Combinatorics, Probability and Computing (2018) 27, 643–666. © Cambridge University Press 2018
doi:10.1017/S0963548318000275

Drift Analysis and Evolutionary Algorithms Revisited

J. LENGLER and A. STEGER

Department of Computer Science, ETH Zürich, Zürich, Switzerland
(e-mail: johannes.lengler@inf.ethz.ch, steger@inf.ethz.ch)

Received 10 August 2016; revised 20 April 2018

One of the easiest randomized greedy optimization algorithms is the following evolutionary algorithm which aims at maximizing a function $f : \{0,1\}^n \to \mathbb{R}$. The algorithm starts with a random search point $\xi \in \{0,1\}^n$, and in each round it flips each bit of ξ with probability c/n independently at random, where $c > 0$ is a fixed constant. The thus created offspring ξ' replaces ξ if and only if $f(\xi') \geqslant f(\xi)$. The analysis of the runtime of this simple algorithm for monotone and for linear functions turned out to be highly non-trivial. In this paper we review known results and provide new and self-contained proofs of partly stronger results.

2010 *Mathematics subject classification*: Primary 68W20, 68W40
Secondary 60G40

1. Introduction

Evolutionary algorithms have a long and successful tradition in solving real world optimization problems. Despite much effort the theoretical understanding of these algorithms, however, is still very limited, even for seemingly 'trivial' versions. One of the easiest evolutionary algorithms aims at maximizing a function $f : \{0,1\}^n \to \mathbb{R}$. This algorithm starts with a random search point $\xi \in \{0,1\}^n$, and in each round it flips each bit of ξ with probability (or *mutation rate*) c/n independently, where $c > 0$ is the so-called *mutation parameter*. The thus created *offspring* ξ' replaces ξ if and only if $f(\xi') \geqslant f(\xi)$. Naturally, the interesting question is to determine the *optimization time* of f, that is, the number of rounds until a maximum of f is found (in expectation). Even for the seemingly trivial case that f is a strictly monotone function (*i.e.* $f(\xi) > f(\xi')$ for all ξ, ξ' so that $\xi \neq \xi'$ and $\xi_i \geqslant \xi_i'$ for all $1 \leqslant i \leqslant n$) determining the asymptotic optimization time turned out to be far from trivial. One of the first rigorous results in this direction was by Mühlenbein [15] who determined the optimization time for the case that $f(\xi) = \sum_{i=1}^n \xi_i$. For general linear functions it required substantial effort [4, 5, 6, 9, 10] until Doerr and Goldberg [2] finally showed that the optimization time is $\Theta(n \log n)$ for all constant mutation parameters $c > 0$. For general monotone functions it is easy to see (*cf. e.g.* [3]) that the optimization time is $\Theta(n \log n)$ for all constants $0 < c < 1$. However, as it turned out, this is not necessarily so for

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

larger mutation parameters. Doerr, Jansen, Sudholt, Winzen and Zarges [3] showed that there are monotone functions such that for $c > 16$ the algorithm takes exponential time.

In this paper we provide short and elegant proofs with various drift theorems for the above statements which are partly stronger than previous results. The exact results can be found in Sections 4.1 and 4.2. The paper is completely self-contained.

2. Basic tools

The main tool in the proofs (both in previous proofs and in ours) is drift analysis. Going back to the seminal paper of Hajek [7], drift analysis has been developed as a convenient tool to estimate the expected hitting time of algorithms and random processes (*cf.* the overview by Kötzing [13]). In this section we collect various known statements and provide short and self-contained proofs. Our emphasis is on elegance of proofs rather than obtaining general statements. We always assume the following set-up. We study a Markov chain $(Z_t)_{t\in\mathbb{N}_0}$ over some state space $\mathcal{Z}$, and we assume that we have a real-valued trace $\alpha : \mathcal{Z} \to \mathcal{S} \subseteq \mathbb{R}$, where we sometimes restrict the target set $\mathcal{S}$ to subsets of $\{0\} \cup [1,\infty)$, or simply to $\mathcal{S} = \mathbb{N}_0$. With $T := \inf\{t \in \mathbb{N}_0 : \alpha(Z_t) = 0\}$ we denote the hitting time of 0 (for the trace of Z_t). We are usually interested in determining the expectation $\mathbb{E}[T]$ as a function of the value $n = \alpha(Z_0)$ of the initial state. This situation occurs frequently in optimization, where $Z_t := \xi_t$ denotes the current search point, and $\alpha(\xi_t)$ is some one-dimensional quality measure, for example the number of bits in ξ_t that differ from the global optimum. Note that the one-dimensional random walk $X_t := \alpha(Z_t)$ is in general not a Markov chain, since the one-dimensional value X_t usually does not carry the full information that is contained in Z_t.

Our first theorem is a reformulation of Wald's equation; *cf.* also He and Yao [8].

Theorem 2.1. *Let $(Z_t)_{t\in\mathbb{N}_0}$ be a Markov chain with state space $\mathcal{Z}$ and with a trace function $\alpha : \mathcal{Z} \to \mathcal{S} \subseteq [0,\infty)$, and assume $\alpha(Z_0) = n$. Let $T := \inf\{t \in \mathbb{N}_0 : \alpha(Z_t) = 0\}$. If there exists $c > 0$ such that for all $z \in \mathcal{Z}$ with $\alpha(z) > 0$ and for all $t \geqslant 0$ we have*

$$\mathbb{E}[\alpha(Z_{t+1}) \mid Z_t = z] \leqslant \alpha(z) - c, \tag{2.1}$$

then

$$\mathbb{E}[T] \leqslant \frac{n}{c}.$$

Proof. We denote $X_t := \alpha(Z_t)$. As we are only interested in the first time where X_t reaches zero, we may assume without loss of generality that $X_{T+1} = X_{T+2} = \cdots = 0$.

One easily checks that the condition of the theorem implies that for all $t \in \mathbb{N}_0$

$$\mathbb{E}[X_{t+1} \mid T > t] \leqslant \mathbb{E}[X_t \mid T > t] - c. \tag{2.2}$$

By conditioning on whether $T > t$ or $T \leqslant t$, we obtain

$$\mathbb{E}[X_t] = \mathbb{P}[T > t] \cdot \mathbb{E}[X_t \mid T > t] + \mathbb{P}[T \leqslant t] \cdot 0 \;=\; \mathbb{P}[T > t] \cdot \mathbb{E}[X_t \mid T > t]. \tag{2.3}$$

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

Proceeding similarly for X_{t+1}, we obtain

$$\begin{aligned}\mathbb{E}[X_{t+1}] &= \mathbb{P}[T > t] \cdot \mathbb{E}[X_{t+1} \mid T > t] + \mathbb{P}[T \leqslant t] \cdot 0 \\ &\overset{(2.2)}{\leqslant} \mathbb{P}[T > t] \cdot (\mathbb{E}[X_t \mid T > t] - c) \\ &\overset{(2.3)}{=} \mathbb{E}[X_t] - \mathbb{P}[T > t] \cdot c. \end{aligned} \tag{2.4}$$

Since T is a random variable that takes values in $\mathbb{N}_0$, we may write $\mathbb{E}[T] = \sum_{t=0}^{\infty} \mathbb{P}[T > t]$. Thus

$$c \cdot \mathbb{E}[T] \overset{\tau\to\infty}{\longleftarrow} \sum_{t=0}^{\tau} c\mathbb{P}[T > t] \overset{(2.4)}{\leqslant} \sum_{t=0}^{\tau} (\mathbb{E}[X_t] - \mathbb{E}[X_{t+1}]) = \mathbb{E}[X_0] - \underbrace{\mathbb{E}[X_{\tau+1}]}_{\geqslant 0} \leqslant n, \tag{2.5}$$

which proves the claim of the theorem. □

Remark. Note that in (2.5), there is some subsequence of $(\mathbb{E}[X_{\tau+1}])_{\tau\geqslant 0}$ which converges to $\limsup_{t\to\infty} \mathbb{E}[X_t]$, which implies the slightly stronger statement

$$\mathbb{E}[T] \leqslant \frac{1}{c}(n - \limsup_{t\to\infty} \mathbb{E}[X_t]).$$

This approach also allows for a reverse version of Theorem 2.1: if condition (2.1) is replaced with the reverse condition $\mathbb{E}[X_{t+1} \mid Z_t = z] \geqslant \alpha(z) - c$, then the same proof shows that

$$\mathbb{E}[T] \geqslant \frac{1}{c}(n - \liminf_{t\to\infty} \mathbb{E}[X_t]),$$

that is, either $\mathbb{E}[T] = \infty$, or $\mathbb{E}[T]$ is a finite value which satisfies the inequality.[1]

Note that the assumption (2.1) of Theorem 2.1 cannot hold for stochastic processes for which the traces $\alpha(Z_t)$ come arbitrarily close to zero. The following theorem allows us to resolve this problem, as it shows that we can partition the analysis of such processes into two phases: one coming 'close' to zero, the other actually reaching zero.

Theorem 2.2. *Let $(Z_t)_{t\in\mathbb{N}_0}$ be a time-homogeneous Markov chain with countable state space $\mathcal{Z}$ and with a trace function $\alpha : \mathcal{Z} \to \mathcal{S} \subseteq [0,\infty)$, and assume $\alpha(Z_0) = n$. Furthermore let $C > 0$ be some positive constant and let $T_C := \min\{t \in \mathbb{N}_0 : \alpha(Z_t) \leqslant C\}$. Assume that there exist constants $p_0 > 0$ and $B > 0$ such that for all $z \in \mathcal{Z}$ with $0 < \alpha(z) \leqslant C$ and for all $t \geqslant 0$ we have*

$$\mathbb{P}[\alpha(Z_{t+1}) = 0 \mid Z_t = z] \geqslant p_0$$

and

$$\sum_{z'\in\mathcal{Z}, \alpha(z')>C} \mathbb{P}[Z_{t+1} = z' \mid Z_t = z] \cdot \mathbb{E}[T_C \mid Z_0 = z'] \leqslant B.$$

Then $\mathbb{E}[T] = \mathbb{E}[T_C] + O(1)$.

[1] Note that $\mathbb{E}[X_t]$ does not need to converge to zero, even if $\mathbb{E}[T]$ is finite. For example, consider the Markov chain where Z_{t+1} is either 0 or $2Z_t$, both with probability 1/2, and where $X_t = Z_t$. Here $\mathbb{E}[T] = 2$, but $\mathbb{E}[X_t] = X_0 = n$ for all $t \geqslant 0$. This example also shows that the 'naive' reverse inequality $\mathbb{E}[T] \geqslant n/c$ does not hold in general, since the precondition $\mathbb{E}[X_{t+1} \mid Z_t = z] \geqslant z - c$ is satisfied for all $c > 0$.

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

Proof. Let $\mathcal{Z}_C := \{z \in \mathcal{Z} \mid \alpha(z) \leqslant C\}$. By the definition of T_C, the expectation of the number of steps until we reach a state $z \in \mathcal{Z}_C$ is $\mathbb{E}[T_C]$. So assume now that we are in some non-zero state $z \in \mathcal{Z}_C$. With probability at least p_0 we jump right into zero in the next step. Otherwise we either end up in some (possibly different) state in $\mathcal{Z}_C$ or in some state $z' \in \mathcal{Z} \setminus \mathcal{Z}_C$. In the latter case we need, due to our assumption, in expectation at most B steps to return to a state in $\mathcal{Z}_C$. We conclude that we have in expectation at least every Bth step a chance of jumping to zero with probability at least p_0. As the expectation of a geometric random variable with success probability p_0 is $1/p_0$ this implies that, once we reach a state within $\mathcal{Z}_C$ for the first time, the expectation for the number of additional steps to reach zero is bounded from above by a constant, as claimed. □

Consider a Markov chain on the non-negative integers which is defined as follows. From all states $x > 0$ we move to one of the states $0, 1, \dots, x$ uniformly at random. How long does it take till we reach zero? Intuitively, we expect to halve the distance towards zero in each step, so we expect that it will take $\log_2 n$ steps, if we start in n. What if we move uniformly to a state in $0, 1, \dots, ax$? For which $a > 1$, if at all, do we have time $O(\log n)$ till we reach zero? We will give the (surprising) answer for this in the next section. The clue is to consider a Markov chain in which the distance to zero is rescaled. Although the following theorem is a rather trivial extension of Theorem 2.1, it is sometimes helpful to handle rescaling explicitly.

Theorem 2.3. *Let $(Z_t)_{t \in \mathbb{N}_0}$ be a Markov chain with state space $\mathcal{Z}$ and with a trace function $\alpha : \mathcal{Z} \to \mathcal{S} \subseteq \mathbb{N}_0$, and assume $\alpha(Z_0) = n$. Let $C \in \mathbb{N}_0$ be some positive constant and let $T_C := \inf\{t \in \mathbb{N}_0 : \alpha(Z_t) \leqslant C\}$. Assume furthermore that there exists a constant $c > 0$ and a function $g : \mathcal{S} \to \mathbb{R}$ such that $g(x) = 0$ for all $x \leqslant C$ and $g(x) > 0$ for all $x > C$ and such that for all $t \geqslant 0$*

$$\mathbb{E}[g(\alpha(Z_{t+1})) \mid Z_t = z] \leqslant g(\alpha(z)) - c \quad \text{for all } z \in \mathcal{Z} \text{ with } \alpha(z) > C.$$

Then $\mathbb{E}[T_C] \leqslant g(n)/c$.

Proof. The statement follows immediately from Theorem 2.1, applied to the trace $\alpha'(z) := g(\alpha(z))$. □

The next theorem is a simplified version of the previous one that is often easier to apply. The present formulation is from Mitavskiy, Rowe and Cannings [14]; the proof follows Theorem 4.6 of [12].

Theorem 2.4. *Let $(Z_t)_{t \in \mathbb{N}_0}$ be a Markov chain with state space $\mathcal{Z}$ and with a trace function $\alpha : \mathcal{Z} \to \mathcal{S} \subseteq \{0\} \cup [1, \infty)$, and assume $\alpha(Z_0) = n$. Let $T := \inf\{t \in \mathbb{N}_0 : \alpha(Z_t) = 0\}$. Suppose furthermore that there is a positive, increasing function $h : [1, \infty) \to \mathbb{R}_{>0}$ such that for all $z \in \mathcal{Z}$ with $\alpha(z) > 0$ and all $t \geqslant 0$ we have*

$$\mathbb{E}[\alpha(Z_{t+1}) \mid Z_t = z] \leqslant \alpha(z) - h(\alpha(z)).$$

Then

$$\mathbb{E}[T] \leqslant \frac{1}{h(1)} + \int_1^n \frac{1}{h(u)} du.$$

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

Proof. Let

$$g(x) := \frac{1}{h(1)} + \int_1^x \frac{1}{h(u)}du \quad \text{for } x \in \mathcal{S}, x > 0 \text{ and } g(0) := 0.$$

We claim that

$$g(x) - g(y) \geqslant \frac{x-y}{h(x)} \quad \text{for all } x, y \in \mathcal{S}$$

such that $x > 0$. To see this, observe that since h is increasing and positive,

$$g(x) - g(y) = \begin{cases} \displaystyle\int_y^x \frac{1}{h(u)}du \geqslant (x-y) \cdot \frac{1}{h(x)} & \text{if } x > y > 0, \\ \displaystyle\frac{1}{h(1)} + \int_1^x \frac{1}{h(u)}du \geqslant \frac{x}{h(x)} & \text{if } x > y = 0, \\ 0 & \text{if } x = y > 0, \\ \displaystyle -\int_x^y \frac{1}{h(u)}du \geqslant -(y-x) \cdot \frac{1}{h(x)} & \text{if } 0 < x < y. \end{cases}$$

The above inequality for g implies that for all $z \in \mathcal{Z}$ with $\alpha(z) > 0$ we have

$$\mathbb{E}[g(\alpha(Z_t)) - g(\alpha(Z_{t+1})) \mid Z_t = z] \geqslant \mathbb{E}\left[\frac{\alpha(Z_t) - \alpha(Z_{t+1})}{h(\alpha(Z_t))} \mid Z_t = z\right] \geqslant 1,$$

where the last inequality is just a reformulation of the assumption of the theorem.

Hence, we have

$$\mathbb{E}[g(\alpha(Z_{t+1})) \mid Z_t = z] \leqslant g(\alpha(z)) - 1,$$

and the claim of the theorem thus follows by applying Theorem 2.1 to the trace $\alpha'(z) := g(\alpha(z))$. □

A special case of Theorem 2.4 is the case $h(x) = \delta x$, which is called *multiplicative drift*. We then get $\mathbb{E}[T] \leqslant \delta^{-1}(1 + \log n)$; see also [4]. In fact, a straightforward application of Markov's inequality even supplies us with exponential tail bounds, as was first noticed by Doerr and Goldberg [2].

Theorem 2.5. *Let $(Z_t)_{t \in \mathbb{N}_0}$ be a Markov chain with state space $\mathcal{Z}$ and with a trace function $\alpha : \mathcal{Z} \to \mathcal{S} \subseteq \{0\} \cup [1, \infty)$, and assume $\alpha(Z_0) = n$. Let $T := \inf\{t \in \mathbb{N}_0 : \alpha(Z_t) = 0\}$. Assume that there is $\delta > 0$ such that for all $z \in \mathcal{Z}$ with $\alpha(z) > 0$ and for all $t \geqslant 0$ we have*

$$\mathbb{E}[\alpha(Z_{t+1}) \mid Z_t = z] \leqslant (1 - \delta)\alpha(z).$$

Then

$$\mathbb{P}\left[T > \left\lceil \frac{\log n + k}{|\log(1-\delta)|} \right\rceil\right] \leqslant e^{-k}.$$

Proof. Let

$$t := \lceil (\log n + k)/|\log(1-\delta)| \rceil.$$

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

Since

$$\mathbb{E}[\alpha(Z_t)] \leqslant (1-\delta)^t \alpha(Z_0) \leqslant e^{-k},$$

the claim follows from Markov's inequality:

$$\mathbb{P}[T > t] = \mathbb{P}[\alpha(Z_t) \geqslant 1] \leqslant e^{-k}.$$

Note that here we used $\mathcal{S} \subseteq \{0\} \cup [1,\infty)$; replacing the (somewhat arbitrary) boundary 1 by some other constant would give a multiplicative constant in front of the probability e^{-k}. □

An important implication of Theorem 2.5 is that if the probability of making large jumps is geometrically bounded, then an additive drift also suffices to get exponential tail bounds. Moreover, it is exponentially unlikely to cross an area with *negative* drift in less than exponentially many steps. This idea again goes back to Hajek [7] and was first proved in this simplified form by Oliveto and Witt [16, 17]. However, that proof is still based on some results from Hajek's paper. Here we give a short and fully self-contained proof.

Theorem 2.6. *For all $a, b, \delta, \varepsilon, \gamma, r > 0$, with $a < b$, there is $c > 0$, $n_0 \in \mathbb{N}$ such that the following holds. Let $(Z_t)_{t\in\mathbb{N}_0}$ be a Markov chain with state space $\mathcal{Z}$ and with a trace function $\alpha : \mathcal{Z} \to \mathcal{S} \subseteq \mathbb{R}$, and let $n \geqslant n_0$ such that for all $t \geqslant 0$ the following conditions hold:*

(1) $\mathbb{E}[\alpha(Z_{t+1}) - \alpha(Z_t) \mid Z_t = z] \leqslant -\varepsilon$ *for all $z \in \mathcal{Z}$ with $\alpha(z) > an$,*

(2) $\mathbb{P}[|\alpha(Z_{t+1}) - \alpha(Z_t)| \geqslant j \mid Z_t = z] \leqslant r(1+\delta)^{-j}$ *for all $j \in \mathbb{N}_0$ and all $z \in \mathcal{Z}$.*

Let $T_a := \min\{t \geqslant 0 : \alpha(Z_t) \leqslant an\}$ and $T_b := \min\{t \geqslant 0 : \alpha(Z_t) \geqslant bn\}$.

(a) *If $\alpha(Z_0) = bn$ then*

$$\mathbb{P}\left[T_a \geqslant \frac{(1+\gamma)(b-a)n}{\varepsilon}\right] \leqslant e^{-cn}.$$

(b) *If $\alpha(Z_0) = an$ then $\mathbb{P}[T_b \leqslant e^{cn}] \leqslant e^{-cn}$.*

Proof. We may assume $\gamma < 1/2$. Let $0 < \eta < 1$ be so small that $e^\eta < (1+\delta)^{1/2}$ holds. We claim that if η is sufficiently small, then we can choose $j_0 > 0$ such that $e^{\eta j} = 1 + \eta j \pm \gamma\eta\varepsilon/6$ is satisfied for all $-j_0 \leqslant j \leqslant j_0$ (where we use $x = y \pm \varepsilon$ as shorthand for $x \in [y-\varepsilon, y+\varepsilon]$), and such that at the same time

$$\sum_{j=j_0+1}^{\infty} r(1+\delta)^{-j/2} < \gamma\eta\varepsilon/24 \quad \text{and} \quad \sum_{j=j_0+1}^{\infty} jr(1+\delta)^{-j} < \gamma\eta\varepsilon/24 \tag{2.6}$$

holds. Indeed, by setting (rather arbitrarily) $j_0 = \eta^{-1/3}$ we satisfy all three conditions if $\eta = \eta(\delta, \varepsilon, \gamma, r)$ is small enough.

Consider $X_t := \alpha(Z_t)$ and $Y_t := e^{\eta X_t}$. For any $z \in \mathcal{Z}$ with $x := \alpha(z) > an$, we have

$$\begin{aligned}
&\mathbb{E}[Y_{t+1} - Y_t \mid Z_t = z] = e^{\eta x}\mathbb{E}[e^{\eta(X_{t+1}-x)} - 1 \mid Z_t = z] \\
&= e^{\eta x}\Bigg(\sum_{j=-j_0}^{j_0} \mathbb{P}[X_{t+1} - X_t = j \mid Z_t = z] \underbrace{(e^{\eta j} - 1)}_{=\eta j \pm \gamma\eta\varepsilon/6} \pm 2 \cdot \underbrace{\sum_{j=j_0+1}^{\infty} r(1+\delta)^{-j} \underbrace{(e^{\eta j} + 1)}_{\leqslant (1+\delta)^{j/2}+j}}_{\leqslant \gamma\eta\varepsilon/12 \text{ by (2.6)}}\Bigg)
\end{aligned}$$

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

$$= e^{\eta x}\left(\sum_{j=-j_0}^{j_0} \mathbb{P}[X_{t+1} - X_t = j \mid Z_t = z]\eta j \pm \gamma\eta\varepsilon/3\right)$$
$$\overset{\text{cond. (2) and (2.6)}}{=} e^{\eta x}(\mathbb{E}[X_{t+1} - X_t \mid Z_t = z]\eta \pm \gamma\eta\varepsilon/2) \overset{\text{cond. (1)}}{<} -e^{\eta x}\underbrace{\eta\varepsilon(1-\gamma/2)}_{=:\nu}.$$

Recalling that $Y_t = e^{\eta x}$ if $\alpha(Z_t) = x$, we thus find that Y_t has a multiplicative drift with factor $1-\nu$. For (a), we now apply Theorem 2.5 to $\alpha'(Z_t)$ defined by $\alpha'(Z_t) := e^{-\eta an} \cdot Y_t$ if $\alpha(Z_t) \geqslant an$, and $\alpha'(Z_t) := 0$ otherwise. Then $\alpha'(Z_t)$ also has multiplicative drift with factor $1-\nu$, and it has state space within $\{0\} \cup [1,\infty)$. We also choose $k := \gamma\eta(b-a)n/16$. To apply Theorem 2.5, we use that $|\log(1-\nu)| \geqslant \eta\varepsilon(1-5\gamma/8)$ if η (and thus ν) is sufficiently small. Then

$$(\log(\alpha'(Z_0)) + k)/|\log(1-\nu)| \leqslant (1+\gamma)(b-a)n/\varepsilon,$$

and thus $\mathbb{P}[T > \lceil(1+\gamma)(b-a)n/\varepsilon\rceil] \leqslant e^{-k}$ by Theorem 2.5, from which (a) follows.

For (b), consider the event $\mathcal{E}$ that for the first e^{cn} points in time where $X_t \leqslant an$ we have $X_{t+1} \leqslant X_t + (b-a)n/2$. Note that $\mathbb{P}[\neg\mathcal{E}] \leqslant e^{cn} \cdot r(1-\delta)^{n(b-a)/2} = e^{-\Omega(n)}$ if we choose $c > 0$ sufficiently small. From now on we may thus assume $\mathcal{E}$. Consider some time t_1 at which we cross the lower bound from below, that is, $X_{t_1-1} \leqslant an$ and $X_{t_1} > an$. Let t_2 be the next point in time where we drop back below an, so $t_2 = \min\{t > t_1 \mid X_{t_2} \leqslant an\}$. Note that for all such $t_1 < e^{cn}$ we have $X_{t_1} < n(a+b)/2$, unless the event $\mathcal{E}$ fails. We now show that it is very unlikely that we reach bn between t_1 and $\min\{t_2, e^{cn}\}$, that is, $Y_t \geqslant e^{\eta bn}$. Using the multiplicative drift of Y_t and Markov's inequality, for all $t_1 < t < \min\{t_2, e^{cn}\}$ and all $z \in Z$ with $x := \alpha(z) \in [an, (a+b)n/2]$,

$$\mathbb{P}[Y_t > e^{\eta bn} \mid Z_{t_1} = z] \leqslant \frac{\mathbb{E}[Y_t \mid Z_{t_1} = z]}{e^{\eta bn}} \leqslant \frac{(1-\nu)^{t-t_1}e^{\eta x}}{e^{\eta bn}} \leqslant (1-\nu)^{t-t_1}e^{-\eta(b-a)n/2}.$$

A union bound over all $t_1 < t < \min\{t_2, e^{cn}\}$ shows that the probability that bn is reached in the interval $[t_1, t_2]$ is $e^{-\Omega(n)}$. A second union bound over all $t_1 < e^{cn}$ thus shows the second claim for c sufficiently small. □

3. Example: random decline

With Theorems 2.4 and 2.5 to hand we can now analyse the Markov chain that, from state x, moves to a state uniformly at random within $\{0, \dots, \lfloor ax \rfloor\}$. Clearly, $\mathbb{E}[X_{t+1} \mid X_t = x] \leqslant (a/2)x$ and we can apply Theorem 2.5 to deduce that the number of steps T until we reach zero is $O(\log n)$ for all $a < 2$. For $a > 2$ it may seem that $\mathbb{E}[T] = \infty$, as we have a drift to the right. Perhaps surprisingly, this is not true. In fact, we have the following theorem, which we could not find elsewhere in the literature.

Theorem 3.1. *For the Markov chain defined above with (constant) parameter a, we have* $\mathbb{E}[T] = O(\log n)$ *if and only if* $a < e$, *where* $e = 2.718\dots$ *is the Euler constant.*

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

Proof. Assume $a < e$. Let $g(x) = \log(x)$ for $x > C$ (where C is some constant depending on a that we will fix below) and $g(x) = 0$ for all $x \leqslant C$. Then, for all $x > C$,

$$\sum_{m>C} \mathbb{P}[X_{t+1} = m \mid X_t = x] \cdot g(m) = \frac{1}{\lfloor ax \rfloor + 1} \sum_{m=C+1}^{\lfloor ax \rfloor} \log(m)$$
$$\leqslant \log(x) - 1 + \log(a) + \frac{1}{\lfloor aC \rfloor + 1}.$$

As $\log(a) < 1$ for all $a < e$ we can thus find for each $a < e$ a constant C so that we can apply Theorem 2.3 (with α the identity function) to deduce that $\mathbb{E}[T_C] \leqslant O(\log n)$. Theorem 2.2 then implies that we also have $\mathbb{E}[T] = O(\log n)$. Indeed, the assumptions of Theorem 2.2 are easily seen to be fulfilled for $p_0 = 1/(C+1)$ and $B = \max_{C+1 \leqslant n \leqslant aC} \mathbb{E}[T_C \mid X_0 = n]$ (which is finite, as we maximize over a finite number of expectations and we have already shown that these expectations are finite).

For the other direction, we restrict ourselves for concreteness to $a = e$. (The case $a > e$ stochastically dominates the case $a = e$, and thus the statement for $a > e$ follows easily by a coupling argument.) For the sake of contradiction we assume that $\mathbb{E}[T] = O(\log n)$, so assume that there is $D > 0$ such that $\mathbb{E}[T] \leqslant D \log n$ for all n. This implies in particular that $\mathbb{E}[T]$ is finite. Note that we may assume $D \geqslant 2$, and that D is an integer. Let $g(x) = 2D \log(x/D^2)$ for $x > D^2$, $g(x) = 1$ for $1 \leqslant x \leqslant D^2$, and $g(x) = 0$ for $x = 0$. We set $Y_t := g(X_t)$, where X_t is the state in the tth step. Note that $X_t = 0$ if and only if $Y_t = 0$. We claim that the reverse of condition (2.1) in Theorem 2.1 is satisfied for $\alpha(X_t) := Y_t$ for $c = 1$. To see this consider some $y > 0$ of the form $g(x)$, where $x \in \mathbb{N}^+$. We want to show that $\mathbb{E}[Y_{t+1} \mid X_t = x] \geqslant Y_t - 1$. For $x \leqslant D^2$ this is trivial since then $y = g(x) \leqslant 1$. So assume that $x > D^2$. We compute as above (using that $s \cdot \log(s/(eD^2))$ is an antiderivative of $\log(s/D^2)$ and that $\lfloor ex \rfloor \geqslant ex - 1$ implies $\log(\lfloor ex \rfloor/(eD^2)) \geqslant \log(x/D^2) - 2/(ex)$) that

$$\mathbb{E}[Y_{t+1} \mid X_t = x] = \mathbb{E}[g(X_{t+1}) \mid X_t = x] \geqslant \frac{1}{\lfloor ex \rfloor + 1} \sum_{m=D^2}^{\lfloor ex \rfloor} (2D \log(m/D^2))$$
$$\geqslant \frac{\lfloor ex \rfloor}{\lfloor ex \rfloor + 1} \left(2D \log\left(\frac{\lfloor ex \rfloor}{eD^2} \right) + \frac{2D^3}{\lfloor ex \rfloor} \right) \geqslant 2D \log\left(\frac{x}{D^2} \right) - 1 \;=\; y - 1.$$

Since this inequality is true for all $y = g(x) > 0$, $x \in \mathbb{N}^+$, we may use the reverse version of Theorem 2.1; *cf.* the remark after the proof of Theorem 2.1. Hence, for $n > D^2$,

$$\mathbb{E}[T] \geqslant Y_0 - \liminf_{t \to \infty} \mathbb{E}[Y_t] = 2D \log(n/D^2) - \liminf_{t \to \infty} \mathbb{E}[Y_t].$$

Since we assumed $\mathbb{E}[T] \leqslant D \log n$, the last inequality implies for all sufficiently large n and t that

$$\mathbb{E}[Y_t] \geqslant D \log(n)/2.$$

Let $p_t := \mathbb{P}[Y_t > 0] = \mathbb{P}[T > t]$, and let $E_t := \mathbb{E}[Y_t \mid Y_t > 0]$. Then $p_t E_t = \mathbb{E}[Y_t] \geqslant D \log(n)/2$ for sufficiently large n and t. On the other hand, since X_t can increase in each step by at most a factor of e, we can bound $X_t \leqslant ne^t$, and thus $E_t \leqslant 2D \log(ne^t/D^2)$. Combining the two bounds, we obtain

$$p_t \geqslant \frac{D \log n}{2E_t} \geqslant \frac{D \log n}{4D \log(ne^t/D^2)} = \frac{D \log n}{4Dt + 4D \log(n/D^2)}.$$

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

However, this implies that

$$\mathbb{E}[T] = \sum_{t \geqslant 0} p_t = \infty,$$

contradicting that $\mathbb{E}[T]$ is finite. Thus the assumption $\mathbb{E}[T] = O(\log n)$ for $a = e$ was false. □

4. Evolutionary algorithm

In this section we (re)prove some fundamental properties of the following classical algorithm [1] for optimizing a *fitness function* $f : \{0,1\}^n \to \mathbb{R}$, which we call the *evolutionary algorithm* or *EA*.[2] The algorithm starts with a search point $\xi^0 \in \{0,1\}^n$ that we usually assume to be chosen uniformly at random. It then proceeds in rounds. In each round it flips each bit of the current search point ξ with probability (or *mutation rate*) c/n, where $c > 0$ is the *mutation parameter*. The thus created *offspring* ξ' replaces ξ if and only if $f(\xi') \geqslant f(\xi)$. We are interested in the *optimization time* of f, that is, the number of rounds until a (global) maximum of f is found. Note that we may assume that the algorithm has access to the function f only via an oracle. That is, the algorithm can query the function value $f(\xi)$ for a given ξ, but has no further knowledge about the function. In particular, the algorithm thus has no chance of 'knowing' which x maximizes the function before having queried all 2^n values. This is the reason for the above definition of the optimization time: we count the number of rounds until the algorithm first queries a global maximum. From then on the algorithm may continue asking queries, but clearly, the function value will never change again.

Throughout this section, we will use the following notation. For all $t \geqslant 0$, we let $\xi^t \in \{0,1\}^n$ denote the search point after t rounds of the algorithm. For any non-empty set of indices $I \subseteq [n]$, let $d(I,t) := |\{i \in I \mid \xi_i^t = 0\}|/|I|$ be the density of zero-bits in the I-substring of ξ^t. We write $d(I)$ instead of $d(I,t)$ if t is clear from the context.

By the *no free lunch* principle, for arbitrary functions f there is no hope that the algorithm will be fast. For example, for the function $f(\xi)$ that is one for $\xi = x_0$ and zero otherwise, *any* algorithm will have exponential optimization time w.h.p.[3] Nevertheless algorithm EA gives good results in many practical applications, indicating the algorithm should be faster for a suitably restricted class of functions. In this section, we will study two classes of functions: *strictly monotone functions*, that is, functions f that satisfy $f(\xi) > f(\xi')$ for all ξ, ξ' so that $\xi \neq \xi'$ and $\xi_i \geqslant \xi_i'$ for all $1 \leqslant i \leqslant n$; and *linear functions*, that is, functions of the form $f(\xi) = \sum_{i=1}^n a_i \xi_i$ with weights $a_1, \ldots, a_n \in \mathbb{R}$. To avoid trivialities, we will always assume that all a_i are non-zero. Moreover, by symmetry of the EA we may even assume that the weights are sorted and positive, $a_1 \geqslant \cdots \geqslant a_n > 0$. Note that with these assumptions every linear function is in particular strictly monotone.

We will frequently use the drift theorems from Section 2, where we always set $Z_t = \xi^t$, and where we use the notation $X_t := \alpha(Z_t)$ for brevity, for various traces α that we specify in the corresponding sections. For strictly monotone functions it easily follows from the drift theorems

[2] In the evolutionary algorithms community it is called the (1+1)-evolutionary algorithm.
[3] *With high probability*, that is, with probability tending to one as $n \to \infty$.

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

that the optimization time of algorithm EA is $\Theta(n\log n)$ for all mutation parameters $c < 1$; see also [3].

Theorem 4.1. *For every constant $c < 1$ and all strictly monotone functions f, the optimization time of algorithm EA with mutation rate c/n is $\Theta(n\log n)$ in expectation and w.h.p.*

Proof. For the lower bound, simply observe that w.h.p. the initial string has at least $n/3$ zero-bits, and that each of these bits needs to be flipped at least once to reach the optimum. Therefore, we need to flip at least $\Omega(n\log n)$ bits in expectation and w.h.p. (as this is essentially a coupon collector process). Thus we also need $\Omega(n\log n)$ rounds to achieve so many flips. We omit the details.

For the upper bound, let $X_t := |\{i \in [n] : \xi_i^t = 0\}|$ be the number of zero-bits in the tth step of the algorithm. Fix $\xi \in \{0,1\}^n$, and assume that $\xi^t = \xi$ and $X_t = x$. We want to bound $\mathbb{E}[X_{t+1} \mid \xi^t = \xi]$. Assume we flip r zero-bits to a one and s one-bits to a zero. For $r = s = 0$ we have $X_{t+1} = X_t$. If $r = 0$ and $s > 0$, then we reject the offspring ξ' and thus also $X_{t+1} = X_t$. If $r > 0$ and $s = 0$ then we accept ξ' and thus $X_{t+1} \leqslant X_t - 1$, with room to spare. For the remaining case $r > 0$ and $s \geqslant 1$ we use the bound $X_{t+1} \leqslant X_t + s - 1$, which is true regardless of whether we accept or reject ξ'.

Let p_0 denote the probability that $r = 0$. Then the above observations imply

$$\mathbb{E}[X_{t+1} \mid \xi^t = \xi] \leqslant p_0 x + (1-p_0)(x - 1 + \mathbb{E}[s]) = x - (1-p_0)(1 - \mathbb{E}[s]).$$

Note that p_0 is the probability that a binomially distributed random variable with parameters x and $p = c/n$ is equal to zero. Here the following inequality about the binomial distribution is helpful:

$$\mathbb{P}[\mathrm{Bin}(n,p) > 0] \geqslant \frac{np}{1+np} \quad \text{for all } n \in \mathbb{N}_0 \text{ and } 0 \leqslant p \leqslant 1 \tag{4.1}$$

(easy to check by induction on n). With (4.1) and the observation that $\mathbb{E}[s] = (n-x)\cdot(c/n) \leqslant c$ we deduce

$$\mathbb{E}[X_{t+1} \mid \xi^t = \xi] \leqslant x - \frac{xc/n}{1+xc/n}(1-c) \leqslant x - \frac{xc(1-c)}{n(1+c)}$$

and we can apply Theorem 2.4 with $h(x) = xc(1-c)/(n(1+c))$ in order to deduce that $\mathbb{E}[T] = O(n\log n)$. Moreover, by Theorem 2.5 (with $\delta = c(1-c)/(n(1+c))$) the same bound holds w.h.p. □

We will see in the next section that the restriction $c < 1$ is essential: for $c \geqslant 2.2$ the running time is not $O(n\log n)$, but actually exponential in n. Before we proceed to that proof we state a lemma on linear functions that will turn out to be very useful in Section 4.1 as well as Section 4.2.

Lemma 4.2. *Consider the algorithm EA on a linear function, $f(x) = \sum_{i=1}^n a_i x_i$, and assume that $a_i \geqslant a_j \geqslant 0$ for $i \leqslant j$. Let $\alpha, \beta, \varepsilon > 0$ be any constants and let $I, J \subseteq [n]$ be two sets of indices such that $\max I < \min J$ and such that $|I| = \alpha n$ and $|J| = \beta n$. Then there exists $C > 0$ such that for any initial value ξ^0 of the algorithm, w.h.p. we have $d(I,t) \leqslant d(J,t) + \varepsilon$ for all $t \in [Cn, e^{n/C}]$. The interval may be replaced with $[0, e^{n/C}]$ if $d(I,0) \leqslant d(J,0) + \varepsilon/2$.*

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

Proof. We use a coupling that was first applied in [10]. We first consider a single step, so fix $t \geqslant 0$ and ξ^t. Let $i < j$ be two positions such that $\xi_i^t = \xi_j^t = 0$. We claim that $\mathbb{P}[\xi_i^{t+1} = 1] \geqslant \mathbb{P}[\xi_j^{t+1} = 1]$. To see this assume first that all bit flips except the ith and the jth are fixed. Then the following holds. If the bit flips for the ith and the jth bit are identical then we change either none of the two bits or both. Thus, the only interesting case is if exactly one of these two bits is flipped. Here we observe the following: if the case that the jth bit is flipped is accepted then so is the case that the ith bit is flipped (due to the fact that we assumed $a_1 \geqslant a_2 \geqslant \cdots \geqslant a_n$). A few moments of thought show that this already proves our claim.

Similarly we deduce that if $k < \ell$ are two positions such that $\xi_k^t = \xi_\ell^t = 1$, then we have $\mathbb{P}[\xi_k^{t+1} = 0] \leqslant \mathbb{P}[\xi_\ell^{t+1} = 0]$.

Now we can prove the lemma. We claim that by Theorem 2.6 it suffices to show that $n \cdot (d(I) - d(J))$ has a negative drift, whenever $d(I) \geqslant d(J) + \varepsilon/2$. Indeed, then we may first apply Theorem 2.6(a) to $X_t := n \cdot (d(I,t) - d(J,t))$ with $a := \varepsilon/2$ and $b := d(I,0) - d(J,0)$ to conclude that after a linear number of steps, w.h.p. $X_t \leqslant \varepsilon n/2$. Afterwards, we may apply Theorem 2.6(b) again with the same X_t and a, but with $b := \varepsilon$ to conclude that w.h.p. $X_t \leqslant \varepsilon n$ for an exponential number of steps.

So let us show that $n \cdot (d(I) - d(J))$ has a negative drift for $d(I) \geqslant d(J) + \varepsilon/2$. So assume that $d(I,t) \geqslant d(J,t) + \varepsilon/2$ holds for some $t \geqslant 0$. For fixed ξ^t, let

$$p_{I0} := \min\{\mathbb{P}[\xi_i^{t+1} = 1 \mid \xi^t] : i \in I, \xi_i^t = 0\} \quad \text{and} \quad p_{J0} := \max\{\mathbb{P}[\xi_j^{t+1} = 1 \mid \xi^t] : j \in J, \xi_j^t = 0\},$$

and observe that we know from above that $p_{I0} \geqslant p_{J0}$. Similarly, we have for

$$p_{I1} := \max\{\mathbb{P}[\xi_k^{t+1} = 0 \mid \xi^t] : k \in I, \xi_k^t = 1\} \quad \text{and} \quad p_{J1} := \min\{\mathbb{P}[\xi_\ell^{t+1} = 0 \mid \xi^t] : \ell \in J, \xi_\ell^t = 1\}$$

that $p_{I1} \leqslant p_{J1}$. Moreover, observe that $p_{J0} = \Omega(1/n)$, because the probability of flipping the ℓth bit and no other bit is $\Omega(1/n)$. Thus, the the expected change of $n \cdot (d(I,t) - d(J,t))$ is at most

$$\begin{aligned} &n[(1 - d(I,t))p_{I1} - d(I,t)p_{I0}] - n[[(1 - d(J,t))p_{J1} - d(J,t)p_{J0}] \\ &\qquad \leqslant \underbrace{n(d(J,t) - d(I,t))}_{\leqslant -\varepsilon/2}(p_{I1} + p_{J0}), \end{aligned}$$

which is $-\Omega(1)$, as desired. □

4.1. Monotone functions

We already saw that for $c < 1$ the runtime of algorithm EA is $\Theta(n \log n)$ for all strictly monotone functions. For the natural choice $c = 1$ it is still unknown whether the runtime is $\Theta(n \log n)$. The best known upper bound is $O(n^{3/2})$ due to Jansen [11]. The following theorem explains why this may be not so easy to show, as we do get an exponential optimization time for cs that are not much larger.

Theorem 4.3. *For every constant $c \geqslant 2.2$, there is a strictly monotone function such that w.h.p. the optimization time of the EA with mutation rate c/n is $e^{\Omega(n)}$.*

A similar theorem was shown by Doerr, Jansen, Sudholt, Winzen and Zarges [3] with $c > 16$ instead of $c \geqslant 2.2$.

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

Proof. The main idea of our proof is the following. Consider n bits and split them into high-order bits (the first αn, say) and low-order bits (the remaining $(1-\alpha)n$). Now consider the (linear) fitness function $a_i = n$ for $1 \leqslant i \leqslant \alpha n$ and $a_i = 1$ otherwise. Then the density of zero-bits will decrease faster among the high-order bits. In particular, there will be a time when the density among the high-order bits is ε, but the density among the low-order bits is still $\varepsilon + \delta$, for some $\delta = \delta(\alpha, \varepsilon, c) > 0$. The overall density in the string is then $\varepsilon + (1-\alpha)\delta > \varepsilon$.

If we now change the fitness function by picking a random set of αn bits as 'new' high-order bits, we will start with a density of $\varepsilon + (1-\alpha)\delta$ among the high-order bits and will decrease it during the second round to density ε. Whenever we decrease the number of zero-bits among the high-order bits, we accept the offspring regardless of the changes of low-order bits. In particular, in such rounds the density among the low-order bits will *increase* in expectation towards $1/2$. By choosing the constants appropriately, this yields overall a *positive* drift for the density among the low-order bits. The drift is actually strong enough that the density will increase from $\varepsilon + (1-\alpha)\delta$ to $\varepsilon + \delta$ in a very short time; this will finish before the density among high-order bits can decrease from $\varepsilon + (1-\alpha)\delta$ to ε. So after the second round we are again in the situation that the density among high-order bits is ε and the density among low-order bits is (at least) $\varepsilon + \delta$. We will show that we can play this game for exponentially many rounds.

Formally we proceed as follows. For a given $c \geqslant 2.2$ we choose $0 < \alpha < 1/2$, which satisfies

$$\alpha c - e^{-(1-\alpha)c} > \frac{\alpha}{1-\alpha}. \tag{4.2}$$

A numerical calculation shows that this is possible for all $c \geqslant 2.2$. We choose a constant $\beta = \beta(\alpha, c)$ which is sufficiently small compared to α (by abuse of notation we denote this by $\alpha \gg \beta$). To provide an overview, the constants in the proof will satisfy

$$1/2 > \alpha \gg \varepsilon \gg \delta \gg \beta \gg \gamma \gg \mu > 0, \tag{4.3}$$

where γ and μ are constants that will appear later in the proof. The $\gg$-sign hereby indicates the only restriction that we require: the constants are chosen from left to right, so that α satisfies equation (4.2) and then every constant is suitably small compared to all previously chosen constants.

For $1 \leqslant i \leqslant e^{\mu n} + 1$ we choose sets $A_i \subseteq [n]$ of size αn independently and uniformly at random, and we choose sets $B_i \subseteq A_i$ uniformly at random of size βn. We define the *level* $\ell(\xi)$ of a vector $\xi \in \{0,1\}^n$ by

$$\ell(\xi) := \max\{\ell' \in [e^{\mu n}] : |\{j \in B_{\ell'} : \xi_j = 0\}| \leqslant \varepsilon |B_{\ell'}|\}, \tag{4.4}$$

where we set $\ell(\xi) = 0$ if no such ℓ' exists.

With this notation to hand we can then define a fitness function $f : \xi \in \{0,1\}^n \to \mathbb{R}$ as follows:

$$f(\xi) = \ell(\xi) \cdot n^2 + \sum_{i \in A_{\ell(\xi)+1}} \xi_i \cdot n + \sum_{i \notin A_{\ell(\xi)+1}} \xi_i. \tag{4.5}$$

So the set A_{i+1} describes the set of high-order bits at level i, where the level is determined by the sets B_i. One easily checks that this function is monotone. Indeed, assume that ξ is dominated by ξ', that is, $\xi_i \leqslant \xi'_i$ for all $1 \leqslant i \leqslant n$. Then $\ell(\xi) \leqslant \ell(\xi')$. If $\ell(\xi) = \ell(\xi')$ then it is obvious that $f(\xi) \leqslant f(\xi')$ (and we have strict inequality if $\xi \neq \xi'$). On the other hand, if $\ell(\xi) < \ell(\xi')$ then $f(\xi) < \ell(\xi)n^2 + n^2 \leqslant \ell(\xi')n^2 \leqslant f(\xi')$, as desired.

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

Observe also that, by definition of the function f, the algorithm will never accept a change that will decrease the *current level* $\ell(t) := \ell(\xi^t)$. That is, when the current level is i the density of zero-bits will stay below ε within B_i (or the level increases).

Note that the definition of ℓ refers to *all* sets B_i, which makes the analysis quite tricky. In order to avoid this we define a slightly different level function $\tilde{\ell} = \tilde{\ell}(t)$ that depends on the time t. More precisely, recall that ξ^t denotes the search point after round t. In order to compute $\tilde{\ell}(t)$ we proceed as follows: instead of applying (4.4) to the search point ξ^t, which would consider the maximum over all integers $\ell' \in [e^{\mu n}]$, we only consider values $\ell' \in [\tilde{\ell}(t-1)+1]$. Informally, $\tilde{\ell}$ thus coincides with ℓ except that we only accept level gains of exactly one. We note that $\tilde{\ell}(t)$ depends not only on the current search point but on the whole history of the algorithm. We now consider algorithm EA on the time-dependent fitness function

$$\tilde{f}(t) = \tilde{\ell}(t) \cdot n^2 + \sum_{i \in A_{\tilde{\ell}(t)+1}} \xi_i^t \cdot n + \sum_{i \notin A_{\tilde{\ell}(t)+1}} \xi_i^t,$$

that is, if ξ' is the offspring of ξ^t then the algorithm either chooses $\xi^{t+1} := \xi^t$ or $\xi^{t+1} := \xi'$, picking the option that maximizes $\tilde{f}(t+1)$. As long as $\tilde{\ell}(t)$ coincides with $\ell(t)$, we have $\tilde{f}(t) = f(\xi^t)$. The algorithm thus performs the same steps on both functions as long as $\tilde{\ell}(t) = \ell(t)$. We will show later that w.h.p. $\tilde{\ell}(t)$ and $\ell(t)$ indeed coincide for an exponential number of rounds. The advantage of $\tilde{f}$ is that we do not have to choose the sets A_i, B_i in advance: we can postpone choosing A_{i+1}, B_{i+1} until we enter the ith level.

Before we now dive into the proof we fix some notation. Let $T := e^{\mu n}$. Our aim is to show that w.h.p. the algorithm will run for at least T rounds. With t_i we denote the points in time where $\tilde{\ell}$ increases from $i-1$ to i, that is, $\tilde{\ell}(t_i - 1) = i-1$ and $\tilde{\ell}(t_i) = i$. Moreover, for convenience we set $t_0 := 0$ and $A_0 := B_0 := B_{-1} = \emptyset$. Recall that for a set $S \subseteq [n]$ we let $d(S,t)$ denote the density of zero-bits in the set S with respect to the tth search point ξ^t.

Observe that algorithm EA (with respect to the function $\tilde{f}$) does not need to know A_{i+1} before time t_i. By Chernoff bounds (and a union bound over all $1 \leqslant t \leqslant t_i \leqslant T$) we can thus assume (by making μ sufficiently small) that for an arbitrarily small constant γ we have w.h.p. that

$$\text{for all } i \text{ such that } t_i \leqslant T\text{:} \quad d(A_{i+1},t) = d([n],t) \pm \gamma \quad \text{for all } 1 \leqslant t \leqslant t_i. \tag{4.6}$$

From time t_i onwards algorithm EA now optimizes with respect to the set A_{i+1}. Note that we need to know the set B_{i+1} only in order to check whether we should increase $\tilde{\ell}$. If we thus consider a variant of the process in which we only check whether we reject an offspring or not based on the sets $B_1, \dots, B_i$ (*i.e.* we do not check whether we should increase the level), then we get a sequence of search points $\hat{\xi}^t$ that coincide with ξ^t for all $t < t_{i+1}$. On the other hand, Chernoff bounds and a union bound over all $1 \leqslant t \leqslant T$ imply that for this modified sequence a randomly chosen set $B \subseteq A_{i+1}$ of size $|B| = \beta n$ satisfies $d(B) = d(A_{i+1},t) \pm \gamma$ for all $t < T$. In particular we thus have

$$\text{for all } i \text{ such that } t_i \leqslant T \text{ and all } 1 \leqslant t < \min\{t_{i+1}, T\}\text{:} \quad d(B_{i+1},t) = d(A_{i+1},t) \pm \gamma. \tag{4.7}$$

Finally, the probability of flipping a linear number of bits in one round is exponentially small, so w.h.p. we have

$$\text{for all } 1 \leqslant t \leqslant T\text{:} \quad \text{the number of bits flipped in round } t \text{ is less than } \beta\gamma n. \tag{4.8}$$

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

This condition implies in particular that

$$d(A_i,t+1) = d(A_i,t) \pm \gamma \quad \text{and} \quad d(B_i,t+1) = d(B_i,t) \pm \gamma \quad \text{for all } i \geqslant 1, t \leqslant T.$$

Note also that the probability of flipping at least k bits in one round drops at least geometrically as $k \to \infty$, so the number of zero-bits in ξ^t satisfies condition (2) in Theorem 2.6, and similarly for the number of zero-bits within A_i or within B_i in ξ^t. Note that this implies in particular that we may assume that within the first T steps of algorithm EA the density of any of these sets will never change opposite to the direction of its drift by more than γ.

Assume now that we can show (we will do this below: see (iii)) that algorithm EA with respect to the function $\tilde{f}$ satisfies $d([n],t) \geqslant \varepsilon + 3\gamma$ for all $1 \leqslant t \leqslant T$. Then one easily checks that (4.6) and (4.7) imply that for all i such that $t_i \leqslant T$ we have $d(B_i,t) > \varepsilon$ for all $t \leqslant t_{i-1}$, which implies that we never set $\ell(t) = i$ if $\ell(t-1) \leqslant i-2$. Thus $\ell(t) = \tilde{\ell}(t)$ will follow for all $t \leqslant T$. Therefore, it indeed suffices to analyse EA with respect to $\tilde{f}$, and this is what we will now do. More precisely, for a suitably chosen constant $\delta > 0$ we will show by induction that for every fixed i such that $t_i \leqslant T$ the following three properties hold with probability at least $1 - e^{-2\mu n}$:

(i) $d(A_i,t_i) = \varepsilon \pm 2\gamma$,
(ii) $d([n] \setminus (A_i \cup B_{i-1}),t_i) \geqslant \varepsilon + \delta$,
(iii) for all $t_i \leqslant t < \min\{t_{i+1},T\}$, $d([n],t) \geqslant \varepsilon + 3\gamma$.

Note that the choice of T together with a union bound argument thus implies that these properties all hold for all i simultaneously, which then concludes the proof.

First consider (i). We know by (4.7) that $d(A_i,t_i-1) = d(B_i,t_i-1) \pm \gamma$. For $i > 1$, by induction hypothesis (iii) holds for all $t < t_i$, and we have argued already that this implies $d(B_i,t) > \varepsilon$ for all $t < t_i$. Note that the latter is also true for the base case $i = 1$. As $d(B_i,t_i) \leqslant \varepsilon$ by definition of t_i, the claim thus follows easily together with (4.8).

Next we prove (ii) and (iii) together by induction. More precisely, we will show that if (ii) holds for $i-1$ then (iii) also holds for $i-1$ and (ii) also holds for i. Since $t_0 = 0$ and $A_0 = B_{-1} = \emptyset$, and recalling that ξ^0 is chosen randomly, we have that (ii) holds for $i = 0$, which gives us the start of the induction. So consider some $i \geqslant 1$. By the induction assumption we know that

$$d([n],t_{i-1}) \overset{(i),(ii)}{\geqslant} \alpha(\varepsilon - 2\gamma) + (1-\alpha-\beta)(\varepsilon+\delta) \geqslant \varepsilon + (1-\alpha)\delta - \beta \tag{4.9}$$

by suitable choice of the constants (*cf.* (4.3)). By definition of t_{i-1} and (4.8) we know that

$$\varepsilon - \gamma \leqslant d(B_{i-1},t_{i-1}) \leqslant \varepsilon. \tag{4.10}$$

Let $R = [n] \setminus (A_i \cup B_{i-1})$. By the (random) choice of A_i and (4.6) we thus know that

$$\begin{aligned} d(A_i,t_{i-1}) &= d([n],t_{i-1}) \pm \gamma, \quad \text{and} \\ d(R,t_{i-1}) &= d([n] \setminus B_{i-1},t_{i-1}) \pm \gamma \overset{(4.9),(4.10)}{\geqslant} d([n],t_{i-1}) \pm \gamma. \end{aligned} \tag{4.11}$$

We first compute the drift within A_i while $d(A_i,t) > \varepsilon + \gamma$ (and in particular $t_{i-1} \leqslant t < t_i$ by (4.7)). To do this observe that the choice of the weights in the function $\tilde{f}$ are such that the bit flips within A_i dominate: if they lead to a decreased number of zeros within A_i, then the offspring is accepted regardless of what happens for the bits outside A_i (unless the density within B_{i-1} grows above ε and the offspring is thus rejected for this reason). Similarly, if the number of zeros

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

within A_i increases, then the offspring is always rejected (note that the assumption $d(A_i,t) > \varepsilon+\gamma$ together with (4.7) implies that $\tilde{\ell}$ cannot increase). In particular, the density within A_i is thus non-increasing.

Consider

$$t_i' := \inf\{t \in [t_{i-1},T] : d(A_i,t) \leqslant \varepsilon + (1-\alpha)\delta - \beta\},$$

and note that $t_i' \leqslant t_i$ by (4.7). Then $d(A_i,t_i') \geqslant \varepsilon + (1-\alpha)\delta - \beta - \gamma$: for $t_i' > t_{i-1}$ this follows from the definition of t_i' and (4.8), while for $t_i' = t_{i-1}$ it follows from (4.9) and (4.11). Since the density in A_i is non-increasing whenever $d(A_i,t) > \varepsilon+\gamma$, we deduce that for all $t \in [t_i', \min\{t_i,T\}]$ the probability that we flip at least two zero-bits in A_i to a one is bounded by $O(\varepsilon^2)$. The drift within A_i is thus essentially determined by the probability that we flip exactly one bit within A_i (which is $(1+o(1))\alpha c e^{-\alpha c}$) and that this bit is a zero-bit (which gives a factor $d(A_i,t)$). Note also that the probability that we flip a bit within B_{i-1} is bounded by $c\beta$, which we may assume to be also of the order $O(\varepsilon^2)$ by (4.3). We thus know that the expected drift for the density within A_i is for all $t_i' \leqslant t \leqslant \min\{t_i,T\}$

$$-\frac{(1+o(1))\alpha c e^{-\alpha c}\cdot d(A_i,t) + O(\varepsilon^2)}{\alpha n} = -\frac{\varepsilon c}{n} e^{-\alpha c}(1+O(\varepsilon)) =: \Delta_1.$$

Next we consider what happens within R. By (4.11) and Lemma 4.2 we may assume that $d(R,t) \geqslant d(A_i,t) - 3\gamma$ for all $t_{i-1} \leqslant t \leqslant \min\{t_i,T\}$. In particular,

$$d(R,t_i') \geqslant d(A_i,t_i') - 3\gamma \geqslant \varepsilon + (1-\alpha)\delta - \beta - 4\gamma.$$

We want to show that the density $d(R)$ *increases* to at least $\varepsilon+\delta$. Thus, we may assume without loss of generality that $d(R,t) = \varepsilon \pm 2\delta$ for all $t_i' \leqslant t < \min\{t_i,T\}$. We will now compute a lower bound on the drift within R. Consider first the case that the number of zero-bits in A_i remains the same (which happens with probability at most $e^{-\alpha c} + O(\varepsilon)$). In this case we accept the changes within R similarly as we did for those within A_i, as explained above. In other words: with probability at most $e^{-\alpha c}(|R|/n)ce^{-(|R|/n)c}\cdot\varepsilon + O(\delta)$ the number of zero-bits within R decreases by one, and the contribution of events where it may increase is $O(\varepsilon^2)$.

If the number of zero-bits within A_i decreases (which happens with probability $\alpha c e^{-\alpha c}\varepsilon + O(\varepsilon^2)$), then we accept any change within R. In this case the expected change in the number of zero-bits within R is $c|R|/n + O(\varepsilon)$. Summarizing, we observe that the expected drift for the density within R is

$$\begin{aligned}&\frac{\alpha c e^{-\alpha c}\varepsilon\cdot(c|R|)/n - e^{-\alpha c}(|R|/n)ce^{-(|R|/n)c}\cdot\varepsilon + O(\delta+\varepsilon^2)}{|R|}\\ &\quad= \frac{\varepsilon c}{n}e^{-\alpha c}(\alpha c - e^{-(1-\alpha)c}) + O\left(\frac{1}{n}(\delta+\varepsilon^2+\beta)\right) =: \Delta_2,\end{aligned}$$

where we used that $|R|/n = 1-\alpha+O(\beta)$.

Observe that, up to the error terms, the ratio $\Delta_2/|\Delta_1|$ is exactly the left-hand side of (4.2). By making δ,ε and β small enough we may thus assume that

$$\frac{\Delta_2}{|\Delta_1|} > \frac{\alpha}{1-\alpha} + O(\beta+\gamma). \tag{4.12}$$

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

We now employ the tail bounds in Theorem 2.6. As we have $d(A_i,t_i') \geqslant \varepsilon+(1-\alpha)\delta-\beta-\gamma$, Theorem 2.6 tells us that with probability $1-e^{-\Omega(n)}$ it takes at least $S=[(1-\alpha)\delta-\beta-3\gamma]/|\Delta_1|$ steps to decrease it to $\varepsilon+\gamma$. Before this density is reached the density in B_i is strictly larger than ε by (4.7). On the other hand, $d(R,t_i') \geqslant \varepsilon+(1-\alpha)\delta-\beta-4\gamma$. Again by Theorem 2.6 the density in R is increased to at least $\varepsilon+\delta+\gamma$ with probability $1-e^{-\Omega(n)}$ after at most $(\alpha\delta+\beta+6\gamma)/\Delta_2 \overset{(4.12)}{<} S$ steps, and stays above $\varepsilon+\delta$ by Theorem 2.6(b). This proves both (ii) and (iii) of the induction and thus concludes the proof of the theorem. □

A natural question is to determine the behaviour of the algorithm if the mutation parameter c lies in the interval $[1,2.2)$. We leave this as an intriguing open problem.

4.2. Linear functions

In this section we give a new proof that the optimization time of algorithm EA with mutation parameter c is $(1+o(1))(e^c/c)n\log n$ for any constant $c>0$ if we restrict ourselves to *linear* functions $f(\xi)=\sum_{i=1}^n a_i\xi_i$, as was first proved in [18].

Theorem 4.4 ([18]). *For any linear function f and every constant $c>0$, algorithm EA with mutation rate c/n has optimization time $(1+o(1))e^c/c\cdot n\log n$ w.h.p.*

The lower bound in Theorem 4.4 is rather straightforward, and we only give a sketch. Assume the number X_t of zero-bits in ξ^t is small, say $X_t \leqslant n^{1-\varepsilon}$. Then the probability of flipping at least two zero-bits in the same round is $O(n^{-2\varepsilon})$. This will turn out to be negligibly small, so let us ignore such 'bad' rounds for the moment, for the sake of clarity. Then the number of zero-bits can only decrease if we flip exactly one zero-bit, and no one-bits. Let r be the number of rounds in which we flip exactly one position. Then the number of such rounds until no zero-bits are left is (if there were no bad rounds) the same as the number of rounds needed in a coupon collector process to reduce the number of missing coupons from $n^{1-\varepsilon}$ to 0, which is well known to be at least $(1-\varepsilon-o(1))n\log n$ w.h.p. On the other hand, in each round we have probability $(1+o(1))ce^{-c}$ of flipping exactly one bit. So the (total) number of rounds until we have seen w.h.p. $r=(1-\varepsilon-o(1))n\log n$ rounds in which we flip only a single bit is $(1-\varepsilon-o(1))e^c/c\cdot n\log n$, by Chernoff bounds. It is easy to adapt the argument to cope also with bad rounds, for example by dividing the process into phases from $n^{1-\varepsilon}$ to $n^{1-2\varepsilon}$, from $n^{1-2\varepsilon}$ to $n^{1-4\varepsilon}$, *etc.*, and observing that in each phase with very high probability the number of bad rounds is so small that it does not significantly change the time to reach the next phase.

For the upper bound, we split the proof in two steps: we need a linear time until there are fewer than εn zero-bits left (Lemma 4.6), which are then removed in an additional $(1+o(1))e^c/c\cdot n\log n$ steps (Lemma 4.5). Surprisingly, the proof of the latter part is rather straightforward and uses known techniques, while the proof of Lemma 4.6 is significantly harder. For this phase we present a new proof that is different from previous approaches. We do not claim that our proof is simpler than the proof in [18], but our new approach is arguably a very intuitive one: when the bits of large weight are almost optimized, then optimization of bits of lower weight is almost unaffected by those large-weight bits. We make this precise by interpreting the effect of bits of large weight as noise, and we show by an induction-type argument that the amount of noise is tolerable by algorithm EA (Lemma 4.7).

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

Lemma 4.5. *Let $c > 0$ be a constant. Then there is a constant $\varepsilon > 0$ such that the following holds for every $\xi^0 \in \{0,1\}^n$ that has at most εn zero-bits. If algorithm EA with mutation rate c/n starts with initial string ξ^0 then w.h.p. it finds the optimum of any n-bit linear function with positive weights after at most $(1+o(1))e^c/c \cdot n \log n$ steps.*

Proof. Without loss of generality we assume that the weights a_i of the linear function are sorted and positive, that is, $a_1 \geqslant \cdots \geqslant a_n > 0$. To make the upcoming argument smoother, we will assume that in the special case that exactly one one-bit and one zero-bit are flipped, and both have equal weights, then the algorithm rejects the offspring. Note that we may make this assumption because the offspring and the parent are identical up to reordering the indices.

Our aim is to apply Theorem 2.5. For the function g we count the number of zeros, but we also distinguish between the positions where the zeros occur, giving those with a higher coefficient a_i also a higher weight. It seems natural to use the a_i themselves as weights. However, this does not work since Theorem 2.3 only gives a very bad bound if, for example, $a_1/a_n \gg n$. Instead, we fix a constant $C > 0$, to be chosen later, and we use the objective function

$$g(\xi) := 2^C \cdot \sum_{1 \leqslant i \leqslant n, \xi_i = 0} 2^{-Ci/n}.$$

Let X_t be the number of zero-bits in ξ^t, and observe that $X_t \leqslant g(\xi^t) \leqslant 2^C X_t$.

We want to compute the drift $\mathbb{E}[g(\xi^{t+1}) \mid \xi^t = \xi] - g(\xi)$, so assume $\xi^t = \xi$. First observe that the probability of flipping at least two zero-bits in the same round is $O((X_t/n)^2) = O(g(\xi)^2/n^2)$. Moreover, conditioning on flipping at least two zero-bits, the expected number of bit flips is still $O(1)$, so this case contributes at most $O(g(\xi)^2/n^2)$ to the drift (which, as we will see, is negligible). On the other hand, if no zero-bit is flipped, then the offspring is rejected. So it remains to consider the case where exactly one zero-bit is flipped (and possibly additional one-bits are flipped). We call the index of this bit k. If no other bit is flipped, then we accept the offspring, and the probability of this case is $(1+o(1))ce^{-c}/n$ for any fixed k. Hence, this case contributes

$$-(1+o(1))\frac{ce^{-c}}{n} 2^C \cdot \sum_{1 \leqslant k \leqslant n, \xi_k = 0} 2^{-Ck/n} = -(1+o(1))\frac{ce^{-c}}{n} g(\xi)$$

to the drift. It remains to consider the case where also one-bits are flipped. If all one-bits that are flipped have an index i that satisfies $i > k$, then the offspring may or may not be accepted, but in any case the function g increases by at most $S_k := 2^C \sum_{j>k, \xi_j \text{ flipped}} 2^{-Cj/n}$. On the other hand, if a one-bit i is flipped with $i < k$ then the fitness does not increase, so the offspring is rejected. (Here we use the assumption from the beginning of the proof.) Hence, the change in g is 0. Note that S_k is always non-negative, so we can upper-bound the positive contribution to the drift from all cases where we flip exactly one zero-bit by

$$\begin{aligned}\sum_{1 \leqslant k \leqslant n, \xi_k = 0} \mathbb{P}[\xi_k \text{ flips}] \cdot \mathbb{E}[S_k] &= \sum_{1 \leqslant k \leqslant n, \xi_k = 0} \frac{c}{n} \cdot 2^C \sum_{i=k+1}^{n} \frac{c}{n} 2^{-Ci/n} \\ &\leqslant \frac{c2^C}{n} \sum_{1 \leqslant k \leqslant n, \xi_k = 0} 2^{-Ck/n} \underbrace{\sum_{i=1}^{\infty} \frac{c}{n} 2^{-Ci/n}}_{= c/(C \log 2) + o(1)} = \frac{(1+o(1))c^2}{nC \log 2} g(\xi).\end{aligned}$$

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

Assume that C is so large that $\delta := ce^c/(C\log 2) \leqslant 1/3$. Then, bringing all cases together, we obtain for any string ξ,

$$\mathbb{E}[g(\xi^{t+1}) \mid \xi^t = \xi] \leqslant g(\xi)\left(1 - (1+o(1))\frac{(1-\delta)ce^{-c}}{n} + O\left(\frac{g(\xi)}{n^2}\right)\right), \tag{4.13}$$

where the term $O(g(\xi)/n^2)$ covers the case that at least two zero-bits are flipped. This inequality implies two things. Firstly, we apply (4.13) for $C = 3ce^c/\log 2$ (*i.e.* $\delta = 1/3$). Since $g(\xi^0) \leqslant 2^C X_0 \leqslant 2^C \varepsilon n$, we may choose ε such that the error term $O(g(\xi^0)/n^2)$ is at most $\frac{1}{3}ce^{-c}/n$ at time $t = 0$. Then by Theorem 2.6, for every $\gamma > 0$ w.h.p. $X_t \leqslant g(\xi^t)$ becomes smaller than γ in time $t_0 = O(n)$, and it stays so for an exponential number of steps. Having proved this, we may now apply (4.13) for any constant $C > 3ce^c/\ln 2$ (and thus, any constant $0 < \delta < 1/3$). Since then $g(\xi^t) \leqslant 2^C X_t \leqslant 2^C \gamma n$, we may choose γ sufficiently small that the error term $O(g(\xi^t)/n^2)$ is at most $\delta ce^{-c}/n$ for $t \geqslant t_0$. Thus we obtain

$$\mathbb{E}[g(\xi^{t+1}) \mid \xi^t = \xi] \leqslant g(\xi)(1 - (1-3\delta)ce^{-c}/n).$$

Then by Theorem 2.5, w.h.p. we reach zero after at most

$$(1+o(1))\frac{ne^c}{(1-3\delta)c}\log(g(\xi^{t_0}))$$

additional steps. Since $g(\xi^{t_0}) = O(n)$, and since we can choose δ arbitrarily small, this proves that w.h.p. the algorithm finds the optimum after $(1+o(1))e^c/c \cdot n\log n$ steps. □

It remains to show that the number of zeros is at most εn after a short (*i.e.* linear) time.

Lemma 4.6. *For any $c, \varepsilon > 0$ there is $C > 0$ such that after Cn steps of algorithm EA on any n-bit linear function with positive weights, w.h.p. there are at most εn zero-bits left.*

Even though Lemma 4.6 implies that in absolute terms the first phase of algorithm EA takes only linear time, while the second phase requires an additional log-factor, the proof of Lemma 4.6 turns out to be particularly tricky. We will use an induction-type argument, but not over natural numbers, but rather over values of c. For technical reasons, we fix $c_{\max} > 0$ sufficiently large (but otherwise arbitrary) and prove the statement for all $0 < c < c_{\max}$. Note that since $c_{\max}$ is arbitrary, this nevertheless implies the desired result for all constants $c > 0$.

For the inductive argument to go through, we need to use a slightly stronger statement, which we spell out in Lemma 4.7. We will allow two types of noise. We say that there is δ_1-*noise of type 1* if in each round we flip a coin, and with probability δ_1 an adversary may decide whether we go to the offspring or not, regardless of the objective value. We say that there is δ_2-*noise of type 2* if in each round we flip a coin, and with probability $1 - \delta_2$ an adversary may decide on an arbitrary non-negative penalty by which the fitness of the offspring is reduced. Note that the adversary has to set the penalty without looking at the outcome of the offspring.

Note that Lemma 4.7 contains Lemma 4.6 as a special case (the case without noise), so proving Lemma 4.7 also concludes the proof of Theorem 4.4.

Lemma 4.7. *If $c_{\max} > 0$ is a sufficiently large constant, then for all $0 < \varepsilon \leqslant 1$ and all $0 < c < c_{\max}$ there is $C > 0$ such that for all $n \geqslant 1$ the following holds. Consider algorithm EA with*

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

arbitrary starting string, with mutation rate c/n, and with δ_1-noise of type 1 and δ_2-noise of type 2, where $\delta_1 = \delta_1(\varepsilon, c) := \varepsilon \exp\{-ce^{4c_{\max}+e^{5c_{\max}}}\}$ and $\delta_2 = \delta_2(c) := e^{2(c-c_{\max})}$. Then for any n-bit linear function with positive weights, w.h.p. for all $t \in [Cn, e^{n/C}]$ the tth search point ξ^t has at most εn zero-bits.

Proof. We first investigate the case that c is small. Observe that for a fixed $0 < \varepsilon \leqslant 1$, and c sufficiently small, the probability of flipping more than one bit in the same round can be bounded by $O(c^2)$. Thus, if the number of zero-bits is at least $\varepsilon n/2$ then the drift in the number of zero-bits is at most $-\delta_2 ce^{-c}\varepsilon/4 + \delta_1 c + O(c^2)$. Here $ce^{-c}\varepsilon/2$ is a lower bound (for large n) for the probability that we flip exactly one bit and that this bit is a zero-bit and the factor $\delta_2 - \delta_1 \geqslant \delta_2/2$ is a lower bound for the probability that the noise of type 1 or 2 will not obstruct us from accepting this bit flip. The remaining two terms are upper bounds for the cases in which the number of zero-bits increases. That is, if $c_{\max}$ is sufficiently large then for $c = \varepsilon e^{-4c_{\max}}$ this drift is negative, and the lemma follows for the pair ε, c from Theorem 2.6. In other words, for each fixed $0 < \varepsilon \leqslant 1$ the lemma holds for some $c > 0$.

Assume now for the sake of contradiction that the lemma is false for some $0 < \varepsilon \leqslant 1$, and let $\varepsilon_0 > 0$ be the supremum over all ε for which it is false. Then there exists an $\varepsilon > \varepsilon_0/2$ for which the lemma is false. From now on we will stick with this ε and derive a contradiction. Let $c_0 = c_0(\varepsilon)$ be the infimum over all c for which the lemma fails for the pair (ε, c). Then $c_0 \geqslant \varepsilon e^{-4c_{\max}}$ by the considerations above. We will show that there exists $\rho_c = \rho_c(\varepsilon) > 0$ such that the lemma holds for all pairs (ε, c) with $c_0 \leqslant c < \min\{c_0(1+\rho_c), c_{\max}\}$, thus yielding the desired contradiction. In the following we assume that c is an arbitrary but fixed value in the interval $[c_0, \min\{c_0(1+\rho_c), c_{\max}\})$, where ρ_c is defined after equation (4.16) below.

Before going into details we explain the main idea of our proof. First note that the lemma is *true* for $\varepsilon^+ := 2\varepsilon$: if $2\varepsilon > 1$ this is trivial, otherwise this follows from the definition of ε_0 and the fact that $\varepsilon > \varepsilon_0/2$. Thus we know that algorithm EA (with parameter $p = c/n$) reduces the number of zeros in linear time to $2\varepsilon n$ (and will stay below this number for an exponential time). We will show that there exists $\alpha > 0$ and $0 < \hat{\varepsilon} < \varepsilon$ so that after an additional linear number of steps, there are at most $\hat{\varepsilon}\alpha n$ zero-bits left among the first αn bits. The remaining part has only $\tilde{n} := (1-\alpha)n$ bits, thus the mutation rate in this part is $(1-\alpha)c/\tilde{n}$. We will choose α in such a way that $\tilde{c} := (1-\alpha)c < c_0$, so we know from the choice of c_0 that the lemma is true for the pair $(\varepsilon, \tilde{c})$. Thus, the remaining string contains at most $\varepsilon\tilde{n}$ zeros after linear time, at least if we were to apply the algorithm only to that part of the string. In order to really derive a contradiction we must take into account that the initial αn part of the string provides noise for the optimization of the remaining part: if a zero-bit in the first part is flipped into a one-bit (and possibly other bits in this part are also flipped), then this gives noise of type 1; if one or several one-bits are flipped, then this may give noise of type 2. We thus need to check that both types of noise are sufficiently small. More precisely, we will check that this additional noise is small enough, so that the total noise is less than the values for δ_1 and δ_2 with respect to ε and $\tilde{c}$.

To prove these claims we need to define some constants. Let $\delta_1 := \delta_1(\varepsilon, c)$ and $\delta_2 := \delta_2(c)$ be as defined in the statement of the lemma. Moreover, let

$$\hat{\varepsilon} := e^{4c_{\max}+e^{5c_{\max}}}\delta_1 \quad \text{and} \quad \beta := \delta_1. \tag{4.14}$$

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

Recall that $\varepsilon e^{-4c_{\max}} \leqslant c_0 \leqslant c < c_{\max}$. So we may assume without loss of generality that $c_{\max}$ is sufficiently large that

$$(2e\varepsilon c \exp\{-e^{5c_{\max}}\})^{\exp\{e^{5c_{\max}}\}} \leqslant \delta_1 \quad \text{and} \quad c_{\max}\beta = c_{\max}\delta_1 \leqslant \frac{1}{200}\hat{\varepsilon}\delta_2 e^{-c_{\max}-e^{5c_{\max}}} \leqslant \frac{1}{16}. \tag{4.15}$$

Note that (4.15) implies in particular that

$$1 - c\beta \geqslant e^{-2c\beta} \quad \text{and} \quad 1 - 2c\beta \geqslant e^{-4c\beta}. \tag{4.16}$$

With these definitions to hand, we let ρ_c be so small that $c(1-\beta) < c_0$ whenever $c_0 \leqslant c \leqslant c_0(1+\rho_c)$. Note that β implicitly depends on c and ε here, but it is easy to check that the definition $\rho_c := \min_{\varepsilon e^{-4c_{\max}} \leqslant c \leqslant c_{\max}} \{\beta(c)\}$ yields the desired inequalities.

As before, we may assume that the bits are sorted by weights, in descending order. We call the first βn bits *the first region* R_1, the next βn bits *the second region* R_2, and the remaining $(1-2\beta)n$ bits the *third region* R_3. We assume further without loss of generality that the maximum weight in the second section is 1 (by scaling all n weights by the same factor), and we split the second and third region into blocks as follows. For $i \geqslant 1$, the block B_i consists of all bits in $R_2 \cup R_3$ with weight in $[(1+\mu)^{-i-1}, (1+\mu)^{-i})$, where we will define μ in a moment. Note that some of the blocks may be empty. Recall that $d(B_i) = d(B_i, t)$ is the density of zero-bits in B_i with respect to ξ^t, for any non-empty block B_i, and similarly for $d(R_1)$, *etc.*

We distinguish two cases (in which we will choose $\alpha = \beta$ and $\alpha = 2\beta$, respectively) depending on how many blocks are fully contained in the second region. Note that we also count empty blocks here, so the number of fully contained blocks in the second region is (up to rounding issues) $\log_{1+\mu} 1/a_i$, where $i = 2\beta n$ is the first index in the third region. To describe the cases precisely we need two more constants. Let

$$D := \min\{D' \in \mathbb{N}_0 \mid (1+\mu)^{D'} \geqslant \exp\{e^{5c_{\max}}\}\}, \tag{4.17}$$
$$\text{where } \mu := \frac{\exp\{-c_{\max} - 2e^{5c_{\max}}\}\delta_2\beta\hat{\varepsilon}}{100}.$$

First case. There are at least D blocks fully contained in R_2.

In this case we will set $\alpha = \beta$. Recall from above that we may assume that the number of zero-bits is at most $2\varepsilon n$. We consider the effect of the algorithm on R_1, that is, on the first βn bits. We want to argue that we can bring $d(R_1)$ below $\hat{\varepsilon}$ in linear time, and that it stays so for exponential time. To this end, we will show that we have a negative drift whenever $d(R_1) \geqslant \hat{\varepsilon}/2$, so let us assume $d(R_1) \geqslant \hat{\varepsilon}/2$. The probability that we flip exactly one zero-bit in R_1, and no other bit anywhere, is at least $ce^{-c_{\max}}\beta\hat{\varepsilon}/4$ if n is sufficiently large; in this case we accept this bit flip with probability at least $\delta_2 - \delta_1 > \delta_2/2$, and the number of zero-bits decreases by one. On the other hand, we can increase the number of zero-bits in R_1 only if one of the following (non-disjoint) events occurs.

(i) At least two bits in R_1 are flipped. This happens with probability at most $c^2\beta^2$ and, conditioned on this, the expected increase is then at most $\beta c + 2 \leqslant 3$, with room to spare.

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

(ii) At least one bit is flipped in R_2. This happens with probability at most $c\beta$ and the expected increase (in R_1) is then at most βc.

(iii) More than $(1+\mu)^D$ zero-bits are flipped in R_3. Since there are at most $2\varepsilon n$ zero-bits, this probability is bounded by the probability that a binomially distributed random variable with parameters $2\varepsilon n$ and c/n is at least $(1+\mu)^D \geqslant \exp\{e^{5c_{\max}}\}$. Using the Chernoff bound

$$\mathbb{P}[\mathrm{Bin}(N,q) \geqslant (1+\delta)Nq] \leqslant (e/(1+\delta))^{(1+\delta)Nq}$$

with $Nq = 2\varepsilon c$ and $(1+\delta)Nq = \exp\{e^{5c_{\max}}\}$, we see that this probability is at most

$$(2e\varepsilon c \cdot \exp\{-e^{5c_{\max}}\})^{\exp\{e^{5c_{\max}}\}}.$$

The expected increase in this case is again at most βc.

(iv) There is noise of type 1. This happens with probability δ_1 and also results in an expected increase of at most βc.

Summarizing the above terms, we see that the expected drift for the number of zero-bits in the first region can be bounded by

$$\begin{aligned}\mathbb{E}[X_{t+1} - X_t] &\leqslant -\frac{1}{8}\delta_2 c e^{-c_{\max}}\beta\hat{\varepsilon} + 3c^2\beta^2 + c^2\beta^2 + (2e\varepsilon c \exp\{-e^{5c_{\max}}\})^{\exp\{e^{5c_{\max}}\}}\beta c + c\beta\delta_1 \\ &\overset{(4.15)}{=} -\Omega(1).\end{aligned}$$

Hence, by Theorem 2.6 after linear time we will reach $d(R_1) \leqslant \hat{\varepsilon}/2$, and since we have negative drift whenever $d(R_1)$ is in the interval $[\hat{\varepsilon}/2, \hat{\varepsilon}]$, it will stay below $\hat{\varepsilon}$ for exponential time.

We want to apply the lemma for ε and $\tilde{c} = c(1-\beta) < c_0$ to conclude that in this period $d(R_2 \cup R_3)$ drops below ε. So we must show that the additional noise caused by the first region is not too large. Noise of type 1 is generated whenever a zero-bit in R_1 is flipped, which happens with probability at most $c\beta\hat{\varepsilon}$. Thus the total amount of noise of type 1 is at most

$$\delta_1(\varepsilon, c) + c\beta\hat{\varepsilon} \overset{(4.14)}{\leqslant} \delta_1(\varepsilon, c) + c\beta\frac{-\partial\delta_1(\varepsilon, c)}{\partial c} \leqslant \delta_1(\varepsilon, \tilde{c}),$$

where in the last step we used convexity of δ_1. For noise of type 2, the fitness of the offspring may decrease if any bit in R_1 is flipped, which happens with probability at most $c\beta$. Hence, the overall probability that the adversary will *not* add a penalty is at least

$$\delta_2(c) \cdot (1 - c\beta) \overset{(4.16)}{\geqslant} \delta_2(\tilde{c}) e^{2c\beta} \cdot e^{-2c\beta} = \delta_2(\tilde{c}),$$

as required. This concludes the first case.

Second case. There are fewer than D blocks fully contained in the second region.

In this case we will apply the lemma to the third region ($\alpha = 2\beta$) with parameters ε and $\tilde{c} = c(1-2\beta) < c_0$. To bound the noise coming from R_1 and R_2, we will first show that w.h.p. $d(R_2) \leqslant \hat{\varepsilon}$ after linear time. We use the auxiliary fitness function

$$\phi(\xi) := \sum_{i \geqslant 1} |\{j \in B_i : \xi_j = 0\}| \cdot (1+\mu)^{-i},$$

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

and we let $\phi_{\max}$ be the fitness of the zero string, that is, $\phi_{\max} = \sum_{i \geqslant 1} |B_i| \cdot (1+\mu)^{-i}$. Note that by definition of the second case

$$\phi_{\max} \geqslant \beta n \cdot (1+\mu)^{-D} \geqslant \beta n \cdot \exp\{-e^{5c_{\max}}\}/(1+\mu)$$

by definition of D.

We call a block in the third region *long* if it has length at least $\beta\mu n/2$, and *short* otherwise. The contribution of short blocks to $\phi_{\max}$ is at most

$$\sum_{B_i \text{ short}} |B_i|(1+\mu)^{-i} < \frac{1}{2}\beta\mu n \sum_{i \geqslant 0} (1+\mu)^{-i} < \beta n. \tag{4.18}$$

As long blocks have linear length, there can only be a constant number of long blocks. Thus we may apply Lemma 4.2 to conclude that (after a linear number of steps) we have $d(B_i,t) \geqslant d(R_2,t)/2$ for all long blocks B_i and for exponential time. Note that the contribution of a block B_i to $\phi(\xi^t)$ can be written as $d(B_i,t)|B_i|(1+\mu)^{-i}$, and the contribution of R_2 is at least

$$d(R_2,t)\beta n(1+\mu)^{-D},$$

where the latter follows from the definition of the second case. Hence,

$$\begin{aligned}\phi(\xi^t) &\geqslant \frac{d(R_2,t)\beta n}{(1+\mu)^D} + \sum_{B_i \text{ long}} \frac{1}{2} d(R_2,t)|B_i|(1+\mu)^{-i} \\ &> \frac{d(R_2,t)}{2(1+\mu)^D}\left(2\beta n + \sum_{B_i \text{ long}} |B_i|(1+\mu)^{-i}\right) \overset{(4.18)}{\geqslant} \frac{d(R_2,t)\phi_{\max}}{2(1+\mu)^D}.\end{aligned} \tag{4.19}$$

Now we are ready to show that ϕ has a negative drift for $d(R_2,t) \geqslant \hat{\varepsilon}/2$, so assume the latter. First observe that with probability $ce^{-c}(1+o(1))$ there is exactly one bit flipped, and there is no noise with probability at least $\delta_2 - \delta_1 > \delta_2/2$. In this case each bit has probability $1/n$ of being the flipped one, so in this case we decrease $\phi(\xi^t)$ by $\phi(\xi^t)/n$ on average. Thus the contribution to the drift is (for sufficiently large n) more negative than

$$-ce^{-c}\delta_2\phi(\xi^t)/(4n) \leqslant -ce^{-c}\delta_2\hat{\varepsilon}(1+\mu)^{-D}\phi_{\max}/(16n)$$

by (4.19). If at least one bit in R_1 is flipped (which happens with probability at most $c\beta$), then this increases $\phi(\xi^t)$ in expectation by at most $c\phi_{\max}/n$, so this adds at most $c^2\beta\phi_{\max}/n$. Similarly, noise of type 1 contributes at most $c\delta_1\phi_{\max}/n$ to the drift. Finally, consider the case where no bit in R_1 flips, and where no noise of type 1 occurs. Let r_i and s_i be the number of bits in the ith block B_i that flip from zero to one and from one to zero, respectively. Then a necessary (but not sufficient) condition for accepting the offspring is

$$\sum_i s_i(1+\mu)^{-i-1} - \sum r_i(1+\mu)^{-i} \leqslant 0.$$

Note that this condition is still necessary if noise of type 2 occurs. Thus, in this case

$$\Delta\phi = \sum_i s_i(1+\mu)^{-i} - \sum r_i(1+\mu)^{-i} \leqslant \mu \sum_i r_i(1+\mu)^{-i} \leqslant \mu \sum_i r_i,$$

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

which is at most $c\mu$ in expectation. Summarizing, we see that for $d(R_2,t) \geqslant \hat{\varepsilon}/2$ the drift is at most

$$\mathbb{E}[\phi(\xi^{t+1}) - \phi(\xi^t)] \leqslant -ce^{-c}\delta_2\hat{\varepsilon}(1+\mu)^{-D}\phi_{\max}/(16n) + c^2\beta\phi_{\max}/n + c\delta_1\phi_{\max}/n + c\mu$$
$$\overset{(4.15),(4.17)}{=} -\Omega(1)$$

for sufficiently large $c_{\max}$.

As $d(R_2,t) > \hat{\varepsilon}/2$ implies a negative drift for $\phi(\xi^t)$, Theorem 2.6 implies that after a linear number of steps we have $d(R_2) \leqslant \frac{3}{4}\hat{\varepsilon}$, and that this bound remains true for an exponential number of steps. By Lemma 4.2 the density in R_1 is, at least after another linear number of steps, at most $\frac{4}{5}\hat{\varepsilon}$. Combining the two facts, we conclude that from then on we have $d(R_1 \cup R_2) \leqslant \hat{\varepsilon}$ for an exponential number of steps.

We now apply the lemma as in the first case, this time to R_3. That is, we set $\alpha = 2\beta$, and apply the lemma for ε and $\tilde{c} = c(1-2\beta) < c_0$. We need to check that we do not introduce too much noise. For noise of type 1, we have at most

$$\delta_1(\varepsilon,c) + 2c\beta\hat{\varepsilon} \overset{(4.14)}{\leqslant} \delta_1(\varepsilon,c) + 2c\beta\frac{-\partial\delta_1(\varepsilon,c)}{\partial c} \leqslant \delta_1(\varepsilon,\tilde{c}),$$

where again we used convexity in the last step. For noise of type 2 we conclude as in the first case that the probability of *not* having noise is at least

$$\delta_2(c)(1-2c\beta) \overset{(4.16)}{\geqslant} \delta_2(\tilde{c})e^{4c\beta}\cdot e^{-4c\beta} \geqslant \delta_2(\tilde{c}),$$

as desired. □

Acknowledgement

After being appointed at ETH Zürich, Jiří Matoušek convinced his colleagues in the algorithms group to establish a joint seminar to teach undergraduate students to do (and love) research. In the third edition of this seminar the second author of this paper presented the topic at hand. More precisely, the task was to come up with fresh ideas to re-prove the state of the art elegantly. While this turned out to be too hard a task for undergraduate students, it eventually led to the present paper, which for us will always be a memory of our much admired colleague Jirka Matoušek.

References

[1] Bäck, T. (1996) *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press.
[2] Doerr, B. and Goldberg, L. A. (2013) Adaptive drift analysis. *Algorithmica* **65** 224–250.
[3] Doerr, B., Jansen, T., Sudholt, D., Winzen, C. and Zarges, C. (2013) Mutation rate matters even when optimizing monotonic functions. *Evolutionary Comput.* **21** 1–27.
[4] Doerr, B., Johannsen, D. and Winzen, C. (2012) Multiplicative drift analysis. *Algorithmica* **64** 673–697.
[5] Doerr, B., Johannsen, D. and Winzen, C. (2012) Non-existence of linear universal drift functions. *Theoret. Comput. Sci.* **436** 71–86.
[6] Droste, S., Jansen, T. and Wegener, I. (2002) On the analysis of the (1+1) evolutionary algorithm. *Theoret. Comput. Sci.* **287** 131–144.

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press

[7] Hajek, B. (1982) Hitting-time and occupation-time bounds implied by drift analysis with applications. *Adv. Appl. Probab.* **13** 502–525.
[8] He, J. and Yao, X. (2001) Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence* **127** 57–85.
[9] He, J. and Yao, X. (2004) A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Comput.* **3** 21–35.
[10] Jägersküpper, J. (2011) Combining Markov-chain analysis and drift analysis. *Algorithmica* **59** 409–424.
[11] Jansen, T. (2007) On the brittleness of evolutionary algorithms. In *FOGA 2007: 9th International Workshop on Foundations of Genetic Algorithms*, Springer, pp. 54–69.
[12] Johannsen, D. (2010) Random combinatorial structures and randomized search heuristics. PhD thesis, Universität des Saarlandes.
[13] Kötzing, T. (2016) Concentration of first hitting times under additive drift. *Algorithmica* **75** 490–506.
[14] Mitavskiy, B., Rowe, J. and Cannings, C. (2009) Theoretical analysis of local search strategies to optimize network communication subject to preserving the total number of links. *Internat. J. Intelligent Comput. Cybernet.* **2** 243–284.
[15] Mühlenbein, H. (1992) How genetic algorithms really work: Mutation and hillclimbing. In *PPSN 1992: 2nd International Conference on Parallel Problem Solving from Nature*, Elsevier, pp. 15–26.
[16] Oliveto, P. S. and Witt, C. (2011) Simplified drift analysis for proving lower bounds in evolutionary computation. *Algorithmica* **59** 369–386.
[17] Oliveto, P. S. and Witt, C. (2012) Erratum: Simplified drift analysis for proving lower bounds in evolutionary computation. arXiv:1211.7184
[18] Witt, C. (2013) Tight bounds on the optimization time of a randomized search heuristic on linear functions. *Combin. Probab. Comput.* **22** 294–318.

https://doi.org/10.1017/S0963548318000275 Published online by Cambridge University Press