 4.1 [Unbiased Random Walk on the Line] to get the desired result. ■

Our next example considers the analysis of the run time of an algorithm. McDiarmid [McD93] studies the following simple randomized algorithm called Recolour, for coloring a given undirected graph G with two colors such that it contains no monochromatic triangle (a subgraph on three pairwise connected vertices which are all colored with the same color). Recolour starts with an arbitrary 2-coloring of G. At every step, it checks whether the current coloring has a monochromatic triangle. If so, Recolour changes the color of one of the vertices of this triangle uniformly at random. Otherwise, the 2-coloring has no monochromatic triangles and it is the output of Recolour.

McDiarmid shows that, when Recolour is applied to a 3-colorable graph G (a graph that can be colored with three colors so that no two neighbors share a color), it returns a 2-coloring of G with no monochromatic triangle in expected time $O(n^4)$. His analysis shows that the expected run time of the algorithm is bounded above by the expected hitting time of a random walk on the line with two absorbing states – which is exactly the setting of Theorem 4.1 [Unbiased Random Walk on the Line]. This analysis in turn relies on previous results on one-dimensional random walks, which usually require lengthy calculations.

We present a simple and self-contained proof of the $O(n^4)$ expected run time of the Recolour algorithm for finding a 2-coloring with no monochromatic triangles on 3-colorable graphs. Our proof follows the proof of McDiarmid [McD93] to reduce the problem to an unbiased random walk on the line and then uses Theorem 4.1 [Unbiased Random Walk on the Line]. A similar analysis can be used to derive an upper bound on the run time of Recolour on hypergraph colorings.

Theorem 4.5: The Recolour Algorithm

The expected run time of Recolour on a 3-colorable graph with $n \in \mathbb{N}_+$ vertices is $O(n^4)$.

Proof. Let $G = (V, E)$ be a 3-colorable graph, and let $\chi \colon V \to \{1, 2, 3\}$ be a 3-coloring of G. Let $U = \{v \in V \mid \chi(v) \in \{1, 2\}\}$ be the set of all vertices which are colored with colors 1 and 2. Note that any 2-coloring of G

that agrees with χ on the vertices from U is a 2-coloring of G with no monochromatic triangles. Thus, the run time of RECOLOUR is bounded from above by the expected time that RECOLOUR takes to find such a coloring.

Let χ_t be the 2-coloring found by RECOLOUR at time $t \in \mathbb{N}$. Let Y_t be the number of vertices $u \in U$ such that $\chi_t(u) = \chi(u)$. The algorithm terminates when $Y_t \in \{0, |U|\}$, since agreeing on all vertices of U is a coloring without monochromatic triangles, but disagreeing on all vertices from U is also such a valid coloring, since the use of the colors is symmetric.

Let $s \in [1..|U| - 1]$ denote an outcome of Y_t before the algorithm terminates. We then have that $\Pr[Y_{t+1} = Y_t + 1 \mid Y_t = s] = 1/3$, as, for every monochromatic triangle, there is exactly one vertex in $u \in U$ with $\chi_t(u) \neq \chi(u)$ which can be recolored to obtain another vertex where the colors match. Similarly, $\Pr[Y_{t+1} = Y_t - 1 \mid Y_t = s] = 1/3$. Thus, Y_t is an unbiased random walk on the line with first-hitting time $T = \inf\{t \in \mathbb{N} \mid Y_t \in \{0, |U|\}\}$. We have

$$\mathrm{Var}[Y_{t+1} - Y_t \mid Y_0, \ldots, Y_t] = \frac{1}{3} \cdot 1^2 + \frac{1}{3} \cdot 0^2 + \frac{1}{3} \cdot (-1)^2 = \frac{2}{3}.$$

Applying Theorem 4.1 [Unbiased Random Walk on the Line] we get

$$\mathrm{E}[T] = \frac{3\,\mathrm{E}[Y_0(|U| - Y_0)]}{2} \leq \frac{3n^2}{8}.$$

At each step, the algorithm requires $O(n^2)$ time to find a monochromatic triangle and modify this to obtain a new coloring, which concludes the proof. ∎

The analysis of the RECOLOUR algorithm for finding 2-colorings with no monochromatic triangles appears as an exercise in [MU05].

In the final example of this section, we consider finding satisfying assignments of 2-SAT formulas. Papadimitriou [Pap91] studies the following simple randomized algorithm that returns a satisfying assignment of a satisfiable 2-SAT formula ϕ with n variables and m clauses within $O(n^2 m)$ time in expectation. The algorithm starts with a random assignment of the variables of ϕ. At every step, the algorithm checks whether there is an unsatisfied clause for this assignment. If so, the algorithm changes the assignment of one of the variables of this assignment uniformly at random. Otherwise, the assignment is satisfying and it is the output of the algorithm.

The analysis given is similar to the RECOLOUR algorithm, and it also relies on the previous results on one-dimensional random walks. An extensive analysis of this algorithm appears in [MU05]. Here, we present a simpler proof that uses Theorem 4.2 [Unbiased Random Walk on the Line, One Barrier].

Theorem 4.6: Random 2-SAT

The randomized 2-SAT algorithm, when run on a satisfiable 2-SAT formula over $n \in \mathbb{N}_+$ variables and $m \in \mathbb{N}_+$ clauses, terminates in $O(n^2 m)$ time in expectation.

Proof. Let ϕ be a satisfiable 2-SAT formula and a a satisfying assignment. At each time step $t \in \mathbb{N}_+$, the randomized 2-SAT algorithm finds a (not necessarily satisfying) assignment a_t. Let X_t be the random variable denoting the number of variables that have the same truth assignment in both a and a_t. Let T be the first time the algorithm reaches a satisfying assignment for ϕ. Assume that a clause $x \vee y$ is not satisfied by a_t. Since a is a satisfying assignment, a and a_t differ in the assignment of at least one of the variables in this clause. Thus,

$$\Pr[X_{t+1} = X_t + 1 \mid X_0, \ldots, X_t] \geq 1/2; \text{ and}$$
$$\Pr[X_{t+1} = X_t - 1 \mid X_0, \ldots, X_t] \leq 1/2.$$

When $a_t = a$, the algorithm terminates. By Theorem 4.2 [Unbiased Random Walk on the Line, One Barrier] with variance bounded by 1, we have $\mathrm{E}[T] \leq n^2$. In order to transition from a_t to a_{t+1}, the algorithm requires $O(m)$ time (since the 2-SAT formula has m distinct clauses), concluding the proof. ■

5 The Zoo: A Tour of Drift Theorems

We have seen the basic two drift theorems, the additive drift theorem and the multiplicative drift theorem, in Section 2 [A Gentle Introduction to Classic Drift Theorems]. In this section we provide a list of more advanced drift theorems with applications.

(1) In Section 5.1 [Additive Drift] we start by extending the additive drift theorem; we see how to avoid the requirement of non-negativity (allowing overshooting of the target) and explore different conditions for the drift.

(2) Section 5.2 [Additive Drift: Concentration] provides a different view on additive drift by considering concentration.

(3) In Section 5.3 [Multiplicative Drift] we give lower bounds for the case of multiplicative drift.

(4) While additive drift required drift to be constant and multiplicative drift required proportional drift, in Section 5.4 [Variable Drift] we give theorems allowing for an arbitrary *monotone* dependence of the drift on the current state.

(5) Somewhat different in flavor is Section 5.5 [Negative Drift]. Here we discuss drift theorems providing *exponential lower bounds* given drift away from the target.

(6) In Section 5.6 [Finite State Spaces] we consider the special case of random processes on finite search spaces.

(7) Some settings allow drift only when far away from the target, but in the proximity of the target the drift is negative. In this case, the theorem of Section 5.7 [Headwind Drift] can offer an upper bound on the run time nonetheless.

(8) In order to derive good upper bounds even when the drift gets stronger when getting closer to the optimum, Section 5.8 [Multiplicative Up-Drift] provides a drift theorem for the case of proportionally increasing drift.

(9) A completely different approach to understanding drift is given by Wormald and briefly discussed in Section 5.9 [Wormald's Method].

Note that there are a few novel approaches to analyzing multi-dimensional potential functions [Row18, JL22]; while the initial works are promising, they have not gained traction yet and we will not discuss them here.

5.1 Additive Drift

We want to start with an illustrative proof for a strong version of the additive drift theorem; the proof is adapted from the proof of Theorem 2.3.1 in [Len20].

Theorem 5.1: Additive Drift, Upper Bound, Time Condition

Let $(X_t)_{t\in\mathbb{N}}$ be a stochastic process on $\mathbb{R}$ with deterministic X_0, and let $T = \inf\{t \in \mathbb{N} \mid X_t \leq 0\}$. Suppose that there is a $\delta > 0$ so that we have the following conditions (drift, non-negativity).

(D) For all t with $\Pr[t < T] > 0$, $\mathrm{E}[X_t - X_{t+1} \mid t < T] \geq \delta$.

(NN) For all $t \leq T$, $X_t \geq 0$.

We have
$$E[T] \leq X_0/\delta.$$

Proof. We first replace the process $(X_t)_{t\in\mathbb{N}}$ with a process $(X'_t)_{t\in\mathbb{N}}$ such that, for all $t \leq T$, we have $X'_t = X_t$, and for all $t > T$ we have $X'_t = X'_{t-1}$. Both processes are ≤ 0 at the same time, but $(X'_t)_{t\in\mathbb{N}}$ does not change after that. We have that (NN) gives $X_T = 0$, and, thus, $X_t = 0$ for all $t \geq T$. We will from now on assume that $(X_t)_{t\in\mathbb{N}}$ is exactly this modified process. Together with (NN) we thus have

$$\forall t \in \mathbb{N} : X_t \geq 0. \tag{NN'}$$

Furthermore, for all $t \geq T$ with $\Pr[t \geq T] > 0$, we have $E[X_t - X_{t+1} \mid t \geq T] = 0$.

We now have, for all $t \in \mathbb{N}$ with $\Pr[t < T] > 0$,

$$\begin{aligned} E[X_t - X_{t+1}] &= E[X_t - X_{t+1} \mid t < T] \Pr[t < T] + E[X_t - X_{t+1} \mid t \geq T] \Pr[t \geq T] \\ &= \Pr[t < T]\, E[X_t - X_{t+1} \mid t < T] \\ &\underset{(D)}{\geq} \Pr[t < T]\delta \\ &= \delta \Pr[T > t]. \end{aligned}$$

The first equality is the law of total expectation; the second follows from $X_t = X_{t+1}$ for $t \geq T$. Note that the overall inequality holds trivially for all $t \in \mathbb{N}$ such that $\Pr[t < T] = 0$, so it holds for all t. Explicitly, for all $t \in \mathbb{N}$ we have

$$\Pr[T > t] \leq \frac{1}{\delta} E[X_t - X_{t+1}]. \tag{$*$}$$

Since T takes only values in $\mathbb{N} \cup \{\infty\}$, we have

$$E[T] = \sum_{i=0}^{\infty} \Pr[T > i].$$

We want to use this to compute $E[T]$. For all $n \in \mathbb{N}$ we have

$$\sum_{t=0}^{n} \Pr[T > t] \underset{(*)}{\leq} \frac{1}{\delta} \sum_{t=0}^{n} (E[X_t] - E[X_{t+1}]) = \frac{1}{\delta}(X_0 - E[X_{n+1}]) \underset{(NN')}{\leq} \frac{X_0}{\delta}.$$

Since all partial sums are upper bounded by X_0/δ, so is the infinite sum. ■

Note that we can turn the proof around to get the analogous version for a lower bound. Again the proof is essentially taken from the proof of Theorem 1 in [Len20]. Note that it uses the somewhat strong assumption of a bounded search space, whereas Theorem 2.3 [Additive Drift, Lower Bound] only requires a bound on the size of each step.

Theorem 5.2: Additive Drift, Lower Bound, Time Condition

Let $(X_t)_{t\in\mathbb{N}}$ be a stochastic process on $\mathbb{R}$ with deterministic X_0, and let $T = \inf\{t \in \mathbb{N} \mid X_t \leq 0\}$. Suppose that there is a $\delta \in \mathbb{R}_+$ so that we have the following conditions (drift, upper bounded search space).

(D) For all t with $\Pr[t < T] > 0$, $E[X_t - X_{t+1} \mid t < T] \leq \delta$.

(UB) There is a $c > 0$ such that, for all $t < T$, $X_t \leq c$.

We have

$$E[T] \geq X_0/\delta.$$

Proof. We first replace the process $(X_t)_{t\in\mathbb{N}}$ with a process $(X'_t)_{t\in\mathbb{N}}$ such that, for all $t \leq T$, we have $X'_t = X_t$, and for all $t > T$ we have $X'_t = X'_{t-1}$. Both processes are ≤ 0 at the same time, but $(X'_t)_{t\in\mathbb{N}}$ does not change after that. We will from now on assume that $(X_t)_{t\in\mathbb{N}}$ is exactly this modified process. Thus, for all $t \geq T$ with $\Pr[t \geq T] > 0$ we have $E[X_t - X_{t+1} \mid t \geq T] = 0$.

We now have, for all $t \in \mathbb{N}$ with $\Pr[t < T] > 0$,

$$\begin{aligned} E[X_t - X_{t+1}] &= E[X_t - X_{t+1} \mid t < T] \Pr[t < T] + E[X_t - X_{t+1} \mid t \geq T] \Pr[t \geq T] \\ &= \Pr[t < T]\, E[X_t - X_{t+1} \mid t < T] \\ &\underset{\text{(D)}}{\leq} \Pr[t < T]\delta \\ &= \delta \Pr[T > t]. \end{aligned}$$

The first equality is the law of total expectation; the second follows from $X_t = X_{t+1}$ for $t \geq T$. Note that the overall inequality holds trivially for all $t \in \mathbb{N}$ such that $\Pr[t < T] = 0$, so it holds for all t. Explicitly, for all $t \in \mathbb{N}$ we have

$$\Pr[T > t] \geq \frac{1}{\delta}\, E[X_t - X_{t+1}]. \tag{$*$}$$

Since T takes only values in $\mathbb{N} \cup \{\infty\}$, we have

$$E[T] = \sum_{i=0}^{\infty} \Pr[T > i].$$

We want to use this to compute $E[T]$. For all $n \in \mathbb{N}$, we have

$$\sum_{t=0}^{n} \Pr[T > t] \underset{(*)}{\geq} \frac{1}{\delta} \sum_{t=0}^{n} (E[X_t] - E[X_{t+1}]) = \frac{1}{\delta}(X_0 - E[X_{n+1}]).$$

It remains to be shown that $E[X_{n+1}]$ converges to a value ≤ 0 for n going to infinity. Using the c from (UB), we have for all $n \in \mathbb{N}$ with $\Pr[n < T] > 0$ that

$$\begin{aligned} E[X_n] &= E[X_n \mid n < T] \Pr[n < T] + E[X_n \mid n \geq T] \Pr[n \geq T] \leq c \cdot \Pr[n < T] + 0 \cdot \Pr[n \geq T] \\ &= c \Pr[n < T]. \end{aligned}$$

We distinguish two cases. If $\Pr[n < T]$ converges to 0 for n going to ∞, then $E[X_n]$ converges to 0 as desired. Otherwise, there is a non-zero probability of $T = \infty$, in which case the theorem follows directly from that. ∎

Both in Theorem 2.1 [Additive Drift, Upper Bound], the classic version of the additive drift theorem, as in the version just above, it required that the target of 0 must be hit exactly and not overshot (NN). From [Kre19] we have a stronger version that allows for overshooting. This is frequently helpful, for example for finding approximations, when potentially much better values than required can be achieved.

Theorem 5.3: Additive Drift, Upper Bound with Overshooting

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable process over $\mathbb{R}$, and let $T = \inf\{t \in \mathbb{N} \mid X_t \leq 0\}$. Furthermore, suppose the following (drift).

(D) There is a $\delta > 0$ such that, for all $t < T$, it holds that $\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq \delta$.

Then

$$\mathrm{E}[T] \leq \frac{\mathrm{E}[X_0] - \mathrm{E}[X_T]}{\delta}.$$

Proof. Consider the process $(X'_t)_{t\in\mathbb{N}}$ such that, for all $t \in \mathbb{N}$, $X'_t = X_t - X_T$. ■

In a sense, this drift theorem is simpler than Theorem 2.1 [Additive Drift, Upper Bound]: the requirement (NN) is dropped and the expected time is increased corresponding to the expected additional distance the process will have traveled (note that $\mathrm{E}[X_T]$ is not a positive value, since T is the first point t where $X_t \leq 0$). From the condition (NN) we could derive $X_T = 0$ and thus immediately recover Theorem 2.1 [Additive Drift, Upper Bound].

5.2 Additive Drift: Concentration

One of the reasons why the additive drift theorem is so general (we only really have a requirement on the expectation of change, the first moment, but not on the higher moments) is that we only get a conclusion about the expectation of the first hitting time of the target. With requirements on the higher moments we can derive concentration bounds on the expected first hitting time. This is provided by [Köt16], from which we give two different variants, one using an absolute bound on the step size (B), Theorem 2 in the cited work, and one requiring concentrated step size (C), combining Theorems 10 and 15 from the cited work. Each time we get that there is only a very small probability of arriving significantly later than the expected time of n/δ. An example application of such a concentration result for additive drift is in [KLW15] regarding an analysis of the (1 + 1) EA on a dynamic version of OneMax (see Theorem 10 in the cited work). Another application is given in Theorem 5 of [FKL+17, FKL+20].

Theorem 5.4: Additive Drift, Upper Concentration, Bounded Step Size

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable process over $\mathbb{R}$, and let $T = \inf\{t \in \mathbb{N} \mid X_t \leq 0\}$. Furthermore, suppose that there is $c > 0$ such that we have the following conditions (drift, bounded steps).

(D) There is a $\delta > 0$ such that, for all $t < T$, it holds that $\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq \delta$.

(B) For all $t \in \mathbb{N}$, $|X_{t+1} - X_t| \leq c$.

Let $n \in \mathbb{N}$ such that $X_0 \leq n$. Then, for all $s \geq 2n/\delta$,

$$\Pr[T \geq s] \leq \exp\left(-\frac{s\delta^2}{8c^2}\right).$$

Theorem 5.5: Additive Drift, Upper Concentration, Concentrated Step Size

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable random process over $\mathbb{R}$, and let $T = \inf\{t \in \mathbb{N} \mid X_t \leq 0\}$. Furthermore, suppose that there are $\varepsilon > 0$ and $c > 0$ such that we have the following conditions (drift, concentration).

(D) There is a $\delta > 0$ such that, for all $t < T$, it holds that $E[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq \delta$.

(C) For all $t \in \mathbb{N}$ and all $x \geq 0$, $\Pr[|X_{t+1} - X_t| \geq x \mid X_t] \leq \frac{c}{(1+\varepsilon)^x}$.

Let $n \in \mathbb{N}$ such that $X_0 \leq n$. Then, for all $s \geq 2n/\delta$,

$$\Pr[T \geq s] \leq \exp\left(-\frac{s\delta}{4} \min\left(\frac{\varepsilon}{4}, \frac{\delta\varepsilon^3}{256c}\right)\right).$$

Also in [Köt16] are analogous *lower* bounds. Again we give two different variants, one using an absolute bound on the step size (B), Theorem 1 in the cited work, and one requiring concentrated step size (C), combining Theorems 10 and 14 from the cited work. Each time we get that there is only a very small probability of arriving significantly before the expected time of n/δ.

Theorem 5.6: Additive Drift, Lower Concentration, Bounded Step Size

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable random process over $\mathbb{R}$, and let $T = \inf\{t \in \mathbb{N} \mid X_t \leq 0\}$. Furthermore, suppose that there is $c > 0$ such that we have the following conditions (drift, bounded steps).

(D) There is a $\delta > 0$ such that, for all $t < T$, it holds that $E[X_t - X_{t+1} \mid X_0, \ldots, X_t] \leq \delta$.

(B) For all $t \in \mathbb{N}$, $|X_{t+1} - X_t| \leq c$.

Let $n \in \mathbb{N}$ such that $X_0 \geq n$. Then, for all $s \leq n/(2\delta)$,

$$\Pr[T < s] \leq \exp\left(-\frac{n^2}{8c^2 s}\right).$$

Theorem 5.7: Additive Drift, Lower Concentration, Concentrated Step Size

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable random process over $\mathbb{R}$, and let $T = \inf\{t \in \mathbb{N} \mid X_t \leq 0\}$. Furthermore, suppose that there are $\varepsilon > 0$ and $c > 0$ such that (drift, concentration)

(D) there is a $\delta > 0$ such that, for all $t < T$, it holds that $E[X_t - X_{t+1} \mid X_0, \ldots, X_t] \leq \delta$;

(C) for all $t \in \mathbb{N}$ and all $x \geq 0$, $\Pr[|X_{t+1} - X_t| \geq x \mid X_t] \leq \frac{c}{(1+\varepsilon)^x}$.

Let $n \in \mathbb{N}$ such that $X_0 \geq n$. Then, for all $s \leq n/(2\delta)$,

$$\Pr[T < s] \leq \exp\left(-\frac{n}{4} \min\left(\frac{\varepsilon}{4}, \frac{n\varepsilon^3}{256cs}\right)\right).$$

The overall situation depending on the strength of the drift is depicted in detail in [Köt16]. In particular, there are three main regimes:

(1) If the drift is at least $\delta \geq 1/n$, then we get high concentration of the first hitting time.

(2) If the drift is $\delta \in [-1/n, 1/n]$ but the variance is significant, then we get to hit the optimum with constant chance within $O(n^2)$ steps, see Theorem 4.2 [Unbiased Random Walk on the Line, One Barrier].

(3) If the drift is much smaller than $-1/n$, then we have negative drift and only a superpolynomially small chance to reach the optimum in polynomial time, see Theorem 5.15 [Negative Drift, Bounded Step Size].

The literature knows also the following theorem for bounding additive drift only relying on the variance, given by Semenov and Terkel [ST03].

Theorem 5.8: Additive Drift, Upper Concentration, Bounded Variance

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable random process over $\mathbb{R}$ with $X_0 = 0$. Furthermore, suppose the following (drift, variance).

(D) There is a $\delta > 0$ such that, for all $t < T$, it holds that $\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq \delta$.

(Var) There is a $c > 0$ such that, for all $t \in \mathbb{N}$, $\mathrm{Var}[X_{t+1} \mid X_0, \ldots, X_t] \leq c$.

Then, for all $\varepsilon > 0$, the following holds with probability 1.

$$X_t \geq t\delta - o\left(t^{0.5+\varepsilon}\right).$$

5.3 Multiplicative Drift

The plain multiplicative drift theorem (see Theorem 2.5 [Multiplicative Drift]) is already very strong, in that it requires few conditions on the search space and even gives a concentration (in one direction). What it does not provide is a lower bound. One possible such bound can be found in [Wit13] which we state here.

Theorem 5.9: Multiplicative Drift, Lower Bound, Monotone

Let $(X_t)_{t\in\mathbb{N}}$ be a discrete, integrable process over $\{0,1\} \cup S$, where $S \subset \mathbb{R}_{>1}$ is finite, and let $T = \inf\{t \in \mathbb{N} \mid X_t \leq 0\}$.
We assume that there are $\beta, \delta \in (0,1)$ such that the following conditions (drift, monotonicity, concentration) hold for all $s > 1$ and $t \in \mathbb{N}$ with $\Pr[X_t = s] > 0$.

(D) $\mathrm{E}[X_t - X_{t+1} \mid X_t = s] \leq \delta s$.

(M) $X_{t+1} \leq X_t$.

(C) $\Pr[X_t - X_{t+1} \geq \beta s \mid X_t = s] \leq \beta\delta/\ln(s)$.

Then

$$\mathrm{E}[T \mid X_0] \geq \frac{\ln(X_0)}{\delta} \cdot \frac{1-\beta}{1+\beta} \geq \frac{\ln(X_0)}{\delta} \cdot (1-2\beta).$$

From [DDK18] we have a variant which allows for non-monotone drift. It substitutes the monotonicity with the requirement that we cannot expect more progress from first returning to bigger values of the process. Turned around, progress in any state s cannot be bigger than in a state $s' < s$. We use the notation $(x)_+ := \max(0, x)$.

Theorem 5.10: Multiplicative Drift, Lower Bound

Let $(X_t)_{t\in\mathbb{N}}$ be a discrete random process over $\{0,1\} \cup S$, where $S \subset \mathbb{R}_{>1}$ is finite, and let $T = \inf\{t \in \mathbb{N} \mid X_t \leq 1\}$.
We assume that there are $\beta, \delta \in (0,1)$ such that the following conditions (drift, concentration) hold for all $s > 1$ and $t \in \mathbb{N}$ with $\Pr[X_t = s] > 0$.

(D) For all s' with $1 < s' \leq s$: $\mathrm{E}[(s' - X_{t+1})_+ \mid X_0, \ldots, X_t, X_t = s] \leq \delta s'$.

(C) For all s' with $1 < s' \leq s$: $\Pr[s' - X_{t+1} \geq \beta s' \mid X_0, \ldots, X_t, X_t = s] \leq \beta\delta/\ln(s')$.

Then
$$E[T \mid X_0] \geq \frac{\ln(X_0)}{\delta} \cdot \frac{1-\beta}{1+\beta} \geq \frac{\ln(X_0)}{\delta} \cdot (1-2\beta).$$

As an alternative, we can find a lower bound when the step size is bounded. The following theorem is given in [DKLL20]. A further version can be found in [KK19].

Theorem 5.11: Multiplicative Drift, Lower Bound, Bounded Step Size

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable random process over $\mathbb{R}_+$, let $x_{\min} > 0$, and let $T = \inf\{t \in \mathbb{N} \mid X_t \leq x_{\min}\}$. We assume that there are $c, \delta \in \mathbb{R}_+$ with $x_{\min} \geq \sqrt{2}c$ such that the following conditions (drift, bounded step size) hold for all $t < T$.

(D) $E[X_t - X_{t+1} \mid X_0, \ldots, X_t] \leq \delta X_t$.

(B) $|X_t - X_{t+1}| \leq c$.

Then
$$E[T \mid X_0] \geq \frac{1 + \ln(X_0) - \ln(x_{\min})}{2\delta + \frac{c^2}{x_{\min}^2 - c^2}}.$$

Note that, for typical applications, δ is small; yet the term 2δ should dominate the term $\frac{c^2}{x_{\min}^2 - c^2}$ to give a tight bound. But this is typically not a problem: consider the setting of $\delta = \Theta(1/n)$ and $X_0 = \Theta(n)$. We can let $x_{\min} = \Theta(\sqrt{n})$ and suppose we can bound $c = o(\sqrt{n})$ with sufficiently high probability (which would be typical). Then the theorem lets us derive the asymptotically optimal bound of $\Omega(n \log n)$.

As an example application we provide a lower bound for the coupon collector process (see the upper bound proven in Theorem 2.7 [Coupon Collector with Multiplicative Drift]).

Theorem 5.12: Coupon Collector, Lower Bound

Suppose we want to collect at least one of each color of $n \in \mathbb{N}_{\geq 1}$ coupons. Each round, we are given one coupon with a color chosen uniformly at random from the n kinds. Then, in expectation, we have to collect for at least $\Omega(n \ln n)$ iterations.

Proof. Let X_t be the number of coupons missing after t iterations. We want to apply Theorem 5.11 [Multiplicative Drift, Lower Bound, Bounded Step Size] and note that, since each iteration at most one coupon is gained and none is lost, we can use $c = 1$ to satisfy (B). Furthermore, regarding (D), the probability of making progress (of 1) with coupon $t+1$ is X_t/n. Thus, $E[X_t - X_{t+1} \mid X_0, \ldots, X_t] = X_t/n$ and we can use $\delta = 1/n$. We set $x_{\min} = \sqrt{n}$ and an application of Theorem 5.11 [Multiplicative Drift, Lower Bound, Bounded Step Size] gives an upper bound of
$$\frac{1 + \ln(n) - \ln(\sqrt{n})}{\frac{2}{n} + \frac{1}{n-1}} \geq \frac{1 + \ln(n)/2}{\frac{3}{n-1}} = \frac{1}{6} \cdot (n-1)\ln(n) = \Omega(n \ln n).$$

■

5.4 Variable Drift

A more general version of Theorem 2.1 [Additive Drift, Upper Bound] and Theorem 2.5 [Multiplicative Drift] is the variable drift theorem, allowing for any *monotone* dependency of the drift on the current state (meaning that a larger distance to the target has to imply a larger drift). It is due to [MRC09, Joh10] and was improved in [RS14]. We give here the version from [KK19], where the random process is not assumed to be discrete or Markovian.

Theorem 5.13: Variable Drift

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable random process over $\mathbb{R}$, $x_{\min} \in \mathbb{R}_+$, and let $T = \inf\{t \in \mathbb{N} \mid X_t < x_{\min}\}$. Additionally, let I denote the smallest real interval that contains at least all values $x \geq x_{\min}$ that, for all $t \leq T$, any X_t can take. Furthermore, suppose that there is a function $h\colon I \to \mathbb{R}_+$ such that the following conditions (drift, monotonicity, start, non-negativity) hold for all $t \leq T$.

(D) $\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq h(X_t)$.

(M) The function h is monotonically non-decreasing.

(S) $X_0 \geq x_{\min}$.

(NN) $X_t \geq 0$.

Then

$$\mathrm{E}[T \mid X_0] \leq \frac{1}{h(x_{\min})} + \int_{x_{\min}}^{X_0} \frac{1}{h(z)}\,\mathrm{d}z.$$

Note that the additive drift theorem is the special case of constant h and the multiplicative drift theorem is the special case of linear h. It is surprising that the cases of additive and multiplicative drift are sufficient in many applications, but the variable drift theorem can also in these cases sometimes give tighter bounds.

Concentration bounds for variable drift are also available [LW14]. The same paper also gives a lower bounding variable drift theorem, which requires h to be monotonically non-*decreasing*, the opposite as for the upper bound. Further variants can be found in [KK19], including lower bounds for step-size bounded settings.

An example application of Theorem 5.13 [Variable Drift] is the optimization of LeadingOnes by the (1 + 1) EA. It is known [DJW02] that the expected gain in fitness value per iteration, given that the current fitness value is $n - s$ (and thus s away from the optimum), is (essentially) at least

$$h(s) = 2 \cdot (1 - 1/n)^{n-s} \cdot \frac{1}{n}.$$

The middle term is the probability to not lose a bit already gained; the $1/n$ is the probability to flip the left-most 0 and the 2 is the expected fitness gained when the two just mentioned events happen (one bit flipped, plus an expected one more bit that happens to be correctly set). The middle term can be lower-bounded by $1/e$, which allows for applying Theorem 2.1 [Additive Drift, Upper Bound], giving a total run time of at most

$$\frac{e}{2} \cdot n^2.$$

Using the variable drift theorem directly on the bound given by h, a simple integration gives an upper bound on the optimization time of

$$\frac{e-1}{2} \cdot n^2.$$

This optimization time was first established in [BDN10]. In this example, the use of the variable drift theorem improved the leading constant. Note that the bound can be derived also by other means, for example by the fitness level method, which can also be used to show tightness of this bound, see Theorem 6.7 [Run Time of (1 + 1) EA on LeadingOnes].

An essential application of a variable drift is given in the proof of Theorem 17 in [DFF+19], considering the optimization of OneMax by an islands-based evolutionary algorithm, employing λ islands and an exchange of individuals every τ rounds. In particular, the considered drift function is $h\colon \mathbb{R} \to \mathbb{R}$ such that, for all $s > 0$,

$$h(s) = \ln(\lambda) \Big/ \ln\left(\frac{n \ \ln(\lambda)}{\tau s}\right)$$

The final bound on the run time is shown to be asymptotically tight, thanks to using both upper and lower bounding variable drift theorems.

Further uses of the variable drift theorem are given in Theorem 7 of [FKL+17, FKL+20] and in Theorem 6 of [DDK16, DDK18].

5.5 Negative Drift

When the drift goes away from the target, we speak of *negative drift*. The negative drift theorem [OW11, OW12] gives an exponential lower bound in this setting.

Theorem 5.14: Negative Drift

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable random process over $\mathbb{R}$. Suppose there is an interval $[a, b] \subseteq \mathbb{R}$, two constants $\delta, \varepsilon > 0$ and, possibly depending on $\ell = b - a$, a function $r(\ell)$ satisfying $1 \leq r(\ell) = o(\ell/\log \ell)$ such that, for all $t \in \mathbb{N}$, the following conditions (drift, concentration) hold.

(D) $\mathrm{E}[X_{t+1} - X_t \mid X_0, \ldots, X_t; a < X_t < b] \geq \delta$.

(C) For all $j \in \mathbb{N}$, $\Pr[|X_{t+1} - X_t| \geq j \mid X_0, \ldots, X_t; a < X_t] \leq \frac{r(\ell)}{(1+\varepsilon)^j}$.

Then there is a constant c such that, for $T = \min\{t \in \mathbb{N} \mid X_t \leq a\}$, we have

$$\Pr\left[T \leq 2^{c\ell/r(\ell)} \;\middle|\; X_0 \geq b\right] = 2^{-\Omega(\ell/r(\ell))}.$$

Note that drift goes with a strength *independent of the width $\ell = b - a$ of the interval* away from the target a (while the process is in the interval). A version with scaling which allows for more flexibility in this dependence is given in [OW14].

A variant that allows for arbitrary ε (with decaying guarantees) is given in [Köt16] as follows. It requires a bounded step size, but in return gives a very simple and easy-to-apply bound.

Theorem 5.15: Negative Drift, Bounded Step Size

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable random process over $\mathbb{R}$, each with finite expectation, and let $n > 0$. Let $T = \min\{t \in \mathbb{N} \mid X_t \geq n\}$ and suppose there are $0 < c < n$ and $\varepsilon < 0$ such that, for all $t \in \mathbb{N}$, the following conditions hold (drift, boundedness).

(D) $\mathrm{E}[X_{t+1} - X_t \mid X_0, \ldots, X_t] \leq \varepsilon$.

(B) $|X_{t+1} - X_t| < c$.

Then, for all $s \in \mathbb{N}$, we have

$$\Pr[T \leq s] = s \exp\left(-\frac{n|\varepsilon|}{2c^2}\right).$$

Given as Corollary 22 in [Köt16] is a second variant of the negative drift theorem. It allows for very large r while still giving a super-polynomial bound for finding the target in polynomial time.

Theorem 5.16: Negative Drift II

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable random process over $\mathbb{R}$. Suppose there is an interval $[a,b] \subseteq \mathbb{R}$, two constants $\delta, \varepsilon > 0$ and, possibly depending on $\ell = b - a$, a function $r(\ell)$ satisfying $1 \leq r(\ell) = \exp(o(\sqrt[4]{\ell}))$ such that, for all $t \in \mathbb{N}$, the following conditions hold (drift, concentration).

(D) $E[X_{t+1} - X_t \mid X_0, \ldots, X_t; a < X_t < b] \geq \varepsilon$.

(C) For all $j \in \mathbb{N}$, $\Pr[|X_{t+1} - X_t - \varepsilon| \geq j \mid X_0, \ldots, X_t; a < X_t] \leq \frac{r(\ell)}{(1+\delta)^j}$.

Then there is a constant c such that, for $T = \min\{t \in \mathbb{N} \mid X_t \leq a\}$, we have

$$\Pr\left[T \leq 2^{c\sqrt{\ell}} \,\middle|\, X_0 \geq b\right] = 2^{-\Omega(\sqrt[4]{\ell})}.$$

Example applications of negative drift theorems for the analysis of evolutionary algorithms are given in the proofs of the following statements. Lemma 3 of [KM12]; in Lemma 8 of [FKKS15a] (see also Lemma 5 of [FKKS17]); Theorem 3 of [FKKS17]; Lemma 6 [FKS16]; and Lemma 13 [FKK16].

5.6 Finite State Spaces

Most drift theorems consider a random walk on the real numbers, sometimes restricted to non-negative numbers. For the analysis of discrete algorithms, frequently the state space is even more restricted, in particular finite. By numbering the successive states, we can assume the state space to be $[0..n]$. For this setup we have the following drift theorem from [KK18]. Note that the proof given in the paper is derived by the method of step-wise differences, see Section 3.4 [Designing a Potential Function via Step-Wise Differences]. The theorem generalizes a theorem from [DJW00].

Theorem 5.17: Finite State Spaces, Upper Bound

Let $(X_t)_{t\in\mathbb{N}}$ be a time-homogeneous Markov chain on $[0..n]$ and let T be the first time t such that $X_t = 0$. Suppose there are two functions $p^{\leftarrow}\colon [1..n] \to [0,1]$ and $p^{\rightarrow}\colon [0..n-1] \to [0,1]$ such that, for all $t < T$ and all $s \in [1..n]$,

(1) $p^{\leftarrow}(s) > 0$,

(2) $\Pr[X_t - X_{t+1} \geq 1 \mid X_t = s] \geq p^{\leftarrow}(s)$,

(3) $\Pr[X_t - X_{t+1} = -1 \mid X_t = s] \leq p^{\rightarrow}(s)$ (for $s \neq n$), and

(4) $\Pr[X_t - X_{t+1} < -1 \mid X_t = s] = 0$ (for $s \neq n$).

Then

$$E[T \mid X_0] \leq \sum_{s=1}^{X_0} \sum_{i=s}^{n} \frac{1}{p^{\leftarrow}(i)} \prod_{j=s}^{i-1} \frac{p^{\rightarrow}(j)}{p^{\leftarrow}(j)}.$$

A special case of this theorem is the fitness level method (see Theorem 6.1 [Fitness Level Method (FLM)]), where the process is monotone; we can recover this setting by setting $p^{\rightarrow}$ to be constantly 0, significantly simplifying the above formula.

We also have the corresponding lower bound.

Theorem 5.18: Finite State Spaces, Lower Bound

Let $(X_t)_{t\in\mathbb{N}}$ be a time-homogeneous Markov chain on $[0..n]$ and let T be the first time t such that $X_t = 0$. Suppose there are two functions $p^{\leftarrow}\colon \{1,\ldots,n\} \to [0,1]$ and $p^{\rightarrow}\colon [0..n-1] \to [0,1]$ such that, for all $t < T$ and all $s \in [1..n]$,

(1) $p^{\leftarrow}(s) > 0$,

(2) $\Pr[X_t - X_{t+1} = 1 \mid X_t = s] \leq p^{\leftarrow}(s)$,

(3) $\Pr[X_t - X_{t+1} > 1 \mid X_t = s] = 0$, and

(4) $\Pr[X_t - X_{t+1} \leq -1 \mid X_t = s] \geq p^{\rightarrow}(s)$ (for $s \neq n$).

Then

$$\mathrm{E}[T \mid X_0] \geq \sum_{s=1}^{X_0} \sum_{i=s}^{n} \frac{1}{p^{\leftarrow}(i)} \prod_{j=s}^{i-1} \frac{p^{\rightarrow}(j)}{p^{\leftarrow}(j)}.$$

Note that for processes which make steps of at most 1 and given exact $p^{\rightarrow}$ and $p^{\leftarrow}$, the two bounds coincide.

5.7 Headwind Drift

Sometimes drift only carries until shortly before the target, but then, close to the target, turns negative. In case only a small remaining distance needs to be bridged, and the probability of going the right way is still sufficiently high, the following *Headwind* drift theorem can be used to directly get a decent bound without relying on hand crafted potential functions. The theorem was developed and applied in [KLW15].

Theorem 5.19: Headwind Drift

Let $(X_t)_{t\in\mathbb{N}}$ be a time-homogeneous Markov chain on $[0..n]$. Let bounds

$$p^{-}(i) \leq \Pr[X_{t+1} \leq i-1 \mid X_t = i]$$

and

$$p^{+}(i) \geq \Pr[X_{t+1} \geq i+1 \mid X_t = i],$$

where $0 \leq i \leq n$, be given, and define

$$\delta(i) := p^{-}(i) - \mathrm{E}[(X_{t+1} - i) \cdot \mathbb{1}[X_{t+1} > i] \mid X_t = i].$$

Assume that $\delta(i)$ is monotone increasing with respect to i and let $\kappa \geq \max\{i \geq 0 \mid \delta(i) \leq 0\}$ (noting that $\delta(0) \leq 0$). The function $g\colon [0..n+1] \to \mathbb{R}_{+}$ is defined by

$$g(i) := \sum_{k=i+1}^{n} \frac{1}{\delta(k)}$$

for $i \geq \kappa$ (in particular, $g(n) = g(n+1) = 0$), and inductively by

$$g(i) := \frac{1 + (p^+(i+1) + p^-(i+1))g(i+1)}{p^-(i+1)}$$

for $i < \kappa$.
Then it holds for the first hitting time $T := \min\{t \in \mathbb{N} \mid X_t = 0\}$ of state 0 that

$$\mathrm{E}[T \mid X_0] \leq g(0) - g(X_0).$$

We can also get a closed expression for the expected first hitting time $\mathrm{E}[T \mid X_0]$. This expression involves the factor $\sum_{k=\kappa+1}^{N} \frac{1}{\delta(k)}$ that is reminiscent of the formula for the expected first hitting time of state κ under variable drift towards the target (see Theorem 5.13 [Variable Drift]). For the states less than κ, where drift away from the target holds, the product $\prod_{k=1}^{\kappa} \frac{p^+(k)+p^-(k)}{p^-(k)}$ comes into play. Intuitively, it represents the waiting time for the event of taking κ consecutive steps against the drift. Since the product involves probabilities conditioned on leaving the states, which effectively removes self-loops, another sum of products must be added. This sum, represented by the second line of the expression for $\mathrm{E}[T \mid X_0]$, intuitively accounts for the self-loops.

Corollary 5.20: Headwind Drift, Closed Form

Let the assumptions of Theorem 5.19 [Headwind Drift] hold. Then

$$\begin{aligned}\mathrm{E}[T \mid X_0] \leq &\left(\left(\sum_{k=\kappa+1}^{N} \frac{1}{\delta(k)}\right)\left(\prod_{k=1}^{\kappa} \frac{p^+(k)+p^-(k)}{p^-(k)}\right)\right) \\ &+ \left(\sum_{k=1}^{\kappa} \frac{1}{p^-(k)} \prod_{j=1}^{k-1} \frac{p^+(j)+p^-(j)}{p^-(j)}\right).\end{aligned}$$

Note that there is a similarity between the theorems for headwind drift and those from Section 5.6 [Finite State Spaces]. This is because the analysis for the last steps of headwind drift is essentially an analysis brute-forcing the small interval of negative drift, which also happens in Section 5.6 [Finite State Spaces].

5.8 Multiplicative Up-Drift

The idea of Theorem 2.5 [Multiplicative Drift] was to have multiplicative drift going *down* towards 0. While this has many applications (owing to the fact that progress in optimization typically gets harder as better and better solutions are found), there are also a number of processes that gain in speed over time, typically making progress proportional to the current state of the process, such as rumor spreading, epidemics and population take-over. This is known as multiplicative up-drift and was studied in depth in [DK21b]. The main theorem is the following.

Theorem 5.21: Multiplicative Up-Drift

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable random process over $\mathbb{Z}_{\geq 0}$. Let $n, k \in \mathbb{Z}_{\geq 1}$, $E_0 > 0$, $\gamma_0 < 1$, and $\delta > 0$ such that $n - 1 \leq \min\{\gamma_0 k, (1+\delta)^{-1}k\}$. Let $D_0 = \min(\lceil 100/\delta \rceil, n)$ when $\delta \leq 1$ and $D_0 = \min(32, n)$ otherwise. Assume that, for all $t \in \mathbb{N}$ and all $x \in [0..n-1]$ with $\Pr[X_t = x] > 0$, the following two conditions hold (binomial distribution, gain at 0); note that we use the concept of stochastic dominance.

(Bin) If $x \geq 1$, then $(X_{t+1} \succeq \mathrm{Bin}(k, (1+\delta)X_t/k)$.

(0) $\mathrm{E}[\min(X_{t+1}, D_0) \mid X_t = 0] \geq E_0$.

Let $T := \min\{t \in \mathbb{N} \mid X_t \geq n\}$.
Then, **if $\delta \leq 1$**,

$$\mathrm{E}[T] \leq \frac{4D_0}{0.4088E_0} + \frac{15}{1-\gamma_0} D_0 \ln(2D_0) + 2.5 \log_2(n) \lceil 3/\delta \rceil.$$

In particular, when γ_0 is bounded away from 1 by a constant, then $E[T] = O\left(\frac{1}{E_0\delta} + \frac{\log(n)}{\delta}\right)$, where the asymptotic notation refers to n tending to infinity and where $\delta = \delta(n)$ may be a function of n. Furthermore, if $n > 100/\delta$, then we also have that once the process has reached a state of at least $100/\delta$, the probability to ever return to a state of at most $50/\delta$ is at most 0.5912.
If $\delta > 1$, then we have

$$\mathrm{E}[T] \leq \frac{128}{0.78E_0} + 2.6 \log_{1+\delta}(n) + 81 = O\left(\frac{1}{E_0} + \frac{\log(n)}{\log(\delta)}\right).$$

In addition, once the process has reached state 32 or higher, the probability to ever return to a state lower than 32 is at most $\frac{1}{e(e-1)} < 0.22$.

Note that this drift theorem is essentially restricted to processes based on the binomial distribution. For many applications this restriction is satisfied, particularly for the level-based theorem introduced in [Leh11] and refined in [DL16, CDEL18]. We now discuss the currently strongest version in terms of the asymptotics in δ, given in [DK21b] as a consequence to the multiplicative up-drift theorem.

The general setup of level-based theorems is as follows. There is a ground set $\mathcal{X}$, which in typical applications is the search space of an optimization problem. On this ground set, a Markov chain (P_t) induced by a population-based EA is defined. We consider populations of fixed size λ, which may contain elements several times (multi-sets). We write $\mathcal{X}^\lambda$ to denote the set of all such populations. We only consider Markov chains where each element of the next population is sampled independently with repetition. That is, for each population $P \in \mathcal{X}^\lambda$, there is a distribution $D(P)$ on $\mathcal{X}$ such that given P_t, the next population P_{t+1} consists of λ elements of $\mathcal{X}$, each chosen independently according to the distribution $D(P_t)$. As all our results hold for any initial population P_0, we do not make any assumptions on P_0.

In the level-based setting, we assume that there is a partition of $\mathcal{X}$ into levels $A_1, \ldots, A_m$ (leading to the name of a *level-based* theorem). Based on information in particular on how individuals in higher levels are generated, we aim for an upper bound on the first time such that the population contains an element of the highest level A_m.

Theorem 5.22: Level-Based Theorem

Consider a population-based process as described above.
Let $(A_1, \ldots, A_m)$ be a partition of $\mathcal{X}$. Let $A_{\geq j} := \bigcup_{i=j}^m A_i$ for all $j \in [1..m]$. Let $z_1, \ldots, z_{m-1}, \delta \in (0, 1]$, and let $\gamma_0 \in (0, \frac{1}{1+\delta}]$ with $\gamma_0\lambda \in \mathbb{Z}$. Let $D_0 = \min\{\lceil 100/\delta \rceil, \gamma_0\lambda\}$ and $c_1 = 56\,000$. Let

$$t_0 = \frac{7000}{\delta}\left(m + \frac{1}{1-\gamma_0} \sum_{j=1}^{m-1} \log_2^0\left(\frac{2\gamma_0\lambda}{1 + \frac{z_j\lambda}{D_0}}\right) + \frac{1}{\lambda} \sum_{j=1}^{m-1} \frac{1}{z_j}\right),$$

where $\log_2^0(x) := \max(0, \log_2(x))$ for all $x \in \mathbb{R}_+$. Assume that, for any population $P \in \mathcal{X}^\lambda$, the following three conditions are satisfied (drift, zero condition, population size).

(D) For each level $j \in [1..m-2]$ and all $\gamma \in (0, \gamma_0]$, if $|P \cap A_{\geq j}| \geq \gamma_0\lambda/4$ and $|P \cap A_{\geq j+1}| \geq \gamma\lambda$, then

$$\Pr_{y \sim D(P)}\left[y \in A_{\geq j+1}\right] \geq (1+\delta)\gamma.$$

(0) For each level $j \in [1..m-1]$, if $|P \cap A_{\geq j}| \geq \gamma_0\lambda/4$, then

$$\Pr_{y \sim D(P)}\left[y \in A_{\geq j+1}\right] \geq z_j.$$

(PS) The population size λ satisfies

$$\lambda \geq \frac{256}{\gamma_0\delta}\ln(8t_0).$$

Then $T := \min\{\lambda t \mid P_t \cap A_m \neq \emptyset\}$ satisfies

$$\mathrm{E}[T] \leq 8\lambda t_0 = c_1\frac{\lambda}{\delta}\left(m + \frac{1}{1-\gamma_0}\sum_{j=1}^{m-1}\log_2^0\left(\frac{2\gamma_0\lambda}{1+\frac{z_j\lambda}{D_0}}\right) + \frac{1}{\lambda}\sum_{j=1}^{m-1}\frac{1}{z_j}\right).$$

Note that, with $z^* = \min_{j \in [1..m-1]} z_j$ and γ_0 a constant, (PS) in the previous theorem is satisfied for some λ with

$$\lambda = \Omega\left(\frac{1}{\delta}\log\left(\frac{m}{\delta z^*}\right)\right)$$

as well as for all larger λ.

5.9 Wormald's Method

A very different approach to understanding random processes via their step-wise changes is given by Wormald [Wor99], tracking the processes via solutions of a system of differential equations. We briefly state a version of this theorem here.

Consider a stochastic process $(Y^{(t)})_{t \in \mathbb{N}}$, where each random variable $Y^{(t)}$ takes values in some set S. We use H_t to denote a history of the process up to time t, i.e. $H_t = (Y^{(0)}, \ldots, Y^{(t)})$. And S^+ denotes the set of all sequences $(Y^{(0)}, \ldots, Y^{(t)})$ such that $Y^{(t)} \in S$.

We say that a function $f \colon \mathbb{R}^k \to \mathbb{R}$ satisfies a Lipschitz condition on $D \subseteq \mathbb{R}^k$ if there is an $L > 0$ such that, for all $u = (u_1, \ldots, u_k), v = (v_1, \ldots, v_k) \in D$,

$$|f(u) - f(v)| \leq L \max_{1 \leq i \leq k} |u_i - v_i|.$$

Theorem 5.23: Wormald's Method

For some $a \in \mathbb{N}$, let $(Y_i^{(t)})_{1 \leq i \leq a, t \in \mathbb{N}}$ be a stochastic process, such that there is $C \in \mathbb{R}_+$ so that for all $m \in \mathbb{N}_+$ and $t \in \mathbb{N}$, $|Y_i^{(t)}| < m$ for all $H_t \in S^+$. Let D be some bounded connected open set containing the closure of

$$\left\{(0, z_1, \ldots, z_a)\middle|\Pr\left[Y_i^{(0)} = z_i m, 1 \leq i \leq a\right] \neq 0 \text{ for some } m\right\}.$$

Assume the following three conditions hold, where for each $1 \leq i \leq a$ function $f_i \colon \mathbb{R}_+ \times \mathbb{R}^a \to \mathbb{R}$ is continuous, and satisfies a Lipschitz condition on D with the same Lipschitz constant L for all i (drift, boundedness).

(D) $\mathrm{E}\left[Y_i^{(t+1)} - Y_i^{(t)} \mid H_t\right] = f_i(t/m, Y_1^{(t)}/m, \ldots, Y_a^{(t)}/m)$.

(B) For all $t \in \mathbb{N}$, $\max_{1 \leq i \leq a} |Y_i^{(t+1)} - Y_i^{(t)}| \leq 1$.

Then the following are true.

(1) For any $(0, \hat{z}_1, \ldots, \hat{z}_a) \in D$, the system of differential equations

$$\frac{dz_i}{dx} = f_i(x, z_1, \ldots, z_a), \; i = 1, \ldots, a$$

has a unique solution in D for $z_i : \mathbb{R} \to \mathbb{R}$ passing through $z_i(0) = \hat{z}_i, 1 \leq i \leq a$, and which extends to points arbitrarily close to the boundary of D;

(2) Let $\lambda = \lambda(m) = o(1)$. For some constant $C > 0$, with probability $1 - O\left(\frac{1}{\lambda}\exp(-m\lambda^3)\right)$,

$$Y_i^{(t)} = mz_i(t/m) + O(\lambda m)$$

uniformly for $0 \leq t \leq \sigma m$ and for each i, where $z_i(x)$ is the solution in given above with $\hat{z}_i = \frac{1}{m} Y_i^{(0)}$, and $\sigma = \sigma(m)$ is the supremum of those x to which the solution can be extended before reaching within L_∞-distance $C\lambda$ of the boundary of D.

A few analyzes of randomized search heuristics with Wormald's theorem exist [LS15, FKM17, Her18]. It has the advantage that it allows to track multiple interacting random variables (which, for other drift theorems, would have to be combined to a single potential). On the other hand, it requires solving a differential equation (well-known to be not an easy task) and the conclusion is typically deteriorating over time, since the variance is not averaged out but accumulates over time.

Note that there are also theorems closer to the classic drift theorems for tracking multiple random variables in restricted settings [Row18, JL22].

6 No Going Back: The Fitness Level Method (FLM)

Some processes $(X_t)_{t\in\mathbb{N}}$ are *monotone*, that is, we have $\forall t : X_t \leq X_{t+1}$. Monotone processes occur frequently in the analysis of heuristic optimization, since the best fitness found so far is a typical process considered. For some such processes, simpler (and sometimes stronger) analyses are possible than with drift theorems allowing for non-monotone processes (see Proposition 6.2 [Application to Rumor Spreading] for an example).

Wegener [Weg01] proposed the following method, called the *fitness level method* (FLM). We partition the search space into a number m of sections ("levels") in a linear fashion, so that all elements of later levels have better fitness than all elements of earlier levels. For the algorithm to be analyzed we regard the best-so-far individual and the level it is in. Since the best-so-far individual can never move to lower levels, it will visit each level at most once (possibly staying there for some time). Suppose we can show that, for any level $i < m$ which the algorithm is currently in, the probability to leave this level is at least p_i. Then, bounding the expected waiting for leaving a level i by $1/p_i$ (geometric distribution) and pessimistically assuming that we visit (and thus have to leave) each level $i < m$ before reaching the target level m, we can derive an upper bound for the optimization time of

$$\sum_{i=1}^{m-1} \frac{1}{p_i}.$$

The fitness level method allows for simple and intuitive proofs and has therefore frequently been applied. Variations of it come with tail bounds [Wit14], work for parallel EAs [LS14] or regard populations [Wit06]. A similar analysis in levels can be made for non-elitist EAs, but here it is crucially possible (and sometimes not unlikely) to lose a level. See Theorem 5.22 [Level-Based Theorem] for a corresponding theorem along a discussion.

We state the fitness level method (FLM) formally as follows.

Theorem 6.1: Fitness Level Method (FLM)

Let $(X_t)_{t\in\mathbb{N}}$ be a monotone process on $[m]$. For all $i \in [m-1]$, let p_i be a lower bound on the probability of a state change of $(X_t)_{t\in\mathbb{N}}$, conditional on being in state i, formally: for all t with $\Pr[X_t = i] > 0$,

$$\Pr[X_{t+1} > i \mid X_0, \ldots, X_t, X_t = i] \geq p_i.$$

Let T be the random variable describing the first time t such that $X_t = m$. Then

$$\mathrm{E}[T] \leq \sum_{i=1}^{m-1} \frac{1}{p_i}.$$

Proof. For all $i \in [m-1]$, we let $S_i = \{t \in \mathbb{N} \mid X_t = i\}$. Since $(X_t)_{t\in\mathbb{N}}$ is monotone, each S_i is a discrete interval. Calling the leaving of S_i a *success event*, we can use Theorem 2.9 [Geometric Distribution] to see that the expected size of each S_i is at most $1/p_i$. Since $T = \sum_{i=1}^{m-1} |S_i|$, the theorem follows. ∎

> **Note.** The main strength of the fitness level method over drift theorems is that the chance to leave a level i, p_i, can depend arbitrarily on i. In contrast, in Theorem 5.13 [Variable Drift], one of the most general drift theorems, the drift has to depend monotonically on the state of the process.
> Conversely, the main strength of drift theorems over the fitness level method is that the *process* is allowed to be non-monotone, so that we can work with other processes than those based on fitness.

The following example shows a toy application of the fitness level method where drift theorems are not easily applicable. The setting is borrowed from the area of *rumor spreading*, see, for example, [DK14] and also

[DFF+19] for an application in the area of randomized search heuristics.

Propostion 6.2: Application to Rumor Spreading

Let $n \in \mathbb{N}$. Suppose n people each want to obtain a certain information, and suppose in iteration 0 exactly one of them knows this information. In each iteration, one of the n people is chosen uniformly at random and if this person does not know the information, the person will contact another person chosen uniformly at random. If this other person knows the information, then the calling person from now on also knows the information. Then it takes, in expectation, at most $2n(\ln(n-1)+1)$ iterations until all persons know the rumor.

Proof. We let, for each $t \in \mathbb{N}$, X_t be the number of persons who know the rumor after t iterations. Then $(X_t)_{t\in\mathbb{N}}$ is a monotone process on $[n]$. Let some iteration t be given. If, after t iterations, exactly $i \in [n-1]$ persons know the rumor, then the probability that in the next iteration an uninformed person is chosen to make a call is $(n-i)/n$. The probability that this person calls an informed person is independent of that probability and $i/(n-1)$. Thus, the chance p_i to "leave state i" is

$$p_i = \frac{n-i}{n}\,\frac{i}{n-1}.$$

By Theorem 6.1 [Fitness Level Method (FLM)], the total time until all people are informed is thus at most

$$\sum_{i=1}^{n-1} \frac{1}{p_i} = \sum_{i=1}^{n-1} \frac{n(n-1)}{(n-i)i}.$$

[Comment: For didactic reasons, we give two different ways of bounding this sum; the second uses calculus and gives the better bound.] A first way to bound the sum is by splitting it into two and using worst case estimates to simplify. Let $k = \lfloor n/2 \rfloor$; we have

$$\begin{aligned}
\sum_{i=1}^{n-1} \frac{n(n-1)}{(n-i)i} &= \sum_{i=1}^{k} \frac{n(n-1)}{(n-i)i} + \sum_{i=k+1}^{n-1} \frac{n(n-1)}{(n-i)i} \\
&\leq \sum_{i=1}^{k} \frac{n(n-1)}{(n-k)i} + \sum_{i=k+1}^{n-1} \frac{n(n-1)}{(n-i)k} \\
&= \sum_{i=1}^{k} \frac{n(n-1)}{(n-k)i} + \sum_{i=1}^{n-k-1} \frac{n(n-1)}{i\,k} \\
&= \frac{n(n-1)}{n-k} \sum_{i=1}^{k} \frac{1}{i} + \frac{n(n-1)}{k} \sum_{i=1}^{n-k-1} \frac{1}{i} \\
&\leq \frac{n(n-1)}{n-k}(\ln(k)+1) + \frac{n(n-1)}{k}(\ln(n-k-1)+1).
\end{aligned}$$

The last inequality uses Lemma 9.11 [Upper Bound on the Harmonic Sum]. By bounding $1/(n-k) \leq 2/n$ and $1/k \leq 2/(n-1)$ we can further bound the term by

$$2(n-1)(\ln(k)+1) + 2n(\ln(n-k-1)+1) \leq 4n(\ln(n)+1).$$

We can get a tighter bound as follows by turning to calculus. The function

$$f\colon [1, n-1] \to \mathbb{R}, x \mapsto \frac{1}{(n-x)x}$$

has a minimum at $n/2$, is before that monotone decreasing and afterwards monotone increasing. Thus, we can use Lemma 9.12 [Upper Bounding Sum by Integral] twice, once on the interval $[1, \lfloor n/2 \rfloor]$ and once on $[\lceil n/2 \rceil, n-1]$ to bound

$$\sum_{i=1}^{n-1} \frac{n(n-1)}{(n-i)i} \leq n + n + n(n-1) \int_1^{n-1} f(x)\mathrm{d}x.$$

Note that the summands before the integral are the first and the last summand of the large sum (which are not covered by the cited lemma). The indefinite integral over f is given by the function

$$x \mapsto \frac{\ln(x) - \ln(n-x)}{n},$$

which can be seen by taking the derivative of that function. Using the integral bounds, we arrive at

$$\sum_{i=1}^{n-1} \frac{n(n-1)}{(n-i)i} \leq 2n + (n-1)(\ln(n-1) - \ln(1) - \ln(1) + \ln(n-1)) = 2(n-1)\ln(n-1) + 2n$$

as desired. ■

Note that due to the drift being strongest in the middle of the state space, no other drift theorem is directly available without losing asymptotically: one could apply the additive drift theorem to obtain a bound of $O(n^2)$ by using that the drift is $\Omega(1/n)$. Another choice is to us an argument in phases: since the process is monotone, one can analyze the time until reaching $n/2$ separately from the remainder. This would then potentially allow to use some version of multiplicative up-drift (for the first phase) and multiplicative drift (for the second phase). However, this would lead to an unnecessarily complicated analysis.

While very effective for proving upper bounds, it seems much harder to use fitness level arguments to prove lower bounds. The first to devise a lower bound method based on fitness levels that gives competitive bounds was Sudholt [Sud13]. Next we see a lower bound from [DK21a].

Theorem 6.3: Fitness Level Method with Visit Probabilities, Lower Bound

Let $(X_t)_{t \in \mathbb{N}}$ be a monotone process on $[m]$. For all $i \in [m-1]$, let p_i be an upper bound on the probability of a state change of $(X_t)_{t \in \mathbb{N}}$, conditional on being in state i. Furthermore, let v_i be a lower bound on the probability of there being a t such that $X_t = i$ (the *visit probability* of level i). Then the expected time for $(X_t)_{t \in \mathbb{N}}$ to reach the state m is

$$\mathrm{E}[T] \geq \sum_{i=1}^{m-1} \frac{v_i}{p_i}.$$

Proof. We proceed as in the proof for Theorem 6.1 [Fitness Level Method (FLM)]. For all $i \in [m-1]$, we let $S_i = \{t \in \mathbb{N} \mid X_t = i\}$. With probability at most $1 - v_i$ we have that $S_i = \emptyset$. Again using Theorem 2.9 [Geometric Distribution], we see that the expected size of each non-empty S_i is at least $1/p_i$. Since $T = \sum_{i=1}^{m-1} |S_i|$, the theorem follows with linearity of expectation. ■

A corresponding upper bound [DK21a] follows with analogous arguments and shows the tightness of the approach, with the bounds required on p_i and v_i reversed.

Theorem 6.4: Fitness Level Method with Visit Probabilities, Upper Bound

Let $(X_t)_{t\in\mathbb{N}}$ be a monotone process on $[m]$. For all $i \in [m-1]$, let p_i be an lower bound on the probability of a state change of $(X_t)_{t\in\mathbb{N}}$, conditional on being in state i. Furthermore, let v_i be an upper bound on the probability of there being a t such that $X_t = i$. Then the expected time for $(X_t)_{t\in\mathbb{N}}$ to reach the state m is

$$\mathrm{E}[T] \leq \sum_{i=1}^{m-1} \frac{v_i}{p_i}.$$

Proof. Analogous to the proof of Theorem 6.3 [Fitness Level Method with Visit Probabilities, Lower Bound]. ■

In a typical application of the fitness level method, finding good estimates for the leaving probabilities is easy. It is more complicated to estimate the visit probabilities accurately, the following lemma from [DK21a] offers an option.

Lemma 6.5: Computing Visit Probabilities

Let $(X_t)_{t\in\mathbb{N}}$ be a monotone process on $[m]$. Further, suppose that $(X_t)_{t\in\mathbb{N}}$ reaches state m after a finite time with probability 1. Let $i < m$ be given. Suppose there is v_i such that, for all $t \in \mathbb{N}$ with $\Pr[X_{t+1} \geq i > X_t] > 0$,

$$\Pr[X_{t+1} = i \mid X_0, \dots, X_t; X_{t+1} \geq i > X_t] \geq v_i,$$

and

$$\Pr[X_0 = i \mid X_0 \geq i] \geq v_i.$$

Then v_i is a lower bound for visiting level i as required by Theorem 6.3 [Fitness Level Method with Visit Probabilities, Lower Bound].

An analogous bound for upper bounds on visit probabilities also holds.

6.1 Applications

As a first application of these methods we now determine a lower bound for the coupon collector problem (see Theorem 2.7 [Coupon Collector with Multiplicative Drift]).

Theorem 6.6: Coupon Collector, Lower Bound via Fitness Levels

Suppose we want to collect at least one of each kind of $n \in \mathbb{N}_{\geq 1}$ coupons. Each round, we are given one coupon chosen uniformly at random from the n kinds. Then, in expectation, we have to collect for at least $n(1 + \ln n)$ iterations.

Proof. Let X_t be the number of coupons after t iterations. Note that this process is monotone and, since it gains at most one in any iteration, visits all elements of $[n-1]$ before reaching the target of n. The probability of making progress (of 1) with coupon $t+1$ is $p_i = (n-i)/n$, since $n-i$ coupons are missing and each has a probability of $1/n$, and all these events are disjoint. An application of both Theorem 6.3 [Fitness Level Method with Visit Probabilities, Lower Bound] and Theorem 6.4 [Fitness Level Method with Visit Probabilities, Upper Bound] gives an exact value of the expected time to find all n coupons of

$$\sum_{i=0}^{n-1} \frac{v_i}{p_i} = \sum_{i=0}^{n-1} \frac{n}{n-i} = nH_n,$$

Where, for each $m \in \mathbb{N}_+$, we use H_m to denote the mth harmonic number. ■

As a second application we now determine the precise run time of the $(1+1)$ EA on LeadingOnes via the two fitness level theorems. This optimization time was first established in [BDN10].

Theorem 6.7: Run Time of $(1+1)$ EA on LeadingOnes

Consider the $(1+1)$ EA optimizing LeadingOnes with mutation rate p. Let T be the (random) time for the $(1+1)$ EA to find the optimum. Then

$$\mathrm{E}[T] = \frac{1}{2}\sum_{i=0}^{n-1} \frac{1}{(1-p)^i p}.$$

Proof. We want to apply Theorem 6.4 [Fitness Level Method with Visit Probabilities, Upper Bound] and Theorem 6.3 [Fitness Level Method with Visit Probabilities, Lower Bound] simultaneously. For all $t \in \mathbb{N}$, we let X_t be the LeadingOnes-value of the individual which the $(1+1)$ EA has found after t iterations. Now we need a precise result for the probability to leave a level and for the probability to visit a level.

First, we consider the probability p_i to leave a given level $i < n$. Suppose the algorithm has a current search point in level i, so it has i leading 1s and then a 0. The algorithm leaves level A_i now if and only if it flips the first 0 of the bit string (probability of p) and no previous bits (probability $(1-p)^i$). Hence, $p_i = p(1-p)^i$.

Next we consider the probability v_i to visit a level i. We claim that it is exactly $1/2$, following reasoning given in several places before [DJW02, Sud13]. We want to use Lemma 6.5 [Computing Visit Probabilities] and its analogue for upper bounds. Let i be given. For the initial search point, if it is at least on level i (the condition considered by the lemma), the individual is on level i if and only if the $i+1$st bit is a 0, so exactly with probability $1/2$ as desired for both bounds. Before an individual with at least i leading 1s is created, the bit at position $i+1$ remains uniformly random (this can be seen by induction: it is uniform at the beginning and does not experience any bias in any iteration while no individual with at least i leading 1s is created). Once such an individual is created, if the bit at position $i+1$ is 1, the level i is skipped, otherwise it is visited. Thus, the algorithm skips level i with probability exactly $1/2$, giving $v_i = 1/2$. With these exact values for the p_i and v_i, Theorem 6.4 [Fitness Level Method with Visit Probabilities, Upper Bound] and Theorem 6.3 [Fitness Level Method with Visit Probabilities, Lower Bound] immediately yield the claim. ■

By computing the geometric series in Theorem 6.7 [Run Time of $(1+1)$ EA on LeadingOnes], we obtain as a (well-known) corollary that the $(1+1)$ EA with the classic mutation rate $p = 1/n$ optimizes LeadingOnes in an expected run time of $n^2 \frac{e-1}{2}(1 \pm o(1))$.

7 A Different Perspective: Fixed Budget Optimization

In the previous chapters we have seen many theorems regarding the *first hitting time* of a process. This answers the question: "How much time do I have to invest until a desired outcome?" Sometimes we want to answer a different question: "I have a *fixed budget* t_0 of time available; what performance can I expect?" Furthermore, fixed-budget results that hold with high probability are crucial for the analysis of algorithm configurators [HOS19]. These configurators test different algorithms for fixed budgets in order to make statements about their appropriateness in a given setting.

In this chapter we want to discuss general tools for fixed-budget analyses. We still want to use knowledge about step-wise changes and translate them into the global view, just as for the drift theorems for first hitting times.

We start by analyzing the most basic setting in Section 7.1 [The Additive Case] and generalize it in Section 7.2 [Variable Fixed Budget Drift]. We show sample results derived with these methods in Section 7.3 [Applications to OneMax and LeadingOnes].

7.1 The Additive Case

We start with the simple case of additive drift. If we expect to go down by δ in each iteration, then, after t iterations, we expect to be down $t\delta$, as would be the case for a *completely deterministic* process. Here the proof is simple and instructive.

Theorem 7.1: Additive Fixed-Budget Drift

Let $(X_t)_{t\in\mathbb{N}}$, be an integrable random process on $\mathbb{R}$. Suppose there is a $\delta \in \mathbb{R}_+$ so that we have the drift condition

(D) $\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq \delta$.

Thus, the drift condition is equivalent to

(D') $\mathrm{E}[X_{t+1} \mid X_0, \ldots, X_t] \leq X_t - \delta$.

Then, for all $t \geq 0$,

$$\mathrm{E}[X_t \mid X_0] \leq X_0 - t\delta.$$

Proof. We prove the theorem by induction on t, with a trivial induction basis. Suppose now the statement is true for some $t \geq 0$ (IH). Using the law of total expectation (LTE), we have

$$\begin{aligned} \mathrm{E}[X_{t+1} \mid X_0] &\underset{\text{(LTE)}}{=} \mathrm{E}[\mathrm{E}[X_{t+1} \mid X_0, \ldots, X_t] \mid X_0] \\ &\underset{\text{(D')}}{\leq} \mathrm{E}[X_t - \delta \mid X_0] \\ &\underset{\text{(IH)}}{\leq} X_0 - (t+1)\delta. \end{aligned}$$

This concludes the induction. ■

Note that this version does not take into account that drift might only hold before a target has been reached. We refer to this setting as *unlimited time*. The next theorem considers a potential end point. Note that the proof follows the proof of Theorem 1 in [Len20].

Theorem 7.2: Additive Fixed-Budget Drift, Limited Time

Let $(X_t)_{t\in\mathbb{N}}$, be an integrable random process on $\mathbb{R}$ and let T be any random variable on $\mathbb{N}$. Suppose that there is a $\delta \in \mathbb{R}_+$ so that we have the following drift condition.

(D) For all $t \in \mathbb{N}$ with $\Pr[t < T] > 0$, $\mathrm{E}[X_t - X_{t+1} \mid t < T] \geq \delta$.

(D') For all $t \in \mathbb{N}$ with $\Pr[t \geq T] > 0$, $\mathrm{E}[X_t - X_{t+1} \mid t \geq T] \geq 0$.

Then, for all $t \in \mathbb{N}$,

$$\mathrm{E}[X_t \mid X_0] \leq X_0 - t\delta \Pr[t \leq T].$$

Proof. First, we show that, for all $t \in \mathbb{N}$,

$$\mathrm{E}[X_{t+1} - X_t \mid X_0] \leq -\delta \Pr[t < T]. \tag{$*$}$$

We distinguish three cases. (1) If $\Pr[t < T] = 0$, then Equation ($*$) follows from (D'). (2) If $\Pr[t < T] = 1$, then Equation ($*$) follows from (D). (3) Otherwise, we use the law of total expectation to get

$$\begin{aligned} \mathrm{E}[X_{t+1} - X_t \mid X_0] &= \mathrm{E}[X_{t+1} - X_t \mid X_0, t < T]\Pr[t < T] + \mathrm{E}[X_{t+1} - X_t \mid X_0, t \geq T]\Pr[t \geq T] \\ &\underset{\text{(D)}}{\leq} -\delta \Pr[t < T] + \mathrm{E}[X_{t+1} - X_t \mid X_0, t \geq T]\Pr[t \geq T] \\ &\underset{\text{(D')}}{\leq} -\delta \Pr[t < T]. \end{aligned}$$

We now prove the theorem by induction on $t \in \mathbb{N}$, with a trivial induction basis for $t = 0$. Suppose now the statement is true for some $t \geq 0$ (IH). We now have

$$\begin{aligned} \mathrm{E}[X_{t+1} \mid X_0] &= \mathrm{E}[X_{t+1} - X_t + X_t \mid X_0] \\ &= \mathrm{E}[X_{t+1} - X_t \mid X_0] + \mathrm{E}[X_t \mid X_0] \\ &\underset{\text{(IH)}}{\leq} \mathrm{E}[X_{t+1} - X_t \mid X_0] + X_0 - t\delta \Pr[t \leq T] \\ &\underset{(*)}{\leq} -\delta \Pr[t < T] + X_0 - t\delta \Pr[t \leq T] \\ &\leq -\delta \Pr[t + 1 \leq T] + X_0 - t\delta \Pr[t + 1 \leq T] \\ &= X_0 - (t + 1)\delta \Pr[t + 1 \leq T]. \end{aligned}$$

This concludes the induction. ∎

7.2 Variable Fixed Budget Drift

For the rest of this chapter, we want to generalize Theorem 7.1 [Additive Fixed-Budget Drift] to *state*-varying drift. Suppose that, for some function h, in state x we observe a drift of $h(x)$. In order to understand what kind of result to expect in this context, we consider a completely deterministic process starting in $x_0 \in \mathbb{R}$ and progressing down by $h(x)$ when in state x. Then, after one step, the process is in $x_0 - h(x_0)$, after two steps in $x_0 - h(x_0) - h(x_0 - h(x_0))$ and so on. We write, for all x, $\tilde{h}(x) = x - h(x)$. Thus we can write the sequence of states of the process as $x_0, \tilde{h}(x_0), \tilde{h}(\tilde{h}(x_0))$ and so on. We write $\tilde{h}^t$ for the t-fold application of $\tilde{h}$, so after t steps of the process the state is $\tilde{h}^t$. Thus, we want a theorem that shows that we get a similar expected value for a probabilistic process.

The main question is now what we need to assume about h to get a behavior similar to the deterministic process. Consider the following monotone process on $\{0, 1, 2\}$: X_0 is 2 and the process moves to one of $\{0, 1\}$

uniformly. State 0 is the target state, from state 1 there is only a very small probability to progress to 0 (say 0.1). Then it is better to stay in State 2 instead of being trapped in State 1. Here the drift is 1.5 in State 2 and only 0.1 in State 1. Thus, the *expected next state* for State 2 is 0.5, which is less than the expected next state for State 1, which is 0.9! Intuitively, greedily going forward is a bad idea, if given the choice between States 1 and 2 one should choose (non-greedily) State 2. It turns out that forbidding this kind of situation, formalized in the next definition, leads to a viable generalization of Theorem 7.1 [Additive Fixed-Budget Drift].

Definition 7.3: Greed-Admitting Functions

We say that a drift function $h\colon S \to \mathbb{R}_{>0}$ is *greed-admitting* if and only if $\mathrm{id}-h$ (the function $x \mapsto x-h(x)$) is monotone non-decreasing.

Intuitively, this formalizes the idea that being closer to the goal is always better ("greed is good"). The process described before the definition is, in a sense, badly designed: State 1 is worse than State 2, so it should not have a smaller value.

We now give two different versions of fixed-budget drift theorems. The first considers *unlimited time*, a very strong requirement, leading to a strong conclusion.

Theorem 7.4: Variable Fixed-Budget Drift, Unlimited Time

Let $(X_t)_{t\in\mathbb{N}}$, be an integrable random process on $S \subseteq \mathbb{R}$, where $0 = \min S$. Let $h\colon S \to \mathbb{R}_{\geq 0}$ be a *twice differentiable, convex and greed-admitting* function such that $\tilde{h}'(0) \in\,]0,1]$ and we have the drift condition

(D-ut) $\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq h(X_t)$.

Define $\tilde{h}(x) = x - h(x)$. Thus, the drift condition is equivalent to

(D-ut') $\mathrm{E}[X_{t+1} \mid X_0, \ldots, X_t] \leq \tilde{h}(X_t)$.

Then, for all $t \geq 0$,

$$\mathrm{E}[X_t \mid X_0] \leq \tilde{h}^t(X_0)$$

and, in particular,

$$\mathrm{E}[X_t] \leq \tilde{h}^t(\mathrm{E}[X_0]).$$

Crucial for this theorem is that the drift condition is *unlimited time*, by which we mean that the drift condition has to hold for all times t, not just (which is the typical case in the literature for drift theorems) those before the optimum is hit. This theorem is applicable if there is no optimum (and the optimization progresses indefinitely) or if the drift is 0 in the optimum. In order to bypass these limitations we also give a variant which allows for *limited time* drift, where the drift condition only needs to hold before the optimum is hit; however, in this case we pick up an additional error term in the result, derived from the possibility of hitting the optimum within the allowed time budget of t. Thus, in order to apply this theorem, one will typically need concentrations bounds for the time to hit the optimum.

A special case of the previous theorem is given in [LS15], where the drift is necessarily multiplicative. Note that in this case we can typically consider unlimited time, since after reaching the state 0 the multiplicative drift holds vacuously.

Now we give a version of the variable fixed-budget drift where the time is limited in the sense that the drift condition might no longer hold at some point in time.

Theorem 7.5: Variable Fixed-Budget Drift, Limited Time

Let $(X_t)_{t \in \mathbb{N}}$, be an integrable random process on $S \subseteq \mathbb{R}$, where $0 = \min S$. Let $T = \inf\{t \in \mathbb{N} \mid X_t = 0\}$ and $h\colon S \to \mathbb{R}_{\geq 0}$ be a *twice differentiable, convex and greed-admitting* function such that $\tilde{h}'(0) \in]0, 1]$ and we have, for all $t < T$, the drift condition

(D-lt) $\mathrm{E}[X_t - X_{t+1} \mid X_0, \dots, X_t] \geq h(X_t)$.

Define $\tilde{h}(x) = x - h(x)$. Thus, the drift condition is equivalent to

(D-lt') $\mathrm{E}[X_{t+1} \mid X_0, \dots, X_t] \leq \tilde{h}(X_t)$.

Then, for all $t \geq 0$,

$$\mathrm{E}[X_t \mid X_0] \leq \tilde{h}^t(X_0) + \frac{\tilde{h}(0)}{\tilde{h}'(0)}$$

and, in particular,

$$\mathrm{E}[X_t] \leq \tilde{h}^t(\mathrm{E}[X_0]) - \frac{\tilde{h}(0)}{\tilde{h}'(0)} \cdot \Pr[t \geq T \mid X_0].$$

For both these theorems, the drift function bounding the drift has to be convex and *greed-admitting*, which intuitively says that being closer to the goal is always better in terms of the expected state after an additional iteration, while search points closer to the goal are required to have weaker drift. These conditions are fulfilled in many sample applications.

In order to interpret the conclusions of the last two theorems properly, we need to estimate the term $\tilde{h}^t$. With the following theorem we give a general way of making this estimation.

Theorem 7.6: Estimation of Iterated Functions

Let $h\colon \mathbb{R} \to \mathbb{R}_+$ be a monotone non-decreasing and integrable function. Let $\tilde{h} = \mathrm{id} - h$. Then, for all starting points n and all target points $x \leq y$ and all time budgets t,

$$\text{if } t \geq \int_x^y \frac{1}{h(z)} \mathrm{d}z \quad \text{then} \quad \tilde{h}^t(y) \leq x.$$

We can specialize the previous theorem to the discrete case.

Theorem 7.7: Estimation of Iterated Functions, Sum Formula

Let $h\colon \mathbb{N} \to \mathbb{R}_+$ be a monotone non-decreasing function and let $\tilde{h} = \mathrm{id} - h$. Then, for all starting points $n \in \mathbb{N}$ and all target points $m \leq n$ and all time budgets t,

$$\text{if } t \geq \sum_{i=m}^{n-1} \frac{1}{h(i)} \quad \text{then} \quad \tilde{h}^t(n) \leq m.$$

Proof. We apply Theorem 7.6 [Estimation of Iterated Functions] to $\overline{h}\colon \mathbb{R} \to \mathbb{R}_{>0}, x \mapsto h(\max(0, \lfloor x \rfloor))$; note that we only care about non-negative arguments to h. Further, we use that, for all $i \in \mathbb{N}$,

$$\frac{1}{h(i)} = \int_i^{i+1} \frac{1}{\overline{h}(z)} \mathrm{d}z.$$

■

7.3 Applications to OneMax and LeadingOnes

In this section we show results from applications of Theorem 7.4 [Variable Fixed-Budget Drift, Unlimited Time] as given in [KW20]. We consider the optimization of the $(1+1)$ EA on OneMax and on LeadingOnes as examples. We start with OneMax, where we have multiplicative drift.

Theorem 7.8: Fixed Budget for OneMax

For all $t \in \mathbb{N}$, let X_t be the number of 1s which the $(1+1)$ EA on OneMax has found after t iterations of the algorithm. Then we have, for all t,

$$\mathrm{E}[X_t] \geq \begin{cases} \frac{n}{2} + \frac{t}{2\sqrt{e}} - O(1), & \text{if } t = O(\sqrt{n}); \\ \frac{n}{2} + \frac{t}{2\sqrt{e}}(1 - o(1)), & \text{if } t = o(n). \end{cases}$$

Furthermore, for all t, we have $\mathrm{E}[X_t] \geq n(1 - \exp(-t/(en))/2)$.

For the $(1+1)$ EA on OneMax, no concrete formula for a bound on the fitness value after t iterations was known: The original work [JZ12] could only handle RLS on OneMax, not the $(1+1)$ EA. The multiplicative drift theorem of [LS15] allows for deriving a lower bound of $n/2 + t/(2e)$ for $t = o(n)$, using a multiplicative drift constant of $(1 - 1/n)^n/n$. Since our drift theorem allows for variable drift, we can give the better bound of $n/2 + t/(2\sqrt{e}) - o(t)$ for the $(1+1)$ EA on OneMax with $t = o(n)$. Note that [LS15] also gives bounds for values of t closer to the expected optimization time.

Our second example shows the progress of the $(1+1)$ EA on LeadingOnes, where we have additive drift. The result is summarized in the following theorem.

Theorem 7.9: Fixed Budget for LeadingOnes

For all $t \in \mathbb{N}$, let X_t be the number of leading 1s which the $(1+1)$ EA on LeadingOnes has found after t iterations of the algorithm. We have, for all t,

$$\mathrm{E}[X_t] \geq \begin{cases} \frac{2t}{n} - O(1), & \text{if } t = O(n^{3/2}); \\ \frac{2t}{n} \cdot (1 - o(1)), & \text{if } t = o(n^2); \\ n \ln(1 + \frac{2t}{n^2}) - O(1), & \text{if } t \leq \frac{e-1}{2} n^2 - n^{3/2}. \end{cases}$$

For the $(1+1)$ EA on LeadingOnes with a budget of $t = o(n^2)$ iterations, the paper [JZ12] gives a lower bound of $2t/n - o(t/n)$ for the expected fitness after t iterations, which are recovered with a simpler proof. The general theorems from this section also allow budgets closer to the expected optimization time, where we get a lower bound of $n \ln(1 + 2t/n^2) - O(1)$.

7.4 Bibliographic Remarks

The setting of fixed-budget analysis was introduced to the analysis of randomized search heuristics by Jansen and Zarges [JZ12], who derived fixed-budget results for the classical example functions OneMax and LeadingOnes by bounding the expected progress in each iteration. A different perspective was proposed by Doerr, Jansen, Witt and Zarges [DJWZ13], who showed that fixed-budget statements can be derived from bounds on optimization times if these exhibit strong concentration. Lengler and Spooner [LS15] proposed a variant of multiplicative drift for fixed-budget results and the use of differential equations in the context of OneMax and general linear functions. Nallaperuma, Neumann and Sudholt [NNS17] applied fixed-budget theory to the analysis of evolutionary algorithms on the traveling salesman problem and Jansen and Zarges [JZ14] to artificial immune systems. The quality gains of optimal black-box algorithms on OneMax in a fixed-budget perspective were analyzed by Doerr, Doerr and Yang [DDY20]. He, Jansen and Zarges [HJZ19] consider the so-called unlimited budgets to estimate fitness values in particular for points of time larger than the expected optimization time. A survey by Jansen [Jan20] summarizes the state of the art in the area of fixed-budget analysis.

In contrast to the numerous drift theorems available for bounding the optimization time, there was no corresponding theorem for making a fixed-budget analysis apart from one for the multiplicative case given in [LS15]. This changed with [KW20], introduced in this chapter, providing several such drift theorems.

Note that a further fixed-budget drift theorem can be found in [KW20], where a detour of the computation of fixed budget results via first hitting times is made.

8 Drift as an Average: A Closer Look on the Conditioning of Drift

Consider a deterministic process which starts at 100 and goes down by 1 in each iteration; we trivially see that the expected time until the process reaches 0 is 100. If the process goes down by 1 or $1/2$, then a worst-case view would state an upper bound of 200 until the process reaches 0.

Drift theory allows for an *average case* view. In fact, the main strength of drift theory is that even the possibility of going *away* from the target is incorporated, as long as *in expectation* we have a bias towards the target. For example, if a process goes down by 10 with probability $11/20$ and up by 10 with probability $9/20$, then the progress is *in expectation* also 1, but not in the worst case. The additive drift theorem tells us that, also in this case, we expect to arrive at 0 after 100 steps. Thus, instead of a worst case bound, averaging different outcomes leads to a useful bound.

In this section we investigate three questions.

(1) How do drift theorems allow to exploit that drift is an average over a range of possibilities?

(2) Why do drift theorems in the literature condition on various different things?

(3) How do we account for insufficient drift after reaching the target?

8.1 Drift as an Average

In Theorem 2.1 [Additive Drift, Upper Bound] we have seen the standard (additive) drift condition to be

(D) there is a $\delta > 0$ such that, for all $t < T$, it holds that $E[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq \delta$.

In this section we want to take a closer look at the conditioning on $X_0, \ldots, X_t$. First we show that *not* conditioning on anything leads to counterexamples.

> **Example 8.1: Global Averaging** — Suppose we first flip a coin in secret. Then, if the coin shows tails, for all $i \in \mathbb{N}$ we let $X_i = 1$. If, on the other hand, the coin shows heads, we let $X_0 = 1$ and, for all $i \in \mathbb{N}$, we draw independently uniformly at random bits $B_i \in \{0, 1\}$ and set $X_{i+1} = X_i - B_i$. For any t with $t < T$ we now have $E[X_t - X_{t+1}] \geq 1/4$ (we make a progress of exactly 1 if the coin shows tails and the bit is 1, and otherwise of 0). Thus, the conclusion of the additive drift theorem states an upper bound of 4 on the expected time to hit 0, while, in fact, $E[T]$ is infinite.

The previous example shows that the conclusion of the additive drift theorem can be false while the drift condition holds, averaged over *all* possible situations. What we *can* do is average over *all possible situations with the same history of* $X_0, \ldots, X_t$, as stated by the additive drift theorem. To illustrate this, we have the following example.

> **Example 8.2: Local Averaging** — Let us play a game where your goal is to draw a total number of 10 red balls. In each iteration I randomly fill in secret a bag of balls of different colors, and draw a ball uniformly at random from that bag. Suppose I either fill the bag with 10 balls, 9 of which are blue and one is red, or with 100 balls, where 99 are blue and one is red. Suppose I choose either situation with equal probability of $1/2$. Then, on average, in any iteration your probability to pick a red ball is $11/200$. Thus, you arrive at a value of 10 drawn red balls after an expected number of $2000/11$ iterations.

Note that in this example there seem to be two different possible situations in each iteration with different drift. One way to address this is to bound drift by the smaller of the two drift values; but the drift theorem allows for averaging the drift of the different situations, since we are given the probabilities of the two values.

We cannot average over global decisions that we can learn about from the history, see Example 8.1 [Global Averaging]; this depends on our choice of what is the history in this context. If we cannot learn about the global decisions from the history, we can use the *principle of deferred decision* to model the random decision as a decision in that given iteration, as illustrated by the following example.

Example 8.3: Deferred Decisions — Let us again play a game where your goal is to draw a total number of 10 red balls. This time, before the game starts, I fill an infinite sequence of bags, making for each the exact same decision as given in Example 8.2 [Local Averaging]. In each iteration you get the next bag from this sequence. Since the outcome of one bag is independent of other bags, we cannot learn anything about future bags from the history. Thus, using the principle of deferred decision, we compute as if the bag was only packed in the current iteration, after all previous (random) decisions have been made. Thus, the analysis proceeds exactly as in Example 8.2 [Local Averaging] and you arrive at a value of 10 drawn red balls after an expected number of $2000/11$ iterations.

8.2 The Conditioning of Drift

Let us take a closer look at the drift condition. For this section, we define four variants with different conditioning of the drift as follows.

Definition 8.4: Variants on the Conditioning of Drift

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable random process over $\mathbb{R}$ and let $(F_t)_{t\in\mathbb{N}}$ be a filtration such that $(X_t)_{t\in\mathbb{N}}$ is adapted to $(F_t)_{t\in\mathbb{N}}$, let f be a measurable function and $t \in \mathbb{N}$.

(D-filtration) $E[X_t - X_{t+1} \mid F_t] \geq f(X_t)$ with probability 1.

(D-history) $E[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq f(X_t)$ with probability 1.

(D-events) For all $s_0, \ldots, s_t$ with $\Pr[X_0 = s_0, \ldots, X_t = s_t] > 0$,

$$E[X_t - X_{t+1} \mid X_0 = s_0, \ldots, X_t = s_t] \geq f(s_t).$$

(D-Markov) For all s_t with $\Pr[X_t = s_t] > 0$, $E[X_t - X_{t+1} \mid X_t = s_t] \geq f(s_t)$.

Some researchers prefer to state all drift theorems in terms of filtrations, see, for example, [Wit23]; some prefer conditioning on the history [KK19].

In Section 2.1.2 of [Len20], Lengler discusses the differences between (D-filtration) and (D-Markov), phrasing drift theorems in terms of (D-Markov). Many applications of drift theorems involve states of algorithms which typically behave as Markov chains. This ubiquity of Markov chains sometimes leads to drift theorems being stated for Markov chains only. However, the states of algorithms need to be mapped to real numbers in order to apply drift theorems (see Section 3 [The Art of Potential Functions] for a discussion on potential functions). If this mapping is not 1-to-1, then, in general, the resulting mapped process is not Markovian any more, so drift theorems applicable for Markov chains on $\mathbb{R}$ are no longer applicable. As we will see, (D-history) is a necessary condition for (D-Markov) (see Theorem 8.6 [Conditioning on Filtration vs. History vs. Events]), which is why in this work all drift theorems are stated analogously to (D-history).

In the following we want to discuss how the four given conditions differ and in what sense they are equivalent. First we recall that conditioning on the history $X_0, \ldots, X_t$ is defined as conditioning on the σ-algebra $\sigma(X_0, \ldots, X_t)$, leading to the canonical filtration. In this sense, the condition (D-history) implies that there is a filtration such that (D-filtration) holds.

Propostion 8.5: Canonical Filtration as Filtration

Let $(X_t)_{t \in \mathbb{N}}$ be an integrable random process $\mathbb{R}$, f a measurable function and $t \in \mathbb{N}$. Suppose (D-history). Then there is a filtration $(F_t)_{t \in \mathbb{N}}$ such that $(X_t)_{t \in \mathbb{N}}$ is adapted to $(F_t)_{t \in \mathbb{N}}$ and such that (D-filtration) holds.

Proof. Using $F_t = \sigma(X_0, \ldots, X_t)$, (D-history) and (D-filtration) are identical. ■

The question now arises whether anything can be gained from using other filtrations than the canonical filtration. Sometimes it can be easier to assume a different filtration, which gives more information for the analysis to work with; more outcomes of random variables can be fixed, allowing the analysis to proceed with these concrete outcomes (see, for example, the proof of Theorem 4.1 [Unbiased Random Walk on the Line]). But how should the drift theorem be stated? Using the following theorem, we see that if the drift theorem is only stated conditional on the history, any other filtration (where the process is adapted to) can also be used, since (D-filtration) implies (D-history).

Theorem 8.6: Conditioning on Filtration vs. History vs. Events

Let $(X_t)_{t \in \mathbb{N}}$ be an integrable random processover $\mathbb{R}$ and f a measurable function. Suppose further $(F_t)_{t \in \mathbb{N}}$ is a filtration such that $(X_t)_{t \in \mathbb{N}}$ is adapted to $(F_t)_{t \in \mathbb{N}}$. We then have, for all $t \in \mathbb{N}$, the following implications:
(D-filtration) $\Rightarrow$ (D-history) $\Rightarrow$ (D-events) $\Rightarrow$ (D-Markov)
where the last implication holds for discrete $(X_t)_{t \in \mathbb{N}}$.

Proof. Suppose first, for "(D-filtration) $\Rightarrow$ (D-history)", $\mathrm{E}[X_{t+1} \mid F_t] \geq f(X_t)$ with probability 1.

Since $(X_t)_{t \in \mathbb{N}}$ is adapted to $(F_t)_{t \in \mathbb{N}}$, we have that, for all t,

$$\sigma(X_0, \ldots, X_t) \subseteq F_t.$$

Using Lemma 9.15 [Tower Property for Sub-σ-Algebra] in the second equality, we have, with probability 1,

$$\begin{aligned} \mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] &= \mathrm{E}[X_t - X_{t+1} \mid \sigma(X_0, \ldots, X_t)] \\ &= \mathrm{E}[\mathrm{E}[X_t - X_{t+1} \mid F_t] \mid \sigma(X_0, \ldots, X_t)] \\ &\geq \mathrm{E}[f(X_t) \mid \sigma(X_0, \ldots, X_t)] \\ &= f(X_t). \end{aligned}$$

Suppose now, for "(D-history) $\Rightarrow$ (D-events)", $\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq f(X_t)$ with probability 1. Let $s_0, \ldots, s_t$ and let A be the event such that $X_0 = s_0, \ldots, X_t = s_t$. Suppose $\Pr[A] > 0$. Since $A \in \sigma(X_0, \ldots, X_t)$, we get (using Lemma 9.13 [Conditional and Indicator] in the first step and Definition 9.14 [Filtration Conditional] in the second),

$$\begin{aligned} \mathrm{E}[X_t - X_{t+1} \mid A] \Pr[A] &= \mathrm{E}[(X_t - X_{t+1}) \mathbf{1}\{A\}] \\ &= \mathrm{E}[\mathrm{E}[X_t - X_{t+1} \mid X_0, \ldots, X_t] \mathbf{1}\{A\}] \\ &\geq \mathrm{E}[f(X_t) \mathbf{1}\{A\}] \\ &= f(s_t) \Pr[A]. \end{aligned}$$

Regarding (D-events) $\Rightarrow$ (D-Markov), we note that, for discrete $(X_t)_{t \in \mathbb{N}}$, and any s_t with $\Pr[X_t = s_t] > 0$, we can find $s_0, \dots, s_{t-1}$ such that $\Pr[X_0 = s_0, \dots, X_t = s_t] > 0$. For such $s_0, \dots, s_t$ we then have

$$\mathrm{E}[X_t - X_{t+1} \mid X_t = s_t] = \mathrm{E}[X_t - X_{t+1} \mid X_0 = s_0, \dots, X_t = s_t].$$

This gives the desired implication. ■

Note that (D-events) and (D-Markov) implicitly consider the process to be discrete. As we see in the example given next, a drift theorem based on (D-events) without the the requirement of a discrete process would in this generality be wrong.

Example 8.7: Conditioning on Events of Continuous Processes — Let X_0 be a uniformly real random number from $[1, 2]$ and let, for all $t \in \mathbb{N}$, $X_{t+1} = X_t$. Then we have

(D-events) for all $s_0, \dots, s_t$ with $\Pr[X_0 = s_0, \dots, X_t = s_t] > 0$, $\mathrm{E}[X_t - X_{t+1} \mid X_0 = s_0, \dots, X_t = s_t] \geq 1$.

This follows since, for all $t \in \mathbb{N}$ and all $s_0, \dots, s_t$, we have $\Pr[X_0 = s_0, \dots, X_t = s_t] = 0$ (we have a truly continuous random variable). Thus, (D-events) is vacuously true. Furthermore, for all $t \in \mathbb{N}$, $X_t \geq 1$, so there is no t such that $X_t \leq 0$.

The example shows that the problem arises when considering continuous random variables. The next proposition shows that, for discrete processes, (D-events) implies (D-history), making these two conditions equivalent. The proof is due to Marcus Pappik (private communication).

Propostion 8.8: Equivalence of Conditionals for Discrete Search Spaces

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable and *discrete* random process over $\mathbb{R}$, f a measurable function and $t \in \mathbb{N}$. Then (D-events) implies (D-history).

Proof. Fix $t \geq 0$, and let $R := \mathrm{range}(X_0) \times \dots \times \mathrm{range}(X_t)$. For a tuple $s = (s_i)_{0\leq i\leq t} \in R$, let $A(s) := \{\forall 0 \leq i \leq t : X_i = s_i\}$. Let $S := \{s \in R \mid \Pr[A(s)] > 0\}$. For all $s = (s_i)_{0\leq i\leq t} \in S$ and $\omega \in A(s)$, Lemma 9.17 [Conditioning on History vs. Events] and (D-event) yield

$$\mathrm{E}[X_t - X_{t+1} \mid X_0, \dots, X_t](\omega) = \mathrm{E}[X_t - X_{t+1} \mid A(s)] \geq f(s_t) = f(X_t(\omega)).$$

Since further every X_i has countable range, it holds in particular that S and R are countable. Hence, we have that $\bigcup_{s\in S} A(s)$ is measurable and

$$\Pr\left[\bigcup_{s\in S} A(s)\right] \geq 1 - \sum_{s\in R\setminus S} \Pr[A(s)] = 1.$$

Therefore, we have that $\mathrm{E}[X_t - X_{t+1} \mid X_0, \dots, X_t] \geq f(X_t)$ holds with probability 1. ■

Just as (D-events) implicitly assumes a discrete space, (D-Markov) assumes the process to be Markovian. The following theorem shows that, for discrete Markov chains, (D-Markov) implies (D-events), making also these two conditions equivalent in this case.

Propostion 8.9: Equivalence for Markov Chains

Let $(X_t)_{t\in\mathbb{N}}$ be an integrable and *discrete* Markov chain over $\mathbb{R}$, f a measurable function and $t \in \mathbb{N}$. Then (D-Markov) implies (D-events).

Proof. This follows directly from the Markov property that, for all $s_0, \ldots, s_t$ with $\Pr[X_0 = s_0, \ldots, X_t = s_t] > 0$, $E[X_{t+1} \mid X_t = s_t] = E[X_{t+1} \mid X_0 = s_0, \ldots, X_t = s_t]$. ■

We have seen that for discrete spaces and for Markov chains, one can give specialized formulations of drift theorems. However, drift theorems conditioning on the history are strictly more general. Furthermore, conditioning on the history is easy to state and understand for users of the theorem.

Conditioning on a filtration results in drift theorems equally general as those conditioning on the history, since the history is one possible filtration, and in fact the least restrictive.

8.3 Reaching the Target

Let us consider again the drift condition

(D) there is a $\delta > 0$ such that, for all $t < T$, it holds that $E[X_t - X_{t+1} \mid X_0, \ldots, X_t] \geq \delta$.

We want to take a closer look at the requirement "for all $t < T$". Since T is a random variable, this does not properly define a range for t. This is desirable to allow for processes which naturally do not go down anymore after reaching the target. One clean way to write it would be

(D) there is a $\delta > 0$ such that, for all $t \in \mathbb{N}$, $E[X_t - X_{t+1} \mid X_0, \ldots, X_t]\mathbf{1}\{t < T\} \geq \delta\mathbf{1}\{t < T\}$.

This notation resorts to indicator random variables and, while quantifying t over all of $\mathbb{N}$, effectively requires the inequality to hold only in case of $t < T$. This inspires the following convention.

Convention 8.10: Drift While not at Target — We state inequalities that only need to hold for points in time when a random process did not reach its target yet. Formally, let T be a random variable over $\mathbb{N} \cup \{\infty\}$, let X and Y be random variables over $\mathbb{R}$.
Further, let $\sim$ denote a relation symbol, such as $=$, $\leq$, or $\geq$. We define the phrase "for all $t < T$, it holds that $X \sim Y$" to be equivalent to "for all $t \in \mathbb{N}$, it holds that $X \cdot \mathbf{1}\{t < T\} \sim Y \cdot \mathbf{1}\{t < T\}$".

Note that, alternatively, we can *condition* on $t \leq T$, the way chosen in Theorem 5.1 [Additive Drift, Upper Bound, Time Condition] and Theorem 5.2 [Additive Drift, Lower Bound, Time Condition]. This makes the dependence on $t < T$ very explicit (this was the reason for stating it in this way in the two named theorems). However, now one has to quantify over "all $t \in \mathbb{N}$ such that $\Pr[t < T] > 0$", which is again somewhat cumbersome.

9 Notation

In this section we collect some algorithms, notation and lemmas used in this work.

9.1 Algorithms

The most simple search heuristic is *Random Local Search* (RLS). It starts with a random bit string and iteratively tries to improve it by changing the currently best search point in exactly one position (we use this as the "flipOne" function below). The pseudo code for RLS maximizing a given function $f\colon \{0,1\}^n \to \mathbb{R}$ is given as follows.

Algorithm 2: Random Local Search (RLS)

```
Sample x ∈ {0,1}^n uniformly at random
for i = 1 to ∞ do
    y ← flipOne(x)
    if f(y) ≥ f(x) then x ← y
```

The algorithm is set up to maximize the given function f; by turning the inequality around, we get the analogous algorithm for minimization.

RLS constitutes a simple and straightforward hill climber. A slightly more advanced algorithm allows for larger *jumps*, that is, it also considers changing the currently best search point in more than one position. The most common way to achieve this is by flipping not exactly one bit, but each bit independently with some predefined probability p. This independently random flipping of bits is called *mutation*, and the resulting algorithm is called the $(1+1)$ EA; its pseudo code is given as follows.

Algorithm 3: The $(1+1)$ EA

```
Sample x ∈ {0,1}^n uniformly at random
for i = 1 to ∞ do
    y ← mutate(x)
    if f(y) ≥ f(x) then x ← y
```

Note that the standard bit flip probability is $p = 1/n$, implying that, on average, exactly one bit changes.

We consider in this document two concrete test functions and one function class as follows.

Definition 9.1: Test Functions

- OneMax is a function $\{0,1\}^n \to \mathbb{R}$ mapping any bit string to the number of 1s in the bit string.
- LeadingOnes is a function $\{0,1\}^n \to \mathbb{R}$ mapping any bit string to the number of 1s *before the first* 0 (if any) in the bit string (the number of leading 1s).
- A *linear function* is any function $f\colon \{0,1\}^n \to \mathbb{R}$ such that there exists $w_1, \ldots, w_n \in \mathbb{R}$ such that, for all $x \in \{0,1\}^n$, $f(x) = \sum_{i=1}^n w_i x_i$.

9.2 Notation

Next we give some non-standard notation.

Definition 9.2: Discrete Intervals

For any $n, m \in \mathbb{N}$ with $n \leq m$, we use $[n..m]$ to denote the set $\{i \in \mathbb{N} \mid n \leq i \leq m\}$. Furthermore, for any $n \in \mathbb{N}_+$, we will write $[n]$ for $[1..n]$.

Definition 9.3: Function Self-Composition

For any function $f: X \to X$ and $i \geq 0$, we let f^i denote the i-times self-composition of f (with f^0 being the identity on X).

We use the following notation regarding probabilities.

Definition 9.4: Indicator Function

For any event A, let $\mathbf{1}\{A\}$ denote the indicator function for the event A.

Definition 9.5: Conditional Expectation

For any discrete random variable X and any event A such that $\Pr[A] > 0$, we have

$$\mathrm{E}[X \mid A] = \sum_x x \frac{\Pr[\{X = x\} \cap A]}{\Pr[A]}.$$

Note that we can only condition on the event $X = s$ if $\Pr[X = s] > 0$.

Definition 9.6: Integrability

A random variable X is *integrable* if and only if $\mathrm{E}[|X|] < \infty$. In general, a random process $(X_t)_{t \in \mathbb{N}}$ is *integrable* if and only if, for all $t \in \mathbb{N}$, it holds that X_t is integrable.

Definition 9.7: Discrete Random Variable, Discrete Random Process

A random variable X is *discrete* if and only if it has a countable range. We call a random process $(X_t)_{t \in \mathbb{N}}$ discrete if each X_t is discrete.

Definition 9.8: Monotone Process

A random process $(X_t)_{t \in \mathbb{N}}$ is *monotone* (sometimes called *monotone non-decreasing*) if and only if, for all t, $X_t \leq X_{t+1}$ (point-wise, i.e., for all atomic events $\omega \in \Omega$ we have $X_t(\omega) \leq X_{t+1}(\omega)$).

Definition 9.9: Stochastic Dominance

Given two random variables X, Y over $\mathbb{R}$, we say that Y *stochastically dominates* X if and only if, for all $x \in \mathbb{R}$, $\Pr[X \leq x] \geq \Pr[Y \leq x]$; we write $X \preceq Y$.

Definition 9.10: Markov Chain

A discrete random process $(X_t)_{t \in \mathbb{N}}$ is a *Markov chain* if and only if the outcome of each next step only depends on the current state and time point. Formally, for all $t \in \mathbb{N}$ as well as all $s \in \mathbb{R}$ and all $v \in \mathbb{R}^t$, it holds that $\Pr[X_{t+1} = s \mid X_t = v_t] = \Pr[X_{t+1} = s \mid \forall t' \in [0..t] \colon X_{t'} = v_{t'}]$. A Markov chain $(X_t)_{t\in\mathbb{N}}$ is *time-homogeneous* if and only if the outcome of each next step only depends on the current state but not the current time point. Formally, for all $t, k \in \mathbb{N}$ as well as all $s, u \in \mathbb{R}$, it holds that $\Pr[X_{t+1} = s \mid X_t = u] = \Pr[X_{t+k+1} = s \mid X_{t+k} = u]$.

9.3 Lemmas

We will make use of the following lemmas. The first two pertain to bounding certain sums.

Lemma 9.11: Upper Bound on the Harmonic Sum

For all $n \in \mathbb{N}_+$ we have
$$\sum_{i=1}^{n} \frac{1}{i} \leq \ln(n) + 1.$$

Proof. See (1.4.12) in [Doe20]. ■

Lemma 9.12: Upper Bounding Sum by Integral

Let $a, b \in \mathbb{R}$ with $a \leq b$ and let $f\colon [a, b] \to \mathbb{R}$ be integrable. If f is monotone increasing, then
$$\sum_{i=a}^{b-1} f(i) \leq \int_a^b f(x)\,\mathrm{d}x.$$
If f is monotone decreasing, then
$$\sum_{i=a+1}^{b} f(i) \leq \int_a^b f(x)\,\mathrm{d}x.$$

The remaining lemmas pertain to random variables.

Lemma 9.13: Conditional and Indicator

For all events A with $\Pr[A] > 0$, we have
$$\mathrm{E}[Y \mid A] \cdot \Pr[A] = \mathrm{E}[Y\mathbf{1}\{A\}].$$

Definition 9.14: Filtration Conditional

Let a sub-σ-algebra F of the underlying probability space be given and let X be a random variable. Then $\mathrm{E}[Y \mid F]$ refers to any F-measurable random variable such that, for all $A \in F$ with probability 1,
$$\mathrm{E}[Y \cdot \mathbf{1}\{A\}] = \mathrm{E}[\mathrm{E}[Y \mid F] \cdot \mathbf{1}\{A\}].$$

Lemma 9.15: Tower Property for Sub-σ-Algebra

Let two sub-σ-algebras $F_1 \subseteq F_2$ of the underlying probability space be given and let X be a random variable. Then, with probability 1,

$$\mathrm{E}[\mathrm{E}[X \mid F_2] \mid F_1] = \mathrm{E}[X \mid F_1].$$

Very much related to the preceding lemma is the law of total expectation (also known under other names and with different formulations).

Lemma 9.16: Law of Total Expectation

Let X be a random variable and $A_i, \dots, A_n$ disjoint measurable events with positive probability that partition the probability space. Then we have

$$\mathrm{E}[X] = \sum_{i=1}^{n} \mathrm{E}[X \mid A_i] \Pr[A_i].$$

We will also have cause to use a different way to state this. Let X, Y be two random variables. Then

$$\mathrm{E}[X] = \mathrm{E}[\mathrm{E}[X \mid Y]].$$

We need the following lemma to make a specific proof in this document rigorous. The proof is due to Marcus Pappik (private communication).

Lemma 9.17: Conditioning on History vs. Events

Let $Y, X_0, \dots, X_t$ be random variables and let Z be any version of the conditional expectation $\mathrm{E}[Y \mid X_0, \dots, X_t]$. For all $s_0, \dots, s_t \in \mathbb{R}$ such that $\Pr[X_0 = s_0, \dots, X_t = s_t] > 0$ and all $\omega \in \{X_0 = s_0, \dots, X_t = s_t\}$ it holds that

$$Z(\omega) = \mathrm{E}[Y \mid X_0 = s_0, \dots, X_t = s_t].$$

Proof. We start by proving the following claim.

Claim. Consider two measurable spaces $(\Omega, \mathcal{F})$ and $(S, \mathcal{S})$ such that, for all $s \in S$, $\{s\} \in \mathcal{S}$. Let $X, Z\colon \Omega \to S$ be measurable. If Z is $\sigma(X)$-measurable then Z is constant on $\{X = s\}$ for every $s \in S$.

Proof of claim. For the sake of contradiction, assume the statement is false. That is, suppose Z is $\sigma(X)$-measurable and there is some $s_0 \in S$ such that Z is not constant on $A_0 := \{X = s_0\}$. Let $s_1, s_2 \in S$ be distinct values of Z on A_0. Since Z is $\sigma(X)$-measurable, it holds that $A_1 := \{Z = s_1\} \in \sigma(X)$. Consequently, we have that $A_0 \cap A_1 \in \sigma(X)$ and, by construction, $\emptyset \subset A_0 \cap A_1 \subset A_0$. We now show that these two properties of $A_0 \cap A_1$ pose a contradiction. To this end, suppose there exists any set $A \subseteq \Omega$ with $\emptyset \subset A \subset A_0$ and $A \in \sigma(X)$. Note that

$$\sigma(X) = \{X^{-1}(B) \mid B \in \mathcal{S}\}.$$

Hence, if $A \in \sigma(X)$ then $B := \{X(\omega) \mid \omega \in A\}$ must be in $\mathcal{S}$. But since $\emptyset \subset A \subset A_0$ it holds that

$$\emptyset \subset B \subset \{X(a) \mid a \in A_0\} = \{s_0\}.$$

However, since both inclusions are strict, such a set B cannot exist. **End of proof of claim.**

To prove the lemma, suppose now that Z is any version of $\mathrm{E}[Y \mid X_0, \dots, X_t]$ and let $A = \{X_0 = s_0, \dots, X_t = s_t\}$ for any $s_0, \dots, s_t \in \mathbb{R}$. Since Z is $\sigma(X_0, \dots, X_t)$-measurable, the claim above yields that Z is constant on A. Let z be the value of Z on A and note that, for all $\omega \in A$, we have

$$Z(\omega) \cdot \Pr[A] = \mathrm{E}[z \cdot \mathbf{1}\{A\}] = \mathrm{E}[Z \cdot \mathbf{1}\{A\}] = \mathrm{E}[Y \cdot \mathbf{1}\{A\}],$$

where the first equality is due to $Z(\omega) = z$ and $\Pr[A] = \mathrm{E}[\mathbf{1}\{A\}]$, the second equality follows from the fact that $z \cdot \mathbf{1}\{A\} = Z \cdot \mathbf{1}\{A\}$ point-wise, and the last equality is due to the fact that Z is a version of $\mathrm{E}[Y \mid X_0, \ldots, X_t]$. Further, by Lemma 9.13 [Conditional and Indicator], we have, using $\Pr[A] > 0$,

$$\mathrm{E}[Y \cdot \mathbf{1}\{A\}] = \mathrm{E}[Y \mid A] \cdot \Pr[A].$$

Combining both steps yields

$$Z(\omega) \cdot \Pr[A] = \mathrm{E}[Y \mid A] \cdot \Pr[A].$$

Provided $\Pr[A] > 0$, this implies $Z(\omega) = \mathrm{E}[Y \mid A]$ for all $\omega \in A$, which proves the lemma. ■

10 Acknowledgments

While many people were important for this habilitation document, three people come to mind in particular. First of all Tobias Friedrich, who had my back for many years, providing an outstanding work environment for me to thrive. Most inspirational for my development as a researcher was Benjamin Doerr, who acted as a mentor while pretending to be merely a collaborator; I learned a lot from Benjamin and enjoyed our many projects together. Furthermore, Martin Krejca was my PhD student and we had many fruitful collaborations; now he is a good friend, an impeccable discussion partner and generally a person I can rely on in many situations.

The community of researchers working on the Theory of Randomized Search Heuristics gave me a home as a scientist. The reviewing is thorough, the discussions interesting and the atmosphere very friendly. I want to point out particularly Carsten Witt, Carola Doerr, Johannes Lengler and Dirk Sudholt (all are also collaborators), but many other people are also significant for the community.

Finally, many researchers from the research group I am in contributed to this thesis either with discussions or by reading and commenting on drafts. In no particular order, “thank you” to Anton Krohmer, Marcus Pappik, Nicolas Klodt, Aishwarya Radhakrishnan, Xiaoyue Sherry Li, Nadym Mallek, Stefan Neubert, Janosch Ruff, Ziena Zeif and Sam Baguley.

References

[BDN10] Süntje Böttcher, Benjamin Doerr, and Frank Neumann. Optimal fixed and adaptive mutation rates for the leadingones problem. In Robert Schaefer, Carlos Cotta, Joanna Kolodziej, and Günter Rudolph, editors, *Parallel Problem Solving from Nature - PPSN XI, 11th International Conference, Kraków, Poland, September 11-15, 2010, Proceedings, Part I*, volume 6238 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2010.

[BLM+20] Daniel Bertschinger, Johannes Lengler, Anders Martinsson, Robert Meier, Angelika Steger, Milos Trujic, and Emo Welzl. An optimal decentralized $(\Delta + 1)$-coloring algorithm. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 17:1–17:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[CDEL18] Dogan Corus, Duc-Cuong Dang, Anton V. Eremeev, and Per Kristian Lehre. Level-based analysis of genetic algorithms and other search processes. *IEEE Trans. Evol. Comput.*, 22(5):707–719, 2018.

[DDK15] Benjamin Doerr, Carola Doerr, and Timo Kötzing. Solving problems with unknown solution length at (almost) no extra cost. In Sara Silva and Anna Isabel Esparcia-Alcázar, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11-15, 2015*, pages 831–838. ACM, 2015.

[DDK16] Benjamin Doerr, Carola Doerr, and Timo Kötzing. The right mutation strength for multi-valued decision variables. In Tobias Friedrich, Frank Neumann, and Andrew M. Sutton, editors, *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference, Denver, CO, USA, July 20 - 24, 2016*, pages 1115–1122. ACM, 2016.

[DDK18] Benjamin Doerr, Carola Doerr, and Timo Kötzing. Static and self-adjusting mutation strengths for multi-valued decision variables. *Algorithmica*, 80(5):1732–1768, 2018.

[DDY20] Benjamin Doerr, Carola Doerr, and Jing Yang. Optimal parameter choices via precise black-box analysis. *Theor. Comput. Sci.*, 801:1–34, 2020.

[DFF+19] Benjamin Doerr, Philipp Fischbeck, Clemens Frahnow, Tobias Friedrich, Timo Kötzing, and Martin Schirneck. Island models meet rumor spreading. *Algorithmica*, 81(2):886–915, 2019.

[DG10] Benjamin Doerr and Leslie Ann Goldberg. Drift analysis with tail bounds. In Robert Schaefer, Carlos Cotta, Joanna Kolodziej, and Günter Rudolph, editors, *Parallel Problem Solving from Nature - PPSN XI, 11th International Conference, Kraków, Poland, September 11-15, 2010, Proceedings, Part I*, volume 6238 of *Lecture Notes in Computer Science*, pages 174–183. Springer, 2010.

[DJW00] Stefan Droste, Thomas Jansen, and Ingo Wegener. Dynamic parameter control in simple evolutionary algorithms. In Worthy N. Martin and William M. Spears, editors, *Proceedings of the Sixth Workshop on Foundations of Genetic Algorithms, Charlottesville, VA, USA, July 21-23, 2000*, pages 275–294. Morgan Kaufmann, 2000.

[DJW02] Stefan Droste, Thomas Jansen, and Ingo Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theor. Comput. Sci.*, 276(1-2):51–81, 2002.

[DJW10] Benjamin Doerr, Daniel Johannsen, and Carola Winzen. Multiplicative drift analysis. In Martin Pelikan and Jürgen Branke, editors, *Genetic and Evolutionary Computation Conference, GECCO 2010, Proceedings, Portland, Oregon, USA, July 7-11, 2010*, pages 1449–1456. ACM, 2010.

[DJW12] Benjamin Doerr, Daniel Johannsen, and Carola Winzen. Multiplicative drift analysis. *Algorithmica*, 64(4):673–697, 2012.

[DJWZ13] Benjamin Doerr, Thomas Jansen, Carsten Witt, and Christine Zarges. A method to derive fixed budget results from expected optimisation times. In Christian Blum and Enrique Alba, editors, *Genetic and Evolutionary Computation Conference, GECCO '13, Amsterdam, The Netherlands, July 6-10, 2013*, pages 1581–1588. ACM, 2013.

[DK14] Benjamin Doerr and Marvin Künnemann. Tight analysis of randomized rumor spreading in complete graphs. In Michael Drmota and Mark Daniel Ward, editors, *2014 Proceedings of the Eleventh Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2014, Portland, Oregon, USA, January 6, 2014*, pages 82–91. SIAM, 2014.

[DK21a] Benjamin Doerr and Timo Kötzing. Lower bounds from fitness levels made easy. In Francisco Chicano and Krzysztof Krawiec, editors, *GECCO '21: Genetic and Evolutionary Computation Conference, Lille, France, July 10-14, 2021*, pages 1142–1150. ACM, 2021.

[DK21b] Benjamin Doerr and Timo Kötzing. Multiplicative up-drift. *Algorithmica*, 83(10):3017–3058, 2021.

[DKLL17] Benjamin Doerr, Timo Kötzing, J. A. Gregor Lagodzinski, and Johannes Lengler. Bounding bloat in genetic programming. In Peter A. N. Bosman, editor, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017, Berlin, Germany, July 15-19, 2017*, pages 921–928. ACM, 2017.

[DKLL20] Benjamin Doerr, Timo Kötzing, J. A. Gregor Lagodzinski, and Johannes Lengler. The impact of lexicographic parsimony pressure for ORDER/MAJORITY on the run time. *Theor. Comput. Sci.*, 816:144–168, 2020.

[DL16] Duc-Cuong Dang and Per Kristian Lehre. Runtime analysis of non-elitist populations: From classical optimisation to partial information. *Algorithmica*, 75(3):428–461, 2016.

[Doe20] Benjamin Doerr. Probabilistic tools for the analysis of randomized optimization heuristics. In Benjamin Doerr and Frank Neumann, editors, *Theory of Evolutionary Computation - Recent Developments in Discrete Optimization*, Natural Computing Series, pages 1–87. Springer, 2020.

[Dro04] Stefan Droste. Analysis of the (1+1) EA for a noisy onemax. In Kalyanmoy Deb, Riccardo Poli, Wolfgang Banzhaf, Hans-Georg Beyer, Edmund K. Burke, Paul J. Darwen, Dipankar Dasgupta, Dario Floreano, James A. Foster, Mark Harman, Owen Holland, Pier Luca Lanzi, Lee Spector, Andrea Tettamanzi, Dirk Thierens, and Andrew M. Tyrrell, editors, *Genetic and Evolutionary Computation - GECCO 2004, Genetic and Evolutionary Computation Conference, Seattle, WA, USA, June 26-30, 2004, Proceedings, Part I*, volume 3102 of *Lecture Notes in Computer Science*, pages 1088–1099. Springer, 2004.

[FK13] Matthias Feldmann and Timo Kötzing. Optimizing expected path lengths with ant colony optimization using fitness proportional update. In Frank Neumann and Kenneth A. De Jong, editors, *Foundations of Genetic Algorithms XII, FOGA '13, Adelaide, SA, Australia, January 16-20, 2013*, pages 65–74. ACM, 2013.

[FKK16] Tobias Friedrich, Timo Kötzing, and Martin S. Krejca. Edas cannot be balanced and stable. In Tobias Friedrich, Frank Neumann, and Andrew M. Sutton, editors, *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference, Denver, CO, USA, July 20 - 24, 2016*, pages 1139–1146. ACM, 2016.

[FKKS15a] Tobias Friedrich, Timo Kötzing, Martin S. Krejca, and Andrew M. Sutton. The benefit of recombination in noisy evolutionary search. In Khaled M. Elbassioni and Kazuhisa Makino, editors, *Algorithms and Computation - 26th International Symposium, ISAAC 2015, Nagoya, Japan, December 9-11, 2015, Proceedings*, volume 9472 of *Lecture Notes in Computer Science*, pages 140–150. Springer, 2015.

[FKKS15b] Tobias Friedrich, Timo Kötzing, Martin S. Krejca, and Andrew M. Sutton. Robustness of ant colony optimization to noise. In Sara Silva and Anna Isabel Esparcia-Alcázar, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11-15, 2015*, pages 17–24. ACM, 2015.

[FKKS17] Tobias Friedrich, Timo Kötzing, Martin S. Krejca, and Andrew M. Sutton. The compact genetic algorithm is efficient under extreme gaussian noise. *IEEE Trans. Evol. Comput.*, 21(3):477–490, 2017.

[FKL+17] Tobias Friedrich, Timo Kötzing, Gregor Lagodzinski, Frank Neumann, and Martin Schirneck. Analysis of the (1+1) EA on subclasses of linear functions under uniform and linear constraints. In Christian Igel, Dirk Sudholt, and Carsten Witt, editors, *Proceedings of the 14th ACM/SIGEVO Conference on Foundations of Genetic Algorithms, FOGA 2017, Copenhagen, Denmark, January 12-15, 2017*, pages 45–54. ACM, 2017.

[FKL+20] Tobias Friedrich, Timo Kötzing, J. A. Gregor Lagodzinski, Frank Neumann, and Martin Schirneck. Analysis of the (1+1) EA on subclasses of linear functions under uniform and linear constraints. *Theor. Comput. Sci.*, 832:3–19, 2020.

[FKM17] Tobias Friedrich, Timo Kötzing, and Anna Melnichenko. Analyzing search heuristics with differential equations. In Peter A. N. Bosman, editor, *Genetic and Evolutionary Computation Conference, Berlin, Germany, July 15-19, 2017, Companion Material Proceedings*, pages 313–314. ACM, 2017.

[FKN+23] Tobias Friedrich, Timo Kötzing, Aneta Neumann, Frank Neumann, and Aishwarya Radhakrishnan. Analysis of (1+1) EA on leadingones with constraints. In Sara Silva and Luís Paquete, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2023, Lisbon, Portugal, July 15-19, 2023*, pages 1584–1592. ACM, 2023.

[FKS16] Tobias Friedrich, Timo Kötzing, and Andrew M. Sutton. On the robustness of evolving populations. In Julia Handl, Emma Hart, Peter R. Lewis, Manuel López-Ibáñez, Gabriela Ochoa, and Ben Paechter, editors, *Parallel Problem Solving from Nature - PPSN XIV - 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings*, volume 9921 of *Lecture Notes in Computer Science*, pages 771–781. Springer, 2016.

[GK14] Christian Gießen and Timo Kötzing. Robustness of populations in stochastic environments. In Dirk V. Arnold, editor, *Genetic and Evolutionary Computation Conference, GECCO '14, Vancouver, BC, Canada, July 12-16, 2014*, pages 1383–1390. ACM, 2014.

[GK16] Christian Gießen and Timo Kötzing. Robustness of populations in stochastic environments. *Algorithmica*, 75(3):462–489, 2016.

[GKS99] Josselin Garnier, Leila Kallel, and Marc Schoenauer. Rigorous hitting times for binary mutations. *Evol. Comput.*, 7(2):173–203, 1999.

[GLR20] Leslie Ann Goldberg, John Lapinskas, and David Richerby. Phase transitions of the moran process and algorithmic consequences. *Random Struct. Algorithms*, 56(3):597–647, 2020.

[Haj82] Bruce Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied Probability*, 14(3):502–525, 1982.

[Her18] Jorge Pérez Heredia. Modelling evolutionary algorithms with stochastic differential equations. *Evol. Comput.*, 26(4), 2018.

[HJZ19] Jun He, Thomas Jansen, and Christine Zarges. Unlimited budget analysis. In Manuel López-Ibáñez, Anne Auger, and Thomas Stützle, editors, *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2019, Prague, Czech Republic, July 13-17, 2019*, pages 427–428. ACM, 2019.

[HOS19] George T. Hall, Pietro S. Oliveto, and Dirk Sudholt. On the impact of the cutoff time on the performance of algorithm configurators. In Anne Auger and Thomas Stützle, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019, Prague, Czech Republic, July 13-17, 2019*, pages 907–915. ACM, 2019.

[HY01] Jun He and Xin Yao. Drift analysis and average time complexity of evolutionary algorithms. *Artif. Intell.*, 127(1):57–85, 2001.

[HY04] Jun He and Xin Yao. A study of drift analysis for estimating computation time of evolutionary algorithms. *Nat. Comput.*, 3(1):21–35, 2004.

[Jan20] Thomas Jansen. Analysing stochastic search heuristics operating on a fixed budget. In Benjamin Doerr and Frank Neumann, editors, *Theory of Evolutionary Computation - Recent Developments in Discrete Optimization*, Natural Computing Series, pages 249–270. Springer, 2020.

[JL22] Duri Janett and Johannes Lengler. Two-dimensional drift analysis: - optimizing two functions simultaneously can be hard. In Günter Rudolph, Anna V. Kononova, Hernán E. Aguirre, Pascal Kerschke, Gabriela Ochoa, and Tea Tusar, editors, *Parallel Problem Solving from Nature - PPSN XVII - 17th International Conference, PPSN 2022, Dortmund, Germany, September 10-14, 2022, Proceedings, Part II*, volume 13399 of *Lecture Notes in Computer Science*, pages 612–625. Springer, 2022.

[Joh10] Daniel Johannsen. *Random Combinatorial Structures and Randomized Search Heuristics.* PhD thesis, Universität des Saarlandes, 2010.

[JZ12] Thomas Jansen and Christine Zarges. Fixed budget computations: a different perspective on run time analysis. In Terence Soule and Jason H. Moore, editors, *Genetic and Evolutionary Computation Conference, GECCO '12, Philadelphia, PA, USA, July 7-11, 2012*, pages 1325–1332. ACM, 2012.

[JZ14] Thomas Jansen and Christine Zarges. Reevaluating immune-inspired hypermutations using the fixed budget perspective. *IEEE Trans. Evol. Comput.*, 18(5):674–688, 2014.

[KK18] Timo Kötzing and Martin S. Krejca. First-hitting times for finite state spaces. In Anne Auger, Carlos M. Fonseca, Nuno Lourenço, Penousal Machado, Luís Paquete, and L. Darrell Whitley, editors, *Parallel Problem Solving from Nature - PPSN XV - 15th International Conference, Coimbra, Portugal, September 8-12, 2018, Proceedings, Part II*, volume 11102 of *Lecture Notes in Computer Science*, pages 79–91. Springer, 2018.

[KK19] Timo Kötzing and Martin S. Krejca. First-hitting times under drift. *Theor. Comput. Sci.*, 796:51–69, 2019.

[KLW15] Timo Kötzing, Andrei Lissovoi, and Carsten Witt. (1+1) EA on generalized dynamic onemax. In Jun He, Thomas Jansen, Gabriela Ochoa, and Christine Zarges, editors, *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII, Aberystwyth, United Kingdom, January 17 - 20, 2015*, pages 40–51. ACM, 2015.

[KM12] Timo Kötzing and Hendrik Molter. ACO beats EA on a dynamic pseudo-boolean function. In Carlos A. Coello Coello, Vincenzo Cutello, Kalyanmoy Deb, Stephanie Forrest, Giuseppe Nicosia, and Mario Pavone, editors, *Parallel Problem Solving from Nature - PPSN XII - 12th International Conference, Taormina, Italy, September 1-5, 2012, Proceedings, Part I*, volume 7491 of *Lecture Notes in Computer Science*, pages 113–122. Springer, 2012.

[Köt16] Timo Kötzing. Concentration of first hitting times under additive drift. *Algorithmica*, 75(3):490–506, 2016.

[Kre19] Martin S. Krejca. *Theoretical analyses of univariate estimation-of-distribution algorithms.* PhD thesis, Universität Potsdam, 2019.

[KSNO12] Timo Kötzing, Andrew M. Sutton, Frank Neumann, and Una-May O'Reilly. The max problem revisited: the importance of mutation in genetic programming. In Terence Soule and Jason H. Moore, editors, *Genetic and Evolutionary Computation Conference, GECCO '12, Philadelphia, PA, USA, July 7-11, 2012*, pages 1333–1340. ACM, 2012.

[KST11] Timo Kötzing, Dirk Sudholt, and Madeleine Theile. How crossover helps in pseudo-boolean optimization. In Natalio Krasnogor and Pier Luca Lanzi, editors, *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July 12-16, 2011*, pages 989–996. ACM, 2011.

[KU18] Adrian Kosowski and Przemyslaw Uznanski. Brief announcement: Population protocols are fast. In Calvin Newport and Idit Keidar, editors, *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018*, pages 475–477. ACM, 2018.

[KW20] Timo Kötzing and Carsten Witt. Improved fixed-budget results via drift analysis. In Thomas Bäck, Mike Preuss, André H. Deutz, Hao Wang, Carola Doerr, Michael T. M. Emmerich, and Heike Trautmann, editors, *Parallel Problem Solving from Nature - PPSN XVI - 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part II*, volume 12270 of *Lecture Notes in Computer Science*, pages 648–660. Springer, 2020.

[Leh11] Per Kristian Lehre. Fitness-levels for non-elitist populations. In Natalio Krasnogor and Pier Luca Lanzi, editors, *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July 12-16, 2011*, pages 2075–2082. ACM, 2011.

[Len20] Johannes Lengler. Drift analysis. In Benjamin Doerr and Frank Neumann, editors, *Theory of Evolutionary Computation - Recent Developments in Discrete Optimization*, Natural Computing Series, pages 89–131. Springer, 2020.

[LS14] Jörg Lässig and Dirk Sudholt. General upper bounds on the runtime of parallel evolutionary algorithms. *Evol. Comput.*, 22(3):405–437, 2014.

[LS15] Johannes Lengler and Nicholas Spooner. Fixed budget performance of the (1+1) EA on linear functions. In Jun He, Thomas Jansen, Gabriela Ochoa, and Christine Zarges, editors, *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII, Aberystwyth, United Kingdom, January 17 - 20, 2015*, pages 52–61. ACM, 2015.

[LW12] Per Kristian Lehre and Carsten Witt. Black-box search by unbiased variation. *Algorithmica*, 64(4):623–642, 2012.

[LW14] Per Kristian Lehre and Carsten Witt. Concentrated hitting times of randomized search heuristics with variable drift. In Hee-Kap Ahn and Chan-Su Shin, editors, *Algorithms and Computation - 25th International Symposium, ISAAC 2014, Jeonju, Korea, December 15-17, 2014, Proceedings*, volume 8889 of *Lecture Notes in Computer Science*, pages 686–697. Springer, 2014.

[McD93] Colin McDiarmid. A random recolouring method for graphs and hypergraphs. *Combinatorics, Probability and Computing*, 2(3):363–365, 1993.

[MRC09] Boris Mitavskiy, Jonathan E. Rowe, and Chris Cannings. Theoretical analysis of local search strategies to optimize network communication subject to preserving the total number of links. *Int. J. Intell. Comput. Cybern.*, 2(2):243–284, 2009.

[MU05] Michael Mitzenmacher and Eli Upfal. *Probability and computing: randomized algorithms and probabilistic analysis.* Cambridge University Press, 2005.

[NNS17] Samadhi Nallaperuma, Frank Neumann, and Dirk Sudholt. Expected fitness gains of randomized search heuristics for the traveling salesperson problem. *Evol. Comput.*, 25(4), 2017.

[OW11] Pietro S. Oliveto and Carsten Witt. Simplified drift analysis for proving lower bounds in evolutionary computation. *Algorithmica*, 59(3):369–386, 2011.

[OW12] Pietro S. Oliveto and Carsten Witt. Erratum: Simplified drift analysis for proving lower bounds in evolutionary computation. *CoRR*, abs/1211.7184, 2012.

[OW14] Pietro S. Oliveto and Carsten Witt. On the runtime analysis of the simple genetic algorithm. *Theor. Comput. Sci.*, 545:2–19, 2014.

[Pap91] Christos H. Papadimitriou. On selecting a satisfying truth assignment (extended abstract). In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991*, pages 163–169. IEEE Computer Society, 1991.

[Row18] Jonathan E. Rowe. Linear multi-objective drift analysis. *Theor. Comput. Sci.*, 736:25–40, 2018.

[RS14] Jonathan E. Rowe and Dirk Sudholt. The choice of the offspring population size in the $(1, \lambda)$ evolutionary algorithm. *Theor. Comput. Sci.*, 545:20–38, 2014.

[ST03] Mikhail A. Semenov and Dmitri A. Terkel. Analysis of convergence of an evolutionary algorithm with self-adaptation using a stochastic lyapunov function. *Evol. Comput.*, 11(4):363–379, 2003.

[STW04] Jens Scharnow, Karsten Tinnefeld, and Ingo Wegener. The analysis of evolutionary algorithms on sorting and shortest paths problems. *J. Math. Model. Algorithms*, 3(4):349–366, 2004.

[Sud13] Dirk Sudholt. A new method for lower bounds on the running time of evolutionary algorithms. *IEEE Trans. Evol. Comput.*, 17(3):418–435, 2013.

[Weg01] Ingo Wegener. Theoretical aspects of evolutionary algorithms. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *Automata, Languages and Programming, 28th International Colloquium, ICALP 2001, Crete, Greece, July 8-12, 2001, Proceedings*, volume 2076 of *Lecture Notes in Computer Science*, pages 64–78. Springer, 2001.

[Wit06] Carsten Witt. Runtime analysis of the (mu + 1) EA on simple pseudo-boolean functions. *Evol. Comput.*, 14(1):65–86, 2006.

[Wit13] Carsten Witt. Tight bounds on the optimization time of a randomized search heuristic on linear functions. *Comb. Probab. Comput.*, 22(2):294–318, 2013.

[Wit14] Carsten Witt. Fitness levels with tail bounds for the analysis of randomized search heuristics. *Inf. Process. Lett.*, 114(1-2):38–41, 2014.

[Wit23] Carsten Witt. How majority-vote crossover and estimation-of-distribution algorithms cope with fitness valleys. *Theor. Comput. Sci.*, 940(Part):18–42, 2023.

[Wor99] Nicholas C. Wormald. The differential equation method for random graph processes and greedy algorithms. *Lectures on Approximation and Randomized Algorithms*, pages 73–155, 1999.