Medium step sizes are harmful for the compact genetic algorithm

Lengler, Johannes; Sudholt, Dirk; Witt, Carsten

Published in:
2018 Proceedings of the Genetic and Evolutionary Computation Conference

Link to article, DOI:
10.1145/3205455.3205576

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

Medium Step Sizes are Harmful for the Compact Genetic Algorithm

Johannes Lengler
Department of Computer Science
ETH Zürich
Zürich, Switzerland

Dirk Sudholt
Department of Computer Science
University of Sheffield
Sheffield, United Kingdom

Carsten Witt
DTU Compute
Technical University of Denmark
Kongens Lyngby, Denmark

ABSTRACT

We study the intricate dynamics of the Compact Genetic Algorithm (cGA) on OneMax, and how its performance depends on the step size $1/K$, that determines how quickly decisions about promising bit values are fixed in the probabilistic model. It is known that cGA and UMDA, a related algorithm, run in expected time $O(n \log n)$ when the step size is just small enough ($K = \Theta(\sqrt{n} \log n)$) to avoid wrong decisions being fixed. UMDA also shows the same performance in a very different regime (equivalent to $K = \Theta(\log n)$ in the cGA) with much larger steps sizes, but for very different reasons: many wrong decisions are fixed initially, but then reverted efficiently.

We show that step sizes in between these two optimal regimes are harmful as they yield larger runtimes: we prove a lower bound of $\Omega(K^{1/3}n+n \log n)$ for the cGA on OneMax for $K = O(\sqrt{n}/\log^2 n)$. For $K = \Omega(\log^3 n)$ the runtime increases with growing K before dropping again to $O(K\sqrt{n} + n \log n)$ for $K = \Omega(\sqrt{n} \log n)$. This suggests that the expected runtime for cGA is a bimodal function in K with two very different optimal regions and worse performance in between.

CCS CONCEPTS

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

KEYWORDS

Estimation-of-distribution algorithms, compact genetic algorithm, evolutionary algorithms, running time analysis, theory.

ACM Reference Format:
Johannes Lengler, Dirk Sudholt, and Carsten Witt. 2018. Medium Step Sizes are Harmful for the Compact Genetic Algorithm. In *GECCO '18: Genetic and Evolutionary Computation Conference, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3205455.3205576

1 INTRODUCTION

Estimation-of-distribution algorithms (EDAs) are general metaheuristics for optimisation that represent a more recent alternative to classical approaches like evolutionary algorithms (EAs). EDAs typically do not directly evolve populations of search points but build probabilistic models of promising solutions by repeatedly sampling and selecting points from the underlying search space. Hence, information about the search can be stored in a relatively compact way, which can make EDAs space-efficient and time-efficient.

Recently, there has been significant progress in the theoretical understanding of EDAs, which supports their use as an alternative to evolutionary algorithms. It has been shown that EDAs are robust to noise [5] and that they have at least comparable runtime behaviour to EAs. Different EDAs like cGA [13], ACO [11, 13], and UMDA [8, 9, 14] have been investigated from this perspective.

In this paper, we pick up recent research about the runtime behaviour of the Compact Genetic Algorithm (cGA) [6]. The behaviour on the theoretical benchmark function OneMax is of particular interest since this function illustrates important properties and serves as a basis for the analysis on more complicated functions. Droste [2] was the first to prove that cGA is efficient on OneMax by providing a bound of $O(n^{1+\epsilon})$ on the runtime. Recently, this bound was refined to $O(n \log n)$ by Sudholt and Witt [13]. However, this bound only applies to a very specific setting of the step size $1/K$, which is an algorithm-specific parameter of the cGA. Parameters equivalent to step sizes exist in other EDAs, including the UMDA mentioned above.

The choice of the step size is crucial for EDAs. It governs the speed at which the probabilistic model is adjusted towards the structure of recently sampled good solutions. If the step size is too large, the adjustment is too greedy, it is too likely to adapt to incorrect parts of sampled solutions and the system behaves chaotically. If it is too small, adaptation takes very long. However, the dependency of the runtime of cGA and UMDA on the step size is even more subtle[1]. For both cGA and UMDA, small step sizes lead to optimal performance where with high probability all decisions are made correctly, but still as fast as possible. For UMDA it was shown that there is another, much bigger step size that allows incorrect decisions to be reflected in the probabilistic model for a while, but this is compensated by faster updates.

More concretely, the results from [13] show that for $K \geq c\sqrt{n} \log n$, where c is an appropriate constant, cGA and UMDA (with K being replaced by the corresponding parameter λ) optimise OneMax efficiently since for all bits the probabilities of sampling a one increase smoothly towards their optimal value because of the small step size $1/K$. The same holds for UMDA, leading to runtime bounds $O(K\sqrt{n})$ and $O(\lambda\sqrt{n})$, respectively. At $K = c\sqrt{n} \log n$

[1] Unfortunately, our understanding of these algorithms is somewhat fragmented, since some results are proven only for cGA and some are proven only for UMDA. However, despite their different appearances, cGA and UMDA have been shown to be closely related, and where results for both algorithms exist, they coincide. Thus we take results for the UMDA as strong indication for analogous behaviour of the cGA, and vice versa.

GECCO '18, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5618-3/18/07…$15.00
https://doi.org/10.1145/3205455.3205576

(resp. $\lambda = c\sqrt{n}\log n$) both algorithms optimise OneMax in expected time $O(n \log n)$. For smaller step sizes (larger K), at least for cGA it is known that the runtime increases as $\Omega(K\sqrt{n})$ [13].

On the other hand, it has been independently shown in [9] and [14] that the UMDA achieves the same runtime $O(n \log n)$ for $\lambda = c' \log n$ for a suitable constant c'. The bound at these very large step sizes emphasises that the search dynamics seem to proceed very differently from the dynamics at small step sizes. Namely, for many bits the model first learns incorrectly that the optimal value is 0 and then efficiently corrects this decision. The results in [9] and [14] show a general runtime bound of $O(\lambda n)$ for all $\lambda \geq c' \log n$ and $\lambda = o(\sqrt{n}\log n)$. We call this regime the *medium step size* regime, and it is separated from other regimes by two phase transitions: one for small step sizes, $K > c\sqrt{n}\log n$ as discussed above, and one for even larger step sizes, corresponding to $K = o(\log n)$, where the system behaves so chaotically that correct decisions are regularly forgotten and the expected runtime on OneMax becomes exponential[2].

We also know that the runtime of cGA is $\Omega(n \log n)$ for all K [13]. However, it remained an open question whether the runtime is $\Theta(n \log n)$ throughout the whole medium step size regime, or whether the runtime increases with K as suggested by the upper bound $O(\lambda n)$ for UMDA.

Here we show that the runtime of cGA does indeed increase. Our main result is as follows.

Theorem 1.1. *If $K = O(n^{1/2}/(\log(n)\log\log n))$ then the optimisation time of cGA on OneMax is $\Omega(K^{1/3}n + n\log n)$ with probability $1 - o(1)$ and in expectation.*

This result suggests that the runtime and the underlying search dynamics depend in an astonishingly complex way on the step size: as long as the step size is in the large regime ($K = o(\log n)$), the runtime is exponential [11]. Assuming that the upper bound for UMDA also holds for cGA, it then decreases to $O(n \log n)$ at the point where the medium regime is entered. Then the runtime grows with K in the medium regime, where it grows up to $\Omega(n^{7/6}/\log n)$. Before entering the small step size regime ($K = c\sqrt{n}\log n$) the runtime drops again to $O(n \log n)$ [13]. For even smaller step sizes (larger K) the runtime increases again [13]. Preliminary experiments confirm that the runtime indeed shows this complex bimodal behaviour.

The proof of our main theorem is technically demanding but insightful: we obtain insights into the probabilistic process governing cGA through careful drift analysis. In very rough terms, we analyse the drift of a potential function that measures the distance of the current sampling distribution to the optimal distribution. However, the drift depends on the sampling variance, which is a random variable as well. This leads to a complex feedback system between sampling variance and drift of potential function that tends to self-balance. We are confident that the approach and the tools used here yield insights that will prove useful for analysing other stochastic processes where the drift is changing over time.

This paper is structured as follows. Section 2 defines the cGA and presents fundamental properties of its search dynamics. Section 3 elaborates on the intriguing search dynamics of cGA in the medium parameter range, including a proof of the fact that many probabilities in the model initially are learnt incorrectly. Section 4 is the heart of our analysis and presents the so-called Stabilisation Lemma, proving that the sampling variance and, thereby, the drift of the potential approach a steady state during the optimisation. It starts with a general road map for the proof. Finally, Section 5 puts the whole machinery together to prove the main result.

Due to space limitations, many proofs are reduced to proof sketches. In particular, standard arguments like drift analysis and Chernoff bounds are only sketched for the sake of brevity. For background on techniques from the analysis of randomised algorithms used in this work (martingales, gambler's ruin, coupling, principle of deferred decisions) we refer to [10].

2 THE COMPACT GENETIC ALGORITHM AND ITS SEARCH DYNAMICS

The cGA, defined in Algorithm 1, uses marginal probabilities $p_{i,t}$ that correspond to the probability of setting bit i to 1 in iteration t. In each iteration two solutions x and y are being created independently using the sampling distribution $p_{1,t}, \dots, p_{n,t}$. Then the fitter offspring amongst x and y is determined, and the marginal probabilities are adjusted by a step size of $\pm 1/K$ in the direction of the better offspring for bits where both offspring differ. Here K determines the strength of the update of the probabilistic model.

The marginal probabilities are always restricted to the interval $[1/n, 1 - 1/n]$ to avoid fixation at 0 or 1. This ensures that there is always a positive probability of reaching a global optimum. Throughout the paper, we refer to $1/n$ and $1 - 1/n$ as (lower and upper) borders. We call bits *off-border* if their marginal probabilities are outside of $\{1/n, 1 - 1/n\}$.

Algorithm 1: Compact Genetic Algorithm (cGA)

```
t ← 0 and p_{1,t} ← p_{2,t} ← ··· ← p_{n,t} ← 1/2
while termination criterion not met do
    for i ∈ {1,...,n} do
        x_i ← 1 with prob. p_{i,t}, x_i ← 0 with prob. 1 − p_{i,t}
    for i ∈ {1,...,n} do
        y_i ← 1 with prob. p_{i,t}, y_i ← 0 with prob. 1 − p_{i,t}
    if f(x) < f(y) then swap x and y;
    for i ∈ {1,...,n} do
        if x_i > y_i then p'_{i,t+1} ← p_{i,t} + 1/K;
        if x_i < y_i then p'_{i,t+1} ← p_{i,t} − 1/K;
        if x_i = y_i then p'_{i,t+1} ← p_{i,t};
        p_{i,t+1} ← min{max{1/n, p'_{i,t+1}}, 1 − 1/n}
    t ← t + 1
```

Overall, we are interested in the cGA's number of *function evaluations* until the optimum is sampled; this number is typically called *runtime* or *optimisation time*. Note that the runtime is twice the number of iterations until the optimum is sampled.

The behaviour of the cGA is governed by $V_t := \sum_{i=1}^{n} p_{i,t}(1-p_{i,t})$, the sampling variance at time t. We know from previous work [11, 13] that V_t plays a crucial role in the drift of the marginal probabilities. The following lemma makes this precise by stating transition probabilities and showing that the expected drift towards higher $p_{i,t}$ values is proportional to $1/\sqrt{V_t}$.

[2] This second phase transition has been made explicit in [11] with respect to an ACO algorithm that in fact represents a simple EDA, similar to cGA.

Lemma 2.1. *Consider the cGA on OneMax such that $1/K$ divides $1/2 - 1/n$. Then $p_{i,t+1} = \min\{\max\{1/n, p'_{i,t+1}\}, 1 - 1/n\}$ where*

$$p'_{i,t+1} = \begin{cases} p_{i,t}, & \text{w. prob. } 1 - 2p_{i,t}(1-p_{i,1}) \\ p_{i,t} + \frac{1}{K}, & \text{w. prob. } \left(\frac{1}{2} + \Theta\left(1/\sqrt{V_t}\right)\right) 2p_{i,t}(1-p_{i,1}) \\ p_{i,t} - \frac{1}{K}, & \text{w. prob. } \left(\frac{1}{2} - \Theta\left(1/\sqrt{V_t}\right)\right) 2p_{i,t}(1-p_{i,1}) \end{cases} \quad (1)$$

This implies

$$\mathrm{E}[p_{i,t+1} - p_{i,t} \mid p_{i,t}] = \Theta(1) \cdot \frac{p_{i,t}(1-p_{i,t})}{K\sqrt{V_t}}$$

where the lower bound requires $p_{i,t} < 1 - 1/n$ and the upper bound requires $p_{i,t} > 1/n$.

If $1/K$ divides $1/2 - 1/n$ then the state space is always restricted to $p_{i,t} \in \{1/n, 1/n + 1/K, \ldots, 1/2, \ldots, 1 - 1/n - 1/K, 1 - 1/n\}$. In the following we tacitly assume this condition in all results.

Proof Sketch for Lemma 2.1. Note that $p'_{i,t+1} \neq p_{i,t}$ only if the offspring are sampled differently on bit i, which happens with probability $2p_{i,t}(1-p_{i,t})$, thus $\Pr\left(p_{i,t+1} = p_{i,t}\right) = 1 - 2p_{i,t}(1-p_{i,t})$. If there was no selection in the cGA, the remaining probability $2p_{i,t}(1-p_{i,t})$ would be split evenly amongst changes of $+1/K$ and $-1/K$. This is the case in most steps, namely in steps where all bits other than i show a clear majority of ones in one offspring, such that bit i has no effect on the decision whether to update with respect to x or y. Such steps are called *random walk steps (rw-steps)* in [13]. However, if the remaining bits have equal numbers of ones, and if $x_i \neq y_i$, then bit i does determine the decision whether to update with respect to x or y, so that always $p'_{i,t} = p_{i,t} + 1/K$. Such steps are called *biased steps (b-steps)* in [13]. The probability of a biased step is $\Theta(1/\sqrt{V_t})$, inversely proportional to the root of the sampling variance. The lower bound was shown in [11, proof of Lemma 1] and the upper bound follows from a general probability bound for Poisson-Binomial distributions [1].

The expectation follows from the probability bounds. □

Remark 1. *A statement very similar to Lemma 2.1 also holds for the UMDA on OneMax, even though the latter algorithm uses a sampling and update procedure that is rather different from the cGA as it can in principle lead to large changes in a single iteration. However, the expected change of a marginal probability follows the same principle as for the cGA. Roughly speaking, the results from [8] and [14] together show that the UMDA's marginal probabilities evolve according to*

$$\mathrm{E}[p_{i,t+1} - p_{i,t} \mid p_{i,t}] = \Theta(1) \cdot p_{i,t}(1-p_{i,t})/\sqrt{V_t}$$

Note that this drift is by a factor of K larger than in the cGA. However, since each iteration of the UMDA entails λ fitness evaluations, where λ is a parameter that can be compared to K in the cGA, the overall runtime is the same for both algorithms.

The progression of the cGA can be measured by considering a natural potential function: the function $\varphi_t := \sum_{i=1}^{n}(1 - p_{i,t})$ measures the distance to the "ideal" distribution where all $p_{i,t}$ are 1. While the drift on individual bits is inversely proportional to the root of the sampling variance $\sqrt{V_t}$, the following lemma shows that the drift of the potential is proportional to $\sqrt{V_t}$. It also provides a tail bound for the change of the potential.

Lemma 2.2. *Let $\varphi_t := \sum_{i=1}^{n}(1 - p_{i,t})$, then $\mathrm{E}[\varphi_t - \varphi_{t+1} \mid \varphi_t] = O(\sqrt{V_t}/K)$. Moreover, for all t such that $V_t = O(K^2)$,*

$$\Pr\left(|\varphi_t - \varphi_{t+1}| \geq \sqrt{V_t}\log n \mid \varphi_t\right) \leq n^{-\Omega(\log n)}.$$

Proof Sketch. The expectation follows from $\sum_{i=1}^{n} \frac{p_{i,t}(1-p_{i,t})}{K\sqrt{V_t}} = \frac{V_t}{K\sqrt{V_t}} = \frac{\sqrt{V_t}}{K}$ by definition of V_t and Lemma 2.1 and showing that the contribution of bits at the lower border is of smaller order.

For the second statement, $p_{i,t}$ only changes by $\pm 1/K$ with probability $2p_{i,t}(1-p_{i,t})$. We then apply Chernoff-Hoeffding bounds to bound the number of marginal probabilities that change. □

3 DYNAMICS WITH MEDIUM STEP SIZES

As described in the introduction, the cGA in the medium step size regime, corresponding to $K = o(\sqrt{n}\log n)$ and $K = \Omega(\log n)$, will behave less stable than in the small step size regime. In particular, many marginal probabilities will be reinforced in the wrong way and will walk to the lower border before the optimum is found, resulting in an expected optimisation time of $\Omega(n \log n)$ [13]. With respect to the UMDA it is known [14] that such wrong decisions can be "unlearned" efficiently, more precisely the potential φ_t improves by an expected value of $\Omega(1)$ per iteration. This implies the upper bound $O(\lambda n)$ in the medium regime, which becomes minimal for $\lambda = \Theta(\log n)$. Even though formally we have no upper bounds on the runtime of cGA on OneMax in the medium regime, we conjecture strongly that it exhibits the same behaviour as UMDA and has expected optimisation time $O(Kn)$. We finally recall that for extremely large step sizes, corresponding to $K = o(\log n)$ (resp. $\lambda = o(\log n)$), exponential runtimes seem to occur since the system contains too few states to build a reliable probabilistic model.

The following lemma shows that a linear number of bits tends to reach the upper and lower borders in the initial phase of a run.

Lemma 3.1. *Consider the cGA with $K \leq \sqrt{n}$. Then with probability $1 - 2^{-\Omega(n)}$ at least $\Omega(n)$ bits reach the lower border and at least $\Omega(n)$ bits reach the upper border within the first $O(K^2)$ iterations.*

A proof of Lemma 3.1 is essentially contained in the proof of Theorem 5 in [12], where calculations can be simplified because of the assumption on K. Details are omitted.

Bits at any lower border tend to remain there for a long time. The following statement shows that in an epoch of length $r = o(n)$ the fraction of bits at a border only changes slightly.

Definition 3.2. Let $\gamma(t)$ denote the fraction of bits at the lower border at time t.

Lemma 3.3. *For every $r = o(n)$ and every $t \leq t' \leq t + r$ with probability $1 - e^{-\Omega(r)}$ we have $\gamma(t') \geq \gamma(t) - O(r/n)$. With probability $1 - e^{-\Omega(r)}$ there is a time $t_0 = O(K^2)$ such that $\gamma_0 := \gamma(t_0) = \Omega(1)$.*
Both statements also hold for the fraction of bits at the upper border.

The proof uses that a bit at a border has to sample the opposite value in one offspring to leave the border, which has probability at most $2/n$, and applying Chernoff bounds. Details are omitted.

We now show that with high probability, every off-border bit will hit one of the borders after a short number of iterations. The proof of the following lemma uses that the probability of increasing a

marginal probability is always at least the probability of decreasing it. Hence, if every iteration was actually changing the probability, the time bound $O(K^2)$ would follow by standard arguments on the fair random walk on K states. However, the probability of changing the state is only $p_{i,t}(1-p_{i,t})$ and the additional $\log K$-factor covers that the process has to travel through states with a low probability of movement before hitting a border.

Lemma 3.4. *Consider the marginal probability $p_{i,t}$ of a bit i of the cGA with $K = \omega(1)$ on OneMax. Let T be the first time where $p_{i,t} \in \{1/n, 1-1/n\}$. Then for every initial value $p_{i,0}$ and all $r \geq 8$, $\mathrm{E}[T \mid p_{i,0}] \leq 4K^2 \ln K$ and $\Pr(T \geq rK^2 \ln K \mid p_{i,0}) \leq 2^{-\lfloor r/8 \rfloor}$.*

4 STABILISATION OF THE SAMPLING VARIANCE

Now that we have collected the basic properties of the cGA, we can give a detailed road map of the proof. We want to use a drift argument for the potential φ_t. After a short initial phase, most of the bits are at the borders, but since a linear fraction is at the lower border we start with $\varphi_t = \Omega(n)$. As we have seen, the drift of φ_t is $O(\sqrt{V_t}/K)$, so the heart of the proof is to study how V_t evolves.

However, the behaviour of V_t is complex. It is determined by the number and position of the bits in the off-border region (the other bits contribute only negligibly). By Lemma 2.1, each $p_{i,t}$ performs a random walk with (state-dependent) drift proportional to $1/\sqrt{V_t}$. Therefore, V_t affects itself in a complex feedback loop. For example, if V_t is large, then the drift of each $p_{i,t}$ is weak (not to be confused with the drift of φ_t, which is strong for large V_t). This has two opposing effects. Consider a bit that leaves the lower border. On the one hand, the bit has a large probability to be re-absorbed by this border quickly. On the other hand, if it does gain some distance from the lower border then it spends a long time in the off-border region, due to the weak drift. For small V_t and large drift, the situation is reversed. Bits that leave the lower border are less likely to be re-absorbed, but also need less time to reach the upper border. Thus the number and position of bits in the off-border region depends in a rather complex way on V_t.

To complicate things even more, the feedback loop from V_t to itself has a considerable lag. For example, imagine that V_t suddenly decreases, i.e. the drift of the $p_{i,t}$ increases. Then bits close to the lower border are less likely to return to the lower border, and this also affects bits which have already left the border earlier. On the other hand, the drift causes bits to cross the off-border region more quickly, but this takes time: bits that are initially in the off-border region will not jump to a border instantly. Thus the dynamics of V_t plays a role. For instance, if a phase of small V_t (large drift of $p_{i,t}$) is followed by a phase of large V_t (small drift of $p_{i,t}$), then in the first phase many bits reach the off-border region, and they all may spend a long time there in the second phase. This combination could not be caused by any static value of V_t.

Although the situation appears hopelessly complex, we overcome these obstacles using the following key idea: *the sampling variance V_t of all bits at time t can be estimated accurately by analysing the stochastic behaviour of one bit i over a period of time.* More specifically, we split the run of the algorithm into epochs of length $K^2\beta(n) = o(n/\log\log n)$, with $\beta(n) = C\log^2 n$ for a sufficiently large constant C, long enough that the value of V_t may take effect on the distribution of the bits. We assume that in one such epoch we know bounds $V_{\min} \leq V_t \leq V_{\max}$, and we show that, by analysing the dynamics of a single bit, (stronger) bounds $V'_{\min} \leq V_t \leq V'_{\max}$ hold for the next epoch. The following key lemma makes this precise.

Lemma 4.1 (Stabilisation Lemma). *Let $r := K^2\beta(n)$ with $K \geq C\log^3 n$ and with $\beta(n) = C\log^2 n$, for a sufficiently large constant $C > 0$. Let further $t_1 > 0$, $t_2 := t_1 + r$ and $t_3 := t_2 + r$. Assume $\gamma(t_1) = \Omega(1)$. There is $C' > 0$ such that the following holds for all $V_{\min} \in [0, K^{2/3}/C']$ and $V_{\max} \in [C'K^{4/3}, \infty]$. Assume that $V_{\min} \leq V_t \leq V_{\max}$ for all $t \in [t_1, t_2]$. Then with probability $1-q$ we have $V'_{\min} \leq V_t \leq V'_{\max}$ for the time $[t_2, t_3]$, with the following parameters.*

(a) If $V_{\min} = 0$, $V_{\max}$ arbitrary, then $V'_{\min} = \Omega(\sqrt{K})$, $V'_{\max} = \infty$, and $q = \exp(-\Omega(\sqrt{K}))$.

(b) If $V_{\min} = \Omega(\sqrt{K})$, $V_{\max}$ arbitrary, then

- $V'_{\min} = \Omega(\sqrt{K}V_{\min}^{1/4})$;
- $V'_{\max} = O(K \min\{K, \sqrt{V_{\max}}\}/\sqrt{V_{\min}})$;
- $q = \exp(-\Omega(\min\{\sqrt{V_{\min}}, \sqrt{K}/V_{\min}^{1/4}\}))$.

To understand where the values of $V'_{\min}$ and $V'_{\max}$ come from, we recall that $V_t = \sum_{i=1}^{n} p_{i,t}(1-p_{i,t})$, and we regard the terms $p_{i,t}(1-p_{i,t})$ from an orthogonal perspective. For a fixed bit i that leaves the lower border at some time t_1, we consider the total lifetime contribution of this bit to all V_t until it hits a border again at some time t_2, so we consider $P_i = \sum_{t=t_1}^{t_2} p_{i,t}(1-p_{i,t})$. Note that V_t and P_i are conceptually very different quantities, as the first one adds up contributions of all bits for a fixed time, while the second quantifies the total contribution of a fixed bit over its lifetime. Nevertheless, we show in Section 4.1 that their expectations are related, $\mathrm{E}[V_t] \approx 2\gamma(t)\mathrm{E}[P_i]$, where $2\gamma(t)$ is the expected number of bits that leave the lower border in each round.[3] Crucially, $\mathrm{E}[P_i]$ is much easier to analyse: we link $\mathrm{E}[P_i]$ to the expected hitting time $\mathrm{E}[T]$ of a rescaled and loop-free version of the random walks that the bits perform. In Section 4.2 we then derive upper and lower bounds on $\mathrm{E}[T]$ that hold for all random walks with given bounds on the drift, which then lead to upper and lower bounds $V'_{\min} \leq \mathrm{E}[V_t] \leq V'_{\max}$.

To prove Lemma 4.1, it is not sufficient to know $\mathrm{E}[V_t]$, we also need concentration for V_t. Naturally V_t is a sum of random variables $p_{i,t}(1-p_{i,t})$, so we would like to use the Chernoff bound. Unfortunately, all the random walks of the bits are correlated, so the $p_{i,t}$ are not independent. However, we show by an elegant argument in Section 4.3 that we may still apply the Chernoff bound. We partition the set of bits into m batches, and show that the random walks of the bits in each batch do not substantially influence each other. This allows us to show that the contribution of each batch is concentrated with exponentially small error probabilities. The overall proof of Lemma 4.1 is then by induction. Given that we know bounds $V_{\min}$ and $V_{\max}$ for one epoch, we show by induction over all times t in the next epoch that V_t satisfies even stronger bounds $V'_{\min}$ and $V'_{\max}$.

In Section 5 we then apply Lemma 4.1 iteratively to show that the bounds $V_{\min}$ and $V_{\max}$ become stronger with each new epoch, until we reach $V_{\min} = \Omega(K^{2/3})$ and $V_{\max} = O(K^{4/3})$. At this point

[3] The actual statement is a bit more subtle and involves lower and upper bounds on P_i, see Lemma 4.3.

the approach reaches its limit, since then the new bounds $V'_{\min}$ and $V'_{\max}$ are no longer sharper than $V_{\min}$ and $V_{\max}$. Still, the argument shows that $V_t = O(K^{4/3})$ from this point onwards, which gives us an upper bound of $O(K^{-1/3})$ on the drift of φ_t and a lower bound of $\Omega(K^{1/3}n)$ on the runtime of the algorithm.

As the proof outline indicates, the key step is to prove Lemma 4.1, and the rest of the section is devoted to it.

4.1 Connecting V_t to the Lifetime of a Bit

In this section we will lay the foundation to analyse $\mathrm{E}[V_t]$. We consider the situation of Lemma 4.1, i.e., we assume that we know bounds $V_{\min} \le V_t \le V_{\max}$ that hold for an epoch $[t_1, t_2]$ of length $t_2 - t_1 = r = K^2\beta(n)$. We want to compute $\mathrm{E}[V_t]$ for a fixed $t \in [t_2, t_3]$. Since $V_t = \sum_{i=1}^{n} p_{i,t}(1-p_{i,t})$, we call the term $p_{i,t}(1-p_{i,t})$ the *contribution* of the i-th bit to V_t. The main result of this section (and one of the main insights of the paper) is that the contribution of the off-border bit can be described by $\mathrm{E}[V_t] = \Theta(\gamma(t)\mathrm{E}[T])$, where T is the lifetime of a random variable that performs a rescaled and loop-free version of the random walk that each $p_{i,t}$ performs.

First we introduce the rescaled and loop-free random walk. It can be described as the random walk that $p_{i,t}$ performs for an individual bit if we ignore self-loops, i.e., if we assume that in each step $p_{i,t}$ either increases or decreases by $1/K$. Moreover, it will be convenient to scale the random walk by roughly a factor of K so that the borders are 0 and K instead of $1/n$ and $1-1/n$. The exact scaling is given by the formula $X_{i,t} = (p_{i,t} - 1/n)/(K - 2/n)$. Formally, assume that X_t is a random walk on $\{0, \dots, K\}$ where the following bounds hold whenever $X_t \in \{1, \dots, K-1\}$.

$$X_{t+1} = \begin{cases} X_t + 1, & \text{w. prob. } \frac{1}{2} + d(t), \\ X_t - 1, & \text{w. prob. } \frac{1}{2} - d(t), \end{cases} \tag{2}$$

$$\text{where } d(t) = \Omega\big(1/\sqrt{V_{\max}}\big) \text{ and } d(t) = O\big(1/\sqrt{V_{\min}}\big).$$

Note that by Lemma 2.1, if we condition on $p_{i,t+1} \neq p_{i,t}$ then $p_{i,t}$ follows a random walk that increases with probability $1/2 + \Theta(1/\sqrt{V_t})$. Hence, if $V_{\min} \le V_t \le V_{\max}$ then this loop-free random walk of $p_{i,t}$ follows the description in (2) after scaling. Therefore, we will refer to the random walk defined by (2) as the *loop-free random walk* of a bit. We remark that it is slight abuse of terminology to speak of *the* loop-free random walk, since (2) actually describes a class of random walks. Formally, when we prove upper and lower bounds on the hitting time of "the" loop-free random walk, we prove bounds on the hitting time of any random walk that follows (2).

To link $\mathrm{E}[V_t]$ and $\mathrm{E}[T]$, we need one more seemingly unrelated concept. Consider a bit i that leaves the lower border at some time t_0, i.e., $p_{i,t_0-1} = 1/n$ and $p_{i,t_0} = 1/n + 1/K$, and let $t' > 0$ be the first point in time when $p_{i,t}$ hits a border, so $p_{i,t'} = 1/n$ or $p_{i,t'} = 1-1/n$. Then we call

$$P_i := \sum_{t=t_0}^{t'-1} p_{i,t}(1-p_{i,t}), \quad \text{where } p_{i,t_0} = 1/n + 1/K \tag{3}$$

the *lifetime contribution* of the i-th bit. Analogously, we denote by P'_i the lifetime contribution if bit i leaves the upper border,

$$P'_i := \sum_{t=t_0}^{t'-1} p_{i,t}(1-p_{i,t}), \quad \text{where } p_{i,t_0} = 1 - 1/n - 1/K. \tag{4}$$

Note that V_t and P_i are both sums over terms of the form $p_{i,t}(1-p_{i,t})$. But while V_t sums over all i for fixed t, P_i sums over some values of t for a fixed i. Nevertheless, as announced in the proof outline, we will show that the expectations $\mathrm{E}[V_t]$ and $\mathrm{E}[P_i]$ are closely related, and this will be the link between $\mathrm{E}[V_t]$ and $\mathrm{E}[T]$. More precisely, we show the following lemma.

Lemma 4.2. *Consider the situation of Lemma 4.1. Let $t \in [t_2, t_3]$, and assume $V_{\min} \le V_{t'} \le V_{\max}$ for all $t' \in [t_1, t-1]$. Let S_{low} be the set of all bits i with $p_{i,t} \notin \{1/n, 1-1/n\}$, and such that their last visit of a border was in $[t_1, t]$, and it was at the lower border. Formally, we require that $t_0 := \max\{\tau \in [t_1, t] \mid p_{i,\tau} \in \{1/n, 1-1/n\}\}$ exists and that $p_{i,t_0} = 1/n$. Let S_{upp} be the analogous set, where the last visit was at the upper border. Then*

(a) $\mathrm{E}[\sum_{i \in S_{\text{low}}} p_{i,t}(1-p_{i,t})] = \Theta(\mathrm{E}[P_i])$.
(b) $\mathrm{E}[\sum_{i \in S_{\text{upp}}} p_{i,t}(1-p_{i,t})] = \Theta(\mathrm{E}[P'_i])$.
(c) $\mathrm{E}[\sum_{i \in \{1,\dots,n\} \setminus (S_{\text{low}} \cup S_{\text{upp}})} p_{i,t}(1-p_{i,t})] = O(1)$.

Proof. *(a)* Recall that we assume $\gamma(t_1) = \Omega(1)$. Since $\gamma(t)$ is slowly changing by Lemma 3.3, there is a constant $c > 0$ such that $c \le \gamma(t) \le 1$ for all $t \in [t_1, t_3]$. In particular, for every $t' \in [t_1, t_3]$, the expected number of bits $s(t)$ which leave the lower border at time t is $\mathrm{E}[s(t)] = \gamma(t)n \cdot \frac{2}{n}(1 - \frac{1}{n}) = (2 - o(1))\gamma(t) = \Theta(1)$.

Consider a bit that leaves the lower border at time 0, and let $\rho_t := p_{i,t}(1-p_{i,t})$ if i has not hit a border in the interval $[1, t]$, and $\rho_t := 0$ otherwise. Let $E_t := \mathrm{E}[\rho_t]$. Then $\mathrm{E}[P_i] = \sum_{t=0}^{\infty} E_t$. On the other hand, for a fixed $t \in [t_2, t_3]$ let us estimate $V_{t,\text{low}} := \sum_{i \in S_{\text{low}}} p_{i,t}(1-p_{i,t})$. Assume that bit i leaves the border at some time $t - \tau \in [t_1, t]$. If it does not hit a border until time t, then it contributes ρ_τ to $V_{t,\text{low}}$. The same is true if it does hit a border, and doesn't leave the lower border again in the remainder of the epoch, since then $i \notin S_{\text{low}}$ and $\rho_\tau = 0$. For the remaining case, assume that i leaves the lower border several times $t - \tau_1, t - \tau_2, \dots, t - \tau_k$, with $\tau_1 < \tau_2 < \dots < \tau_k$. Then $\rho_{\tau_2} = \dots = \rho_{\tau_k} = 0$, and by the same argument as before, the contribution of i to $V_{t,\text{low}}$ is $\rho_{\tau_1} = \sum_{i=1}^{k} \rho_{\tau_k}$, where ρ_{τ_1} may or may not be zero. Therefore, we can compute $\mathrm{E}[V_{t,\text{low}}]$ by summing up a term E_τ for every bit that leaves the lower border at time $t - \tau$, counting bits multiple times if they leave the lower border multiple times. Recall that the number of bits $s(t)$ that leave the lower border at time $t - \tau$ has expectation $\mathrm{E}[s(t)] = \Theta(1)$. Therefore,

$$\mathrm{E}[V_{t,\text{low}}] = E\Big[\sum\nolimits_{\tau=0}^{t-t_1} s_{t-\tau} \cdot E_\tau\Big] = \Theta(1) \sum\nolimits_{\tau=0}^{t-t_1} E_\tau. \tag{5}$$

The sum on the right hand side is almost $\mathrm{E}[P_i]$, except that the sum only goes to $t - t_1$ instead of ∞. Thus we need to argue that $\sum_{\tau=t-t_1+1}^{\infty} E_\tau$ is not too large. By Lemma 3.4 the probability that a bit does not hit a border state in $\tau > t - t_1 \ge r = K^2\beta(n)$ rounds is $e^{-\Omega(\tau/(K^2 \log K))}$. Hence, we may split the range $[t - t_1 + 1, \infty)$ into subintervals of the form $[i \cdot K^2 \log K, (i+1) \cdot K^2 \log K)$, then the i-th subinterval contributes $O((K^2 \log K)e^{-i})$. Therefore, setting $i_0 := \beta(n)/\log K$, the missing part of the sum is at most

$$\sum_{\tau=r}^{\infty} e^{-\Omega(\tau/(K^2 \log K))} = O\big(K^2 \log K \sum\nolimits_{i=i_0}^{\infty} e^{-i}\big) = o(1/K)$$

since $\beta = C \log^2 n$ for a sufficiently large constant C. This is clearly smaller than the rest of the sum, since already $E_1 \ge 1/K \cdot (1 - 1/K)$. Hence $\mathrm{E}[V_{t,\text{low}}] = \Theta(\mathrm{E}[P_i])$, as required.

The proof of (b) is analogous to (a). Finally, (c) follows from Lemma 3.4. We omit the details. □

The next lemma links the lifetime contribution P_i and P_i' to the hitting time T of the loop-free random walk.

Lemma 4.3. *Consider the situation of Lemma 4.1. Assume for $i = 1$ or $i = K-1$ that $T_{i,\min}$ and $T_{i,\max}$ are a lower and upper bound, respectively, on the expected hitting time of $\{0, K\}$ of every random walk as in (2) with $X_0 = i$. Then the lifetime contributions P_i and P_i' defined in (3) and (4) satisfy*

$$2T_{1,\min} \le \mathrm{E}[P_i] \le 2T_{1,\max}.$$

$$2T_{K-1,\min} \le \mathrm{E}[P_i'] \le 2T_{K-1,\max}.$$

We say that $\mathrm{E}[P_i] = \Theta(\mathrm{E}[T])$, where T is the hitting time of the loop-free random walk starting at 1, and similarly for $\mathrm{E}[P_i']$.

Proof Sketch. Bit i contributes $p_{i,t}(1-p_{i,t})$ to P_i, and the expected time until bit i makes a non-loop step is $1/(2p_{i,t}(1-p_{i,t}))$ by Lemma 2.1. Thus the total contribution to P_i per non-loop step is in expectation exactly $1/2$. The claims then follow because T counts the number of non-loop steps of $p_{i,t}$. □

Lemmas 4.2 and 4.3 together yield the following corollary.

Corollary 4.4. *Consider the situation of Lemma 4.1, and let $T_{i,\min}$ and $T_{i,\max}$ be lower and upper bounds, respectively, on the expected hitting time of $\{0, K\}$ of every random walk as in (2) with $X_0 = i$. Assume $T_{1,\min} = \omega(1)$. Then for all $t \in [t_2, t_3]$,*

$$\Omega(T_{1,\min} + T_{K-1,\min}) \ni \mathrm{E}[V_t] \in O(T_{1,\max} + T_{K-1,\max})$$

By Corollary 4.4, in order to understand $\mathrm{E}[V_t]$ it suffices to analyse the expected hitting time $\mathrm{E}[T]$ of the loop-free random walk.

4.2 Bounds on the Lifetime of a Bit

We now give upper and lower bounds on the expected lifetime of every loop-free random walk, assuming that we only have lower and upper bounds $\Delta_{\min}$ and $\Delta_{\max}$ on the drift that hold the whole time. We start with the upper bound.

Lemma 4.5. *Consider a stochastic process $\{X_t\}_{t\ge 0}$ on $\{0, 1, \dots, K\}$, variables Δ_t that may depend on $X_0, \dots, X_t$ and $\Delta_{\min} > 0$, $\Delta_{\max} \ge 1/(2K)$ such that $\Pr(X_{t+1} = X_t + 1 \mid X_t < K) = 1/2 + \Delta_t$ and $\Pr(X_{t+1} = X_t - 1 \mid X_t > 0) = 1/2 - \Delta_t$ for $\Delta_{\min} \le \Delta_t \le \Delta_{\max}$. Let T be the hitting time of states 0 or K, then regardless of the choice of the Δ_t,*

$$\mathrm{E}[T \mid X_0 = 1] = O(\min\{K^2\Delta_{\max}, K\Delta_{\max}/\Delta_{\min}\}) \text{ and}$$
$$\mathrm{E}[T \mid X_0 = K-1] = O(\min\{K, 1/\Delta_{\min}\}).$$

Remark 2. *The most important term for us is $\mathrm{E}[T \mid X_0 = 1] = O(K\Delta_{\max}/\Delta_{\min})$. This is tight, i.e., there is a scheme for choosing Δ_t that yields a time of $\Omega(K\Delta_{\max}/\Delta_{\min})$ if $\Delta_{\min} = \Omega(1/K)$.*

Proof Sketch. The proof for $X_0 = 1$ fixes an intermediate state $k_0 = \Theta(1/\Delta_{\max})$ and shows, using martingale theory and the upper bound $\Delta_{\max}$ on the drift, that (1) the time to reach either state 0 or state k_0 is $O(1/\Delta_{\max})$, and (2) the probability that k_0 is reached is $O(\Delta_{\max})$. In that case, using the lower bound $\Delta_{\min}$ on the drift, the remaining time to hit state 0 or state K is $O(K/\Delta_{\min})$ by additive drift. The time from k_0 is also bounded by $O(K^2)$ as it is dominated by the expected time a fair random walk would take if state 0 was made reflecting. The statement for $X_0 = K-1$ is proved using similar arguments, starting from $K-1$ instead of k_0. □

The following lemma gives a lower bound on the lifetime of every loop-free random walk.

Lemma 4.6. *Consider a stochastic process $\{X_t\}_{t\ge 0}$ on $\{0, 1, \dots, K\}$, variables Δ_t that may depend on $X_0, \dots, X_t$ and $\Delta_{\min} > 0$, $\Delta_{\max} \ge (4\ln K)/K$ such that $\Pr(X_{t+1} = X_t + 1 \mid X_t < K) = 1/2 + \Delta_t$ and $\Pr(X_{t+1} = X_t - 1 \mid X_t > 0) = 1/2 - \Delta_t$ for $\Delta_{\min} \le \Delta_t \le \Delta_{\max}$. Let T be the hitting time of states 0 or K, then regardless of the choice of the Δ_t,*

$$\Pr\left(T > \tfrac{1}{2}K/\Delta_{\max} \mid X_0 = 1\right) = \Omega(\sqrt{\Delta_{\max}/K} + \Delta_{\min})$$

and

$$\mathrm{E}[T \mid X_0 = 1] = \Omega(\sqrt{K/\Delta_{\max}} + K\Delta_{\min}/\Delta_{\max}).$$

Remark 3. *There is a scheme for choosing Δ_t such that the bound on the expectation from Lemma 4.6 is asymptotically tight.*

Proof Sketch. The lower bound on the expectation follows immediately from the lower bounds on the probabilities. To show the latter, we couple the process with two processes $X_t^{\min}$ and $X_t^{\max}$ that always use the minimum and maximum drift $\Delta_{\min}$ and $\Delta_{\max}$, respectively. The coupling ensures that $X_t^{\min} \le X_t \le X_t^{\max}$, hence as long as $X_t^{\min} > 0$ and $X_t^{\max} < K$, the process cannot have reached a border state. We show for both coupled processes that the probability of reaching their respective borders in time $\frac{1}{2}K/\Delta_{\max}$ is small, and then apply a union bound. For the $X_t^{\max}$ process a negligibly small failure probability follows from additive drift with tail bounds [7] and the condition $\Delta_{\max} \ge (4\ln K)/K$. For the $X_t^{\min}$ process we show that the fair random walk on the integers, starting in state 1, does not reach state 0 in time $\frac{1}{2}K/\Delta_{\max}$ with probability $\Omega(\sqrt{\Delta_{\max}/K})$. In addition, the $X_t^{\min}$ process on the integers never reaches state 0 with probability $\Omega(\Delta_{\min})$ [4, page 351], which yields the second term in the claimed probability. □

4.3 Establishing Concentration

Our major tool for showing concentration will be using the Chernoff bound [3] and the Chernoff-Hoeffding bound [3].

The basic idea is that for fixed t, we define for each bit i a random variable $X_i := p_{i,t}(1-p_{i,t})$ to capture the contribution of the i-th bit to $V_t = \sum_{i=1}^n X_i$. In the previous sections we have computed $\mathrm{E}[V_t]$ by studying the expected lifetime $\mathrm{E}[T]$. Concentration of V_t would follow immediately by the Chernoff bound if the random walks of the different bits were independent of each other. Unfortunately, this is not the case. However, for the initial case of the stabilisation lemma, Lemma 4.1 (a), we show that the random walks behave almost independent, which allows us to show the following lemma.

Lemma 4.7. *Assume the situation of Lemma 4.1 (a). Then $V_t = \Omega(\sqrt{K})$ holds with probability $1 - e^{-\Omega(\sqrt{K})}$ for all $t \in [t_2, t_3]$.*

Proof Sketch. We use an inductive argument over $t \in [t_2, t_3]$. Note that if we choose the constant C' in Lemma 4.1 large enough, then we have $V'_{\min} \ge V_{\min}$ and $V'_{\max} \le V_{\max}$. Therefore, by induction hypothesis we may assume that $V_{\min} \le V'_{\min} \le V_{t'} \le V_{\max} \le V_{\max}$ also holds for $t' \in [t_2, t-1]$.

As mentioned above, we know that $\mathrm{E}[V_t] = \mathrm{E}[T] = \Omega(\sqrt{K})$ by Lemma 4.6 with trivial drift bounds $\Delta_{\min} = 0$ and $\Delta_{\max} = 1/2$, so it remains to show concentration. Fix $i \in \{1, \dots, n\}$, and consider

the random walk that $p_{i,t}$ performs over time. More precisely, we consider one step of this random walk, from t to $t+1$. If the offspring x and y have the same i-th bit, then $p_{i,t+1} = p_{i,t}$, so assume that x and y differ in the i-th bit. We want to understand how the drift of $p_{i,t}$ changes if we condition on what the other bits do.

So assume that we have already drawn all bits of the two offspring x and y at time $t+1$ except for the i-th bit. Assume also that someone tells us which of x, y is the selected offspring. Then conditioning on all this information does influence (and sometimes determine) the behaviour of $p_{i,t}$, However, one can show that even after conditioning, $p_{i,t}$ still has non-negative drift. This allows us to couple the $p_{i,t}$ to *independent* random walks, and to apply the Chernoff bound. We omit the details. □

We would like to use a similar argument also in the cases with non-trivial $\Delta_{\min}$ and $\Delta_{\max}$. Unfortunately, it is no longer true that the drift remains lower bounded by $\Delta_{\min} > 0$ if we uncover the random walk steps of the other bits. However, the bound still remains true if we condition on *only a few* of the other bits. More precisely, if we consider a batch of r bits $b_1, \dots, b_r$ for a suitably chosen $r \in \mathbb{N}$, then even if we condition on the values that the two offspring have in the bits $b_1, \dots, b_{r-1}$ then bit b_r will still perform a random walk where the drift in each round is in $\Theta(1/(K\sqrt{V_t}))$. Hence, we can couple the random walks of $b_1, \dots, b_{r-1}$ to r *independent* random walks, and apply the Chernoff bound to show that the contribution of this batch is concentrated. Afterwards we use a union bound over all batches.

Formally, we show the following pseudo-independence lemma. Note that there are two types of error events in the lemma. One is the explicit event $\mathcal{E}$, the other is the event that $\boldsymbol{B} \notin B$, i.e., that the other bits in the batch display an atypical distribution. However, both events are very unlikely if V_t is large, which we may assume after one application of Lemma 4.7.

Lemma 4.8. *Consider a vector of probabilities $\boldsymbol{p}_t$ with potential $V_t = \sum_{i=1}^n p_{i,t}(1-p_{i,t})$.*

Let $m = m(n) \geq 3$. Let $S \subseteq \{1, \dots, n\}$ be a random set which contains each bit independently with probability $1/m$. Then there is an error event $\mathcal{E}$ of probability $\Pr(\mathcal{E}) = e^{-\Omega(V_t/m)}$ such that, conditioned on $\neg\mathcal{E}$, the following holds for all $i_0 \in S$. Let b_i^1 and b_i^2 be the i-th bit in the first and second offspring, respectively, and let $\boldsymbol{B} := (b_i^j)_{i \in S\setminus\{i_0\}, j\in\{1,2\}}$. There is a set $B \subseteq \{0,1\}^{2(m-1)}$ such that $\Pr(\boldsymbol{B} \in B) = 1 - e^{-\Omega(\min\{m, V_t/m\})}$ and such that for all $\vec{B} \in B$,

$$\mathrm{E}[p_{i_0,t+1} - p_{i_0,t} \mid \boldsymbol{p}_t, \boldsymbol{B} = \vec{B}, \neg\mathcal{E}] \in \Omega\left(\frac{p_{i_0,t}(1-p_{i_0,t})}{K\sqrt{V_t}}\right), \text{ and}$$

$$\mathrm{E}[p_{i_0,t+1} - p_{i_0,t} \mid \boldsymbol{p}_t, \boldsymbol{B} = \vec{B}, \neg\mathcal{E}] \in O\left(\frac{p_{i_0,t}(1-p_{i_0,t})}{K\sqrt{V_t}}\right). \quad (6)$$

Proof Sketch. The error event $\mathcal{E}$ is that the contribution of S to V_t deviates from its expectation V_t/m by more than a factor of 2, which is unlikely by Chernoff bounds. For a set $A \subseteq \{1, \dots, n\}$, let d_A be the difference of the fitnesses between the two offspring caused by the bits in A. Then the set B is defined by $B := \{\vec{B} \in \{0,1\}^{2(m-1)} \mid |d_{S\setminus\{i\}}| \leq \eta\sqrt{V_t}\}$ for a small constant η, and it is unlikely that $\boldsymbol{B} \notin B$ by a careful application of the Chernoff-Hoeffding bounds. The drift of $p_{i,t}$ comes from the cases in which $d_{\{1,\dots,n\}\setminus\{i\}} \in \{-1, 0, 1\}$, in which it may influence selection. However, for $\vec{B} \in B$ we have $d_{S\setminus\{i\}} = k$ for some $|k| \leq \eta\sqrt{V_t}$. For every such k, the probability that $d_{\{1,\dots,n\}\setminus S} = -k$ (or $= -k+1$ or $= -k-1$) is $\Theta(1/\sqrt{V_t})$ [1, 14]. Thus the probability that i influences selection is asymptotically the same as in the proof of Lemma 2.1, and therefore the resulting drift is also asymptotically the same. □

Lemma 4.8 allows us to partition the bits randomly into m batches, such that in each batch the bits perform random walks that can be coupled to *independent* random walks. In particular, we will be able to apply the Chernoff-Hoeffding bounds to each batch. This gives concentration of the V_t as follows.

Lemma 4.9. *Assume the situation of Lemma 4.1 (b), in particular $V'_{\min} = \Omega(\sqrt{K}V_{\min}^{1/4})$ and $V'_{\max} = O(K\min\{K, \sqrt{V_{\max}/V_{\min}}\})$ where we may choose the hidden constants suitably. Then with probability $1 - \exp(-\Omega(\min\{\sqrt{V_{\min}}, \sqrt{K}/V_{\min}^{1/4}\}))$, for all $t \in [t_2, t_3]$, we have $V'_{\min} \leq V_t \leq V'_{\max}$.*

Proof. Apart from the complication with the batches, the proof is analogous to the proof of Lemma 4.7. We omit the details. □

Altogether, we have proven the Stabilisation Lemma 4.1: part (a) is proven in Lemma 4.7, and part (b) is proven in Lemma 4.9.

5 PROOF OF THE MAIN RESULT

Lemma 5.1. *With probability $1-\exp(-\Omega(K^{1/4}))$, $V_{\min} = \Omega(K^{2/3})$ and $V_{\max} = O(K^{4/3})$ after $i^* = O(\log\log K)$ epochs of length $r = K^2\beta(n)$.*

Moreover, for any fixed $t \geq i^*r$, as long as $\gamma(\tau) = \Omega(1)$ for all $\tau \in [i^*r, t-1]$, $V_{\max}$ and $V_{\min}$ are bounded in the same way during $[i^*r, t]$, with a failure probability of at most $t/r \cdot \exp(-\Omega(K^{1/3}))$, and with probability $1 - tn\exp(-\Omega(\beta(n)/\log n))$ the number of off-border bits at any time $t \in [i^*r, t]$ is at most $4K^2\beta(n)$. In particular, if $t = n^2$, $\beta(n) = C\log^2 n$, and $K \geq C\log^3 n$ for a sufficiently large constant $C > 0$, then the error probability is $o(1)$.

Proof Sketch. All subsequent statements hold with some error probability, which we omit due to space restrictions. By Lemma 3.3, we know that the initial fraction of marginal probabilities at the lower border is $\Omega(1)$. We apply the first statement of the Stabilisation Lemma 4.1 (a) with respect to an initial epoch of length r and obtain that $V_t = \Omega(K^{1/2})$ in an epoch $[t_2, t_3]$ of length at least r. Applying the statement again, now with respect to this epoch and with the assumption $V_{\min} = \Omega(K^{1/2})$, we obtain $V_{\min} = \Omega(K^{5/8})$ for the next epoch. Iterating this argument i times, we have $V_{\min} = \Omega(K^{2/3-(2/3)(1/4)^{i+1}})$ after i epochs of length r. Choosing $i^* = c\ln\ln K$ for a sufficiently large constant $c > 0$, we get $V_{\min} = \Omega(K^{2/3-1/\log K}) = \Omega(K^{2/3})$ after $i^*/2$ iterations.

Applying part (b) of the Stabilisation Lemma 4.1 with respect to the i^*-th epoch, we obtain that $V_{\max} = O(K^2)$ for the next epoch. We apply the statement again, and the next epoch will satisfy $V_{\max} = O(K\sqrt{K^2/K^{2/3}}) = O(K^{5/3})$. Iterating this argument using the new value of $V_{\max}$ and still $V_{\min} = \Omega(K^{2/3})$ for $O(\log\log K)$ epochs similarly as above, we arrive at $V_{\max} = O(K^{4/3})$.

For $t \geq i^* r$, we may apply the same argument again, and the statement on $V_{\min}$ and $V_{\max}$ then follows from a union bound over all epochs. For the number of off-border bits, by Lemma 3.4 every bit hits a border after at most $K^2\beta(n)$ rounds. Since the probability that a fixed bit leaves the border is $2 \cdot 1/n \cdot (1-1/n)$ in each round, the expected number of bits that leave the border is at most 2 per round. Thus the expected number of non-border bits at time t is at most $2K^2\beta(n)$, and concentration follows by a union bound.

Finally, the statement for $t = n^2$ follows since $n^2 e^{-\Omega(\log n)} = o(1)$ if the hidden constant is large enough. □

We are finally ready to prove our main result.

Proof of Theorem 1.1. A lower bound of $\Omega(\sqrt{n}K + n\log n)$ was shown in [13]. Hence it suffices to show a lower bound of $\Omega(K^{1/3}n)$ for $K \geq C\log^3 n$, where we may choose the constant C to our liking. In the following, we assume that all events that occur with high probability do occur.

Recall that the potential $\varphi_t := \sum_{i=1}^n (1-p_{i,t})$ is the total distance of all marginal probabilities to the optimal value of 1. By Lemma 3.3, we have a $\gamma_0 = \Omega(1)$ fraction of bits at the lower border at some time within the first $O(K^2)$ iterations with probability $1 - e^{-\Omega(K^2\beta(n))}$. In particular, this implies $\varphi_t \geq \gamma_0(n-1)$.

We show that the expected time until either φ_t has decreased to $\gamma_0/4 \cdot (n-1)$ or the global optimum is found is $\Omega(K^{1/3}n)$ with high probability. This implies the claim since in an iteration where $\varphi_t > \gamma_0/4 \cdot (n-1)$ the probability of sampling the optimum is exponentially small: for fixed φ_t, the best case scenario for sampling the optimum is that all bits have equal values. Hence the probability of sampling the optimum is at most $(\varphi_t/n)^n = 2^{-\Omega(n)}$, which still holds when considering a union bound over $O(K^{1/3}n)$ steps.

By Lemma 5.1, with probability $\exp(-\Omega(K^{1/4})) = o(1)$ we will have $V_t = O(K^{4/3})$ after $T = O(r\log\log K) = o(n)$ steps. By Lemma 3.3, with high probability we will still have at least $\gamma_0/2 \cdot (n-1)$ bits at the lower border.

Moreover, also by Lemma 5.1, if we can show $\gamma(t) = \Omega(1)$ then the bound $V_t = O(K^{4/3})$ remains true for the next $K^{1/3}n$ rounds, with probability $1 - o(1)$. So it remains to show $\gamma(t) = \Omega(1)$ for $t \in [T, \Omega(K^{1/3}n)]$. Note that the prerequisites of Lemma 5.1 only concern times strictly before t, so we can use the statement of the lemma inductively to show that $\gamma(t) = \Omega(1)$. By Lemma 5.1, the number of off-border bits in each epoch is $O(K^2\beta(n))$, hence while $\varphi_t > \gamma_0/4 \cdot (n-1)$, we have $\gamma(t) \geq \gamma_0/4 - O(K^2\beta(n)/n) = \Omega(1)$ as off-border bits (and bits at the upper border) only contribute $O(K^2\beta(n)) = o(n)$ to φ_t. Hence Lemma 5.1 implies that with probability $1 - o(1)$, $V_t = O(K^{4/3})$ holds for all $t \in [T, n^2]$ such that $\varphi_t > \gamma_0/4 \cdot (n-1)$.

By Lemma 2.2, the drift of φ_t is at most $O\big(\sqrt{V_t}/K\big) = O(K^{-1/3})$ and the change of φ_t is bounded by $\sqrt{V_t}\log n = O(K^{2/3}\log n)$ with probability $1 - n^{-\Omega(\log n)}$, even when taking a union bound over $O(K^{1/3}n)$ steps. Applying Theorem 1 in [7] with a maximum step size of $O(K^{2/3}\log n)$, distance $\gamma_0/4 \cdot (n-1)$ and drift $O(K^{-1/3})$, the time until $\varphi_t \leq \gamma_0/4 \cdot (n-1)$ is at least $\Omega(\gamma_0/4 \cdot (n-1) \cdot K^{1/3}) = \Omega(K^{1/3}n)$ with probability $1 - e^{-\Omega\left(n \cdot K^{-1/3}/(K^{4/3}\log^2 n)\right)} = 1 - e^{-\Omega(n^{1/6}/\log^2 n)}$, where the last step uses $K = O(n^{1/2})$. Adding up failure probabilities completes the proof. □

6 CONCLUSIONS

We have shown a lower bound of $\Omega(K^{1/3}n + n\log n)$ for the cGA on OneMax that at its core has a very careful analysis of the dynamic behaviour of the sampling variance and how it stabilises in a complex feedback loop that exhibits a considerable lag. A key idea to handle this complexity was to show that the sampling variance V_t of all bits at time t can be estimated accurately by analysing the stochastic behaviour of one bit i over a period of time.

Assuming that cGA has the same upper bound as UMDA for step sizes $K = \Theta(\log n)$, the expected optimisation time of cGA is a bimodal function in K with worse performance in between its two minima.

We believe that our analysis can be extended towards an upper bound of $O(K^{2/3}n + n\log n)$, using that typically $V_t = \Omega(K^{2/3})$ after an initial phase, which implies a drift of $\Omega(\sqrt{V_t}/K) = \Omega(K^{-2/3})$ for φ_t. This would require additional arguments to deal with $\gamma(t)$ decreasing to sub-constant values where showing concentration becomes more difficult. Another avenue for future work would be to investigate whether the results and techniques carry over to the UMDA, where the marginal probabilities can make larger steps.

ACKNOWLEDGMENTS

This paper was initiated at Dagstuhl seminar 17101 "Theory of Randomized Optimization Heuristics" and is based upon work from COST Action CA15140 'Improving Applicability of Nature-Inspired Optimisation by Joining Theory and Practice (ImAppNIO)' supported by COST (European Cooperation in Science & Technology).

REFERENCES

[1] J.-B. Baillon, R. Cominetti, and J. Vaisman. A sharp uniform bound for the distribution of sums of bernoulli trials. *Combinatorics, Probability and Computing*, 25:352–361, 2016.
[2] S. Droste. A rigorous analysis of the compact genetic algorithm for linear functions. *Natural Computing*, 5(3):257–283, 2006.
[3] D. P. Dubhashi and A. Panconesi. *Concentration of measure for the analysis of randomized algorithms.* Cambridge University Press, 2009.
[4] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, 1968.
[5] T. Friedrich, T. Kötzing, M. S. Krejca, and A. M. Sutton. The benefit of recombination in noisy evolutionary search. In *Proc. of ISAAC '15*, pages 140–150. Springer, 2015.
[6] G. R. Harik, F. G. Lobo, and D. E. Goldberg. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4):287–297, 1999.
[7] T. Kötzing. Concentration of first hitting times under additive drift. *Algorithmica*, 75:490–506, 2016.
[8] M. S. Krejca and C. Witt. Lower bounds on the run time of the univariate marginal distribution algorithm on OneMax. In *Proc. of FOGA '17*, pages 65–79. ACM Press, 2017.
[9] P. K. Lehre and P. T. H. Nguyen. Tight bounds on runtime of the univariate marginal distribution algorithm via anti-concentration. In *Proc. of GECCO '17*, pages 1383–1390. ACM Press, 2017.
[10] M. Mitzenmacher and E. Upfal. *Probability and Computing.* Cambridge University Press, 2005.
[11] F. Neumann, D. Sudholt, and C. Witt. A few ants are enough: ACO with iteration-best update. In *Proc. of GECCO '10*, pages 63–70. ACM Press, 2010.
[12] D. Sudholt and C. Witt. Full version of [13] at http://arxiv.org/abs/1607.04063.
[13] D. Sudholt and C. Witt. Update strength in EDAs and ACO: How to avoid genetic drift. In *Proc. of GECCO '16*, pages 61–68. ACM Press, 2016.
[14] C. Witt. Upper bounds on the runtime of the Univariate Marginal Distribution Algorithm on OneMax. In *Proc. of GECCO '17*, pages 1415–1422. ACM Press, 2017.