Optimal Mutation Rates for the (1+ λ) EA on OneMax Through Asymptotically Tight Drift Analysis

Gießen, Christian; Witt, Carsten

Published in:
Algorithmica

Link to article, DOI:
10.1007/s00453-017-0360-y

Publication date:
2018

Document Version
Peer reviewed version

Algorithmica manuscript No.
(will be inserted by the editor)

Optimal Mutation Rates for the (1+λ) EA on OneMax Through Asymptotically Tight Drift Analysis ⋆

Christian Gießen · Carsten Witt

Received: November 20, 2019/ Accepted: date

Abstract We study the (1+λ) EA, a classical population-based evolutionary algorithm, with mutation probability c/n, where $c > 0$ and λ are constant, on the benchmark function OneMax, which counts the number of one-bits in a bitstring.

We improve a well-established result that allows to determine the first hitting time from the expected progress (drift) of a stochastic process, known as the variable drift theorem. Using our improved result, we show that upper and lower bounds on the expected runtime of the (1+λ) EA obtained from variable drift theorems are at most apart by a small lower order term if the exact drift is known. This reduces the analysis of expected optimization time to finding an exact expression for the drift.

We then give an exact closed-form expression for the drift and develop a method to approximate it very efficiently, enabling us to determine approximate optimal mutation rates for the (1+λ) EA for various parameter settings of c and λ and also for moderate sizes of n. This makes the need for potentially lengthy and costly experiments in order to optimize c for fixed n and λ for the optimization of OneMax unnecessary.

Interestingly, even for moderate n and not too small λ it turns out that mutation rates up to 10% larger than the asymptotically optimal rate $1/n$ minimize the expected runtime. However, in absolute terms the expected runtime does not change by much when replacing $1/n$ with the optimal mutation rate.

⋆ A preliminary version of this paper was published at GECCO 2016 [GW16].

Christian Gießen
DTU Compute, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark, E-mail: cgie@dtu.dk

Carsten Witt
DTU Compute, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark, E-mail: cawi@dtu.dk

Keywords Runtime Analysis; Populations; Mutation

1 Introduction

The runtime analysis of randomized search heuristics has made significant progress over the past two decades. A broad variety of randomized search heuristics, for example evolutionary algorithms (EAs), ant colony optimization and randomized local search have been considered for specific artificial functions, and also for combinatorial optimization problems. The analysis of evolutionary algorithms on simple pseudo-Boolean functions has been tremendously fruitful and led to a wide range of results and techniques. In most cases, runtime analysis is performed in an asymptotic fashion with respect to the expected optimization time (see [AD11, Jan13, NW10] for an overview).

In the past couple of years there has been a surge of interest in non-asymptotic analyses of EAs, i. e. in determining the runtime exactly, up to lower order terms. For example, bounds on the expected runtime of the well-known (1+1) EA on the classical OneMax function have enjoyed a series of improvements. It has been known for a long time that the expected runtime is $\Theta(n \ln n)$ and goes back to Droste, Jansen and Wegener [DJW02]. The first asymptotically tight bound of $(1 \pm o(1))en \ln n$ was given by Witt [Wit13] for the (1+1) EA on general linear pseudo-Boolean functions with positive weights. Recently, the expected runtime of the (1+1) EA on OneMax has been stated exactly, up to subconstant terms [HPR$^+$].

For a long time, the optimal mutation rate has not been of particular interest. By a rule of thumb the mutation rate was set to $1/n$ in many applications, i. e. every bit is flipped independently with probability $1/n$. At least for linear functions, it is known that $1/n$ is asymptotically the optimal mutation rate for the (1+1) EA on all linear functions [Wit13, DJW02]. Böttcher et al. [BDN10] showed that the optimal mutation rate for the (1+1) EA on the LeadingOnes problem is in fact $\approx 1.59/n$, way above the common practice of using $1/n$. Doerr and Goldberg [DG13] showed for the (1+1) EA that the asymptotic bound of $\Theta(n \ln n)$ holds for a mutation rate of c/n for the optimization of linear pseudo-Boolean functions, where c is an arbitrary constant. Chicano et al. [CSWA15] showed that the optimal mutation rate can be up to 50% higher than the asymptotically optimal $1/n$ for small sizes of n for the (1+1) EA on OneMax. Badkobeh et al. [BLS14] showed for the (1+λ) EA on OneMax that the optimal mutation rate increases with λ in an adaptive setting.

The exact relationship between the static mutation rate and the runtime of the (1+1) EA was revealed by Witt in the aforementioned work [Wit13] by showing that the runtime is $(1 \pm o(1))\frac{e^c}{c} n \ln n$ for general linear pseudo-Boolean functions which is asymptotically tight up to lower order terms. Here, the leading constant $\frac{e^c}{c}$ is minimized for $c = 1$, thus justifying the unwritten rule of setting the mutation rate to $1/n$ in many applications. This result was refined and extended to populations by Gießen and Witt [GW15], who gave

the asymptotically tight bound

$$(1 \pm o(1)) \left(\frac{e^c}{c} \cdot \frac{n \ln n}{\lambda} + \frac{1}{2} \cdot \frac{n \ln \ln \lambda}{\ln \lambda} \right)$$

for the number of generations needed by the (1+λ) EA on OneMax with mutation rate c/n.

The impact of the mutation rate on the lower order term remains unclear though. Since the lower order term is only known asymptotically, small problem sizes might benefit from a mutation rate that deviates from $1/n$, as seen in [CSWA15].

The goal of this paper is to find optimal mutation rates for the (1+λ) EA on OneMax for constant λ. To this end, a new drift theorem is given that provides a way of bounding the runtime if the exact drift values are known. We then give an exact formula for the drift by analyzing the distribution of the difference of two binomially distributed random variables and show how to efficiently approximate it by means of the Poisson distribution, such that the multiplicative error of the approximation is $(1 \pm O(1/n))$. By applying our new drift theorem with the approximated drift values, we are able to give approximate optimal values of c in a computationally efficient way avoiding the need for empirical investigations.

The paper is structured as follows. In Section 2 we state the (1+λ) EA and the drift theorems used throughout the paper. In particular, a new drift theorem for lower bounds is given. In Section 3 we present our main result that the lower bounds from the new drift theorem and the upper bounds on the expected runtime are only apart by a lower order term, provided the exact drift is known. In Section 4 we give an exact closed-form expression for the drift. Moreover, we show how to approximate it with only small asymptotic error in a computationally efficient way. Section 5 deals with the practical implications of our theoretical results. We show that by combining our main result with the approximated drift we can approximate the expected runtime with only small relative error. In Section 5.1 we exploit our findings by determining approximate optimal mutation rates for various settings of n and λ. Section 5.2 finally demonstrates in an empirical analysis of the expected runtime that our approximately optimal results (whose error provably can only affect a lower order term of the expected runtime) in fact very well reflect the actual runtimes.

2 Preliminaries

2.1 Algorithm

We consider the (1+λ) EA on pseudo-Boolean functions $f\colon \{0,1\}^n \to \mathbb{R}$ in the minimization version, defined as Algorithm 1. The case of $c = 1$ in the mutation probability was considered in [JJW05, DK13, DK15]. The classical (1+1) EA [AD11] is a special case of the (1+λ) EA for $\lambda = 1$ and $c = 1$.

Throughout the paper, c and λ are for simplicity assumed to be constant, i. e., they may not depend on n. With minor efforts, our results can be generalized to larger values, e. g., to $c = O(\ln n)$ and $\lambda = O(\ln n)$, at the expense of additional logarithmic factors in the error bounds.

Algorithm 1 (1+λ) EA

Select x^* uniformly at random from $\{0,1\}^n$.
for $t \leftarrow 1, 2, \ldots$ **do**
  for $i \leftarrow 1, \ldots, \lambda$ **do**
    Create x_i by flipping each bit of x^* independently with probability c/n.
  $x_m \leftarrow \arg\min_{x_i} f(x_i)$ (breaking ties randomly)
  if $f(x_m) \leq f(x^*)$ **then**
    $x^* \leftarrow x_m$

The runtime of the (1+λ) EA is defined to be the smallest $t \in \mathbb{N}$, such that an individual of minimum f-value is found. These individuals are called optima. This notion of runtime is identical with the minimum number of iterations (also called generations). Since each of these offspring has to be evaluated, the number of function evaluations, which is another classical cost measure, is by a factor of λ larger than the runtime as defined here. However, assuming a massively parallel architecture that allows for parallel evaluation of the offspring, counting the number of generations seems a valid cost measure. In particular, a speed-up on the function $\textsc{OneMax}(x_1, \ldots, x_n) := x_1 + \cdots + x_n$ by increasing λ can only be observed in terms of the number of generations.

In the rest of the paper we will focus on the minimization of the classical function $\textsc{OneMax}(x_1, \ldots, x_n) := x_1 + \cdots + x_n$, which attains its unique optimum, i. e. minimum, in the all-zero-bitstring.

2.2 Drift Theorems

Our analyses use variable drift analysis, a state-of-the art technique for the analysis of expected optimization times. We first state a theorem for upper bounds, which is well known. See [Joh10, MRC09, RS14]. We use a general version that was proposed in [LW14].

Theorem 1 (Variable Drift, Upper Bound; [LW14]) *Let $(X_t)_{t\geq 0}$ be a stochastic process adapted to a filtration $\mathcal{F}_t$ over some state space $S \subseteq \{0\} \cup [x_{\min}, x_{\max}]$, where $x_{\min} > 0$. Let $h(x)\colon [x_{\min}, x_{\max}] \to \mathbb{R}^+$ be a monotone increasing function such that $1/h(x)$ is integrable on $[x_{\min}, x_{\max}]$ and $E(X_t - X_{t+1} \mid \mathcal{F}_t) \geq h(X_t)$ if $X_t \geq x_{\min}$. Then it holds for the first hitting time $T := \min\{t \mid X_t = 0\}$ that*

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

To prove lower bounds on the hitting time by variable drift, we need additional assumptions. The first variable drift theorem for lower bounds on the hitting that we are aware of goes back to [DFW11]. It requires that the process does not make large steps towards the optimum, more precisely from state x it may only go to states $y \geq c(x)$ for some function $c(x)$. This deterministic requirement on the progress is weakened to a stochastic one in the following lemma. Moreover, the condition in item (v) is weaker than in [DFW11]. Instead of demanding $E(X_t - X_{t+1} \mid \mathcal{F}_t) \leq h(\xi(X_t))$, we allow for h-functions that are step functions. This can be useful for discrete state spaces. Somewhat simplifying, if the state space is $\mathbb{N}$ and the drift at point i equals i, we can use $h(x) := \lceil x \rceil$ in the new variant, while the original variant would require a continuous function such as $h(x) = x$. As $\int 1/\lceil z \rceil \, \mathrm{d}z \leq \int 1/z \, \mathrm{d}z$, our step function gives an improved upper bound.

Theorem 2 (Variable Drift, Lower Bound) *Let $(X_t)_{t \geq 0}$, be a stochastic process adapted to a filtration $\mathcal{F}_t$ over some state space $S \subseteq \{0\} \cup [x_{\min}, x_{\max}]$, where $x_{\min} > 0$. Suppose there exist*

(i) two functions $\xi, h \colon [x_{\min}, x_{\max}] \to \mathbb{R}^+$ such that $h(x)$ is monotone increasing and $1/h(x)$ integrable on the interval $[x_{\min}, x_{\max}]$,
(ii) $\beta > 0$,

and, using

$$g(x) := \frac{x_{\min}}{h(x_{\min})} + \int_{x_{\min}}^{x} \frac{1}{h(z)} \, \mathrm{d}z,$$

suppose it holds for all $t \geq 0$ that

(iii) $X_{t+1} \leq X_t$,
(iv) $\Pr(X_{t+1} < \xi(X_t)) \leq \frac{1}{\beta g(X_t)}$ for $X_t \geq x_{\min}$,
(v) $E(X_t - X_{t+1} \mid \mathcal{F}_t) \leq \lim_{\delta \downarrow 0} h(\xi(X_t) + \delta)$ for $X_t \geq x_{\min}$.

Then it holds for the first hitting time $T := \min\{t \mid X_t = 0\}$ that

$$E(T \mid X_0) \geq \frac{\beta}{1+\beta} g(X_0).$$

Corollary to the above: *If ξ is invertible and differentiable on the domain, with inverse function ξ^{-1} and derivative ξ', and (iv) and (v) are replaced by the conditions*

(iv') $\Pr(X_{t+1} < \xi(X_t)) \leq \frac{1}{\beta\left(\frac{x_{\min}}{h(\xi^{-1}(x_{\min}))} + \int_{\xi^{-1}(x_{\min})}^{\xi^{-1}(X_t)} \frac{\xi'(x)}{h(x)} \, \mathrm{d}x\right)}$ *for $X_t \geq x_{\min}$,*
(v') $E(X_t - X_{t+1} \mid \mathcal{F}_t) \leq \lim_{\delta \downarrow 0} h(X_t + \delta)$ *for $X_t \geq x_{\min}$*

then the bound on $E(T \mid X_0)$ is equivalent to

$$E(T \mid X_0) \geq \frac{\beta}{1+\beta} \left(\frac{x_{\min}}{h(\xi^{-1}(x_{\min}))} + \int_{\xi^{-1}(x_{\min})}^{\xi^{-1}(X_0)} \frac{\xi'(x)}{h(x)} \, \mathrm{d}x \right).$$

Proof. We use the potential function $g(x) = \frac{x_{\min}}{h(x_{\min})} + \int_{x_{\min}}^{x} \frac{1}{h(z)}\,\mathrm{d}z$ to apply additive drift analysis w. r. t. to the drift $E(g(X_t) - g(X_{t+1}) \mid \mathcal{F}_t)$. Note that the first bound on $E(T \mid X_0)$ given in the theorem follows if we can establish the bound on the drift

$$E(g(X_t) - g(X_{t+1}) \mid \mathcal{F}_t) \leq 1 + \frac{1}{\beta}$$

for $X_t \geq x_{\min}$ since $g(X_0) = \left(\frac{x_{\min}}{h(x_{\min})} + \int_{x_{\min}}^{X_0} \frac{1}{h(x)}\,\mathrm{d}x\right)$. The second bound on $E(T \mid X_0)$ is equivalent to the first one as can be shown by a simple substitution.

We are left with the bound on the drift. By the law of total probability,

$$\begin{aligned}
&E(g(X_t) - g(X_{t+1}) \mid \mathcal{F}_t)\\
&= E(g(X_t) - g(X_{t+1}) \mid \mathcal{F}_t; X_{t+1} \geq \xi(X_t)) \Pr(X_{t+1} \geq \xi(X_t))\\
&+ E(g(X_t) - g(X_{t+1}) \mid \mathcal{F}_t; X_{t+1} < \xi(X_t)) \Pr(X_{t+1} < \xi(X_t))\\
&\leq E(g(X_t) - g(X_{t+1}) \mid \mathcal{F}_t; X_{t+1} \geq \xi(X_t))\\
&\qquad + \; g(X_t) \cdot \Pr(X_{t+1} < \xi(X_t))\\
&\leq E(g(X_t) - g(X_{t+1}) \mid \mathcal{F}_t; X_{t+1} \geq \xi(X_t))\\
&\qquad + \; g(X_t) \cdot \frac{1}{\beta g(X_t)}\\
&= E(g(X_t) - g(X_{t+1}) \mid \mathcal{F}_t; X_{t+1} \geq \xi(X_t)) \; + \; \frac{1}{\beta},
\end{aligned}$$

where the first inequality estimated the drift in the case $X_{t+1} < \xi(X_t)$ by $g(X_t)$ and the second one used the assumption (iv).

We proceed by bounding

$$\begin{aligned}
&E(g(X_t) - g(X_{t+1}) \mid \mathcal{F}_t; X_{t+1} \geq \xi(X_t))\\
&\leq E(\int_{X_{t+1}}^{X_t} \frac{1}{h(x)}\,\mathrm{d(x)} \mid \mathcal{F}_t; X_{t+1} \geq \xi(X_t))\\
&\leq \frac{E(X_t - X_{t+1} \mid \mathcal{F}_t; X_{t+1} \geq \xi(X_t))}{\lim_{\delta \downarrow 0} h(\xi(X_t) + \delta)}\\
&\leq \frac{E(X_t - X_{t+1} \mid \mathcal{F}_t)}{\lim_{\delta \downarrow 0} h(\xi(X_t) + \delta)},
\end{aligned}$$

where the first inequality follows by expanding g, the second one used that $X_t \geq X_{t+1} \geq \xi(X_t)$ (which uses (iii)) along with the fact that h is monotone increasing (i. e., non-decreasing) (from (i)), and the third one the definition of conditional probability. Using (v), the last bound simplifies to

$$E(g(X_t) - g(X_{t+1}) \mid \mathcal{F}_t) \leq \frac{\lim_{\delta \downarrow 0} h(\xi(X_t) + \delta)}{\lim_{\delta \downarrow 0} h(\xi(X_t) + \delta)} = 1,$$

so that

$$E(g(X_t) - g(X_{t+1}) \mid \mathcal{F}_t) \leq 1 + \frac{1}{\beta}$$

as desired. □

3 Bringing together Lower Bounds and Upper Bounds from Variable Drift

We are interested in how far the lower bound from Theorem 2 and the upper bound from Theorem 1 are apart when an exact expression on the drift $E(X_t - X_{t+1})$ is known. Intuitively, this depends to a great extent on how sharply the progress is concentrated around its expected value. If large jumps towards the optimum are sufficiently likely, then the lower bound seems to reflect the truth better. For example, consider the following artificial process: the fitness function is OneMax and the algorithm is the (1+1) EA with the following modified mutation step: with probability $1/n$ the optimum is created, and with probability $1 - 1/n$ the usual standard-bit mutation operator with mutation probability $1/n$ is used. At distance i from the optimum, the drift towards the optimum is $\Theta(i/n)$, resulting in an upper bound $O(n \log n)$ on the expected runtime according to Theorem 1. However, it is easy to see that the actual expected runtime is $O(n)$. This is a consequence of the possibly large jumps directly into the optimum, which happen in every step with probability $1/n$. Note that such steps are very unlikely with the unmodified mutation operator unless the algorithm is very close to the optimum anyway.

We concentrate now on the (1+λ) EA on OneMax, assuming constant c in the mutation probability and constant λ. Then the following theorem shows using the right drift function, upper and lower bounds on the expected runtime are only apart by a term of lower order, which is indeed considerably lower than the expected time on OneMax.

Theorem 3 *Consider the (1+λ) EA on* OneMax*, choosing c and λ as constants. Denoting by X_t the number of ones in the search point at time $t \geq 0$, the drift $\Delta(i) := E(X_t - X_{t+1} \mid X_t = i)$ for $i \in \{1, \dots, n\}$ is defined. Then it holds for the expected runtime that*

$$(1 - O(n^{-1/3}(\ln n))) \cdot I(X_0) \;\leq\; E(T \mid X_0) \;\leq\; I(X_0),$$

where $I(X_0) = \sum_{i=1}^{X_0} 1/\Delta(i)$.

For its proof, we need a helper lemma, which is concerned with the monotonicity of the drift.

Lemma 4 *Let $\Delta(i)$ be defined as in Theorem 3. Then $\Delta(i+1) \geq \Delta(i)$ for any $i \geq 0$.*

Proof. We first prove the result for $\lambda = 1$ and show then how to extend it to $\lambda > 1$.

Assume search point $x(i)$ with i one-bits at time t and let $x'(i)$ be its random offspring (before selection). We represent

$$\Delta(i) = \sum_{j \geq 0} \Pr(|x(i)| - |x'(i)| \geq j) = \sum_{j \geq i} \Pr(|x'(i)| \leq j),$$

where $|z|$ denotes the number of one-bits in z. If we can prove for all $j \leq i$ the inequality

$$\Pr(|x'(i+1)| \leq j+1) \geq \Pr(|x'(i)| \leq j) \tag{1}$$

the lemma follows. The inequality will be shown by a coupling argument; formally, we map mutations $x'(i)$ of $x(i)$ to mutations $x'(i+1)$ of $x(i+1)$ and show that the mapped mutations are at least as likely. By the symmetry of the mutation operator, we can assume that $x(i+1)$ is bitwise non-less than $x(i)$. Let i^* be the bit that is 1 in $x(i+1)$ but 0 in $x(i)$. We first consider all mutations $x'(i)$ of $x(i)$ with at most $j \leq i$ one-bits in which bit i^* is not flipped. These mutations map one-to-one to mutations $x'(i+1)$ with at most $j+1$ one-bits by flipping the same bits in $x(i+1)$.

The mutations of $x(i)$ where bit i^* is flipped to 1 must also flip another bit k to 0 as $j \leq i$. We map them to mutations of $x(i+1)$ by flipping all bits in the same way, except for that neither i^* nor k are flipped. Such a mutation has one further one-bit. Since we use mutation probability $c/n = \Theta(1/n)$, the probability of not flipping i^* and k is by a factor of

$$\left(\frac{1-c/n}{c/n}\right)^2 = \Theta(n^2)$$

larger than the probability of flipping both of them. Note that the mapping is not bijective in this case as up to $n-1$ different mutations (one for each value of k) of $x(i)$ are mapped to the same mutation of $x(i+1)$. Still, by a union bound, the probability of generating the considered mutation of $x(i+1)$ is by a factor of $\Theta(n)$ larger, proving (1).

If $\lambda > 1$, we represent the drift as a sum of tail probabilities in the same way as above, except for that the best of the λ offspring of $x(i)$ is considered instead of a single offspring $x'(i)$. By independence of the offspring creation, the probability that the best offspring improves by at least j equals $1-(\Pr(|x(i)|-|x'(i)|<j))^\lambda$. Using (1), this is at most $1-(\Pr(|x(i+1)|-|x'(i+1)|<j))^\lambda$, which completes the proof. □

Proof of Theorem 3. To prove the upper bound, we use the variable drift theorem for upper bounds (Theorem 1), using $x_{\min} = 1$, $x_{\max} = n$, and $h(x) = \Delta(\lceil x \rceil)$ for $x \in [x_{\min}, x_{\max}]$. Note that the theorem allows such a discontinuous h-function. Hence, we obtain

$$\begin{aligned} E(T \mid X_0) \;&\leq\; \frac{1}{\Delta(1)} + \int_{x_{\min}}^{X_0} \frac{\mathrm{d}x}{\Delta(\lceil x \rceil)} \;=\; \frac{1}{\Delta(1)} + \lim_{\ell \downarrow x_{\min}} \int_{\ell}^{X_0} \frac{\mathrm{d}x}{\Delta(\lceil x \rceil)} \\ &=\; \sum_{i=1}^{X_0} \frac{1}{\Delta(i)} \;=\; I_0. \end{aligned}$$

For the lower bound, we define

$$\xi(x) := \begin{cases} \lceil x \rceil - 1 & \text{if } x \leq n^{1/3}, \\ \lceil x \rceil - \lceil \ln(n) \rceil & \text{otherwise.} \end{cases}$$

To analyze the probability that $X_{t+1} < \xi(X_t)$, given some $X_t > 0$, we also consider the two cases. If $X_t \leq n^{1/3}$, then it is necessary to flip at least two one-bits to zero to obtain $X_{t+1} < \xi(X_t) = X_t - 1$. This probability is at most $(n^{1/3}\frac{c}{n})^2 = O(n^{-4/3})$, and this asymptotic bound even holds if the best of $\lambda = O(1)$ offspring is considered. The probability of flipping at least $\ln n$ bits in at least one of $\lambda = O(1)$ offspring is clearly smaller, more precisely $n^{-\Omega(\ln n)}$. Hence, $P(X_{t+1} < \xi(X_t)) \leq O(n^{-4/3})$. Furthermore, we get $h(x) \geq (1-c/n)^{n-1}cx/n \geq e^{-2c}cx/n$ by considering the expected number of one-bits flipped in an arbitrary offspring, conditioned on that no zero-bit flips. Hence,

$$\left(\frac{x_{\min}}{h(x_{\min})} + \int_{x_{\min}}^{X_t} \frac{1}{h(x)}\,\mathrm{d}x\right) \leq O(n) + \int_1^n \frac{ne^{2c}}{c\lceil x\rceil}\ ,$$

which is $O(n \ln n)$ as c is constant. This establishes Condition (iv) of Theorem 2 for $\beta = c'n^{1/3}/(\ln n)$, where c' is a sufficiently large constant. We also note that a union bound over $\lambda = O(1)$ offspring implies $h(x_{\min}) = O(\lambda c/n) = O(1/n)$, hence

$$I_0 \geq \frac{x_{\min}}{h(x_{\min})} = \Omega(n),$$

which we will use later.

Condition (iii) of Theorem 2 holds trivially due to the selection mechanism of the (1+λ) EA. To verify the remaining conditions, we set $h(x) := \Delta(\lceil x\rceil)$ for $x \leq n^{1/3}$ and $h(x) := \Delta(\lceil x\rceil + \lceil \ln(n)\rceil)$ otherwise. Note that $\lim_{\delta\downarrow 0} h(\xi(x)+\delta) = \Delta(x)$ if $x \leq n^{1/3}$. Otherwise,

$$\begin{aligned} E(X_t - X_{t+1} \mid \mathcal{F}_t; X_t = x) &= \Delta(\lceil x\rceil) \leq \lim_{\delta\downarrow 0} \Delta(\lceil x\rceil + \delta) \\ &= \lim_{\delta\downarrow 0} h(\lceil x\rceil - \lceil \ln(n)\rceil + \delta) = \lim_{\delta\downarrow 0} h(\xi(\lceil x\rceil) + \delta) \end{aligned}$$

since by Lemma 4 $\Delta(i)$ is monotone increasing in its argument. This establishes (v). Also (i) is satisfied by definition of h. Altogether, the drift theorem yields

$$\begin{aligned} E(T \mid X_0) &\geq \frac{\beta}{1+\beta}\left(\frac{1}{\Delta(1)} + \int_1^{X_0} \frac{1}{h(x)}\,\mathrm{d}x\right) \\ &= \frac{\beta}{1+\beta}\left(\sum_{i=1}^{n^{1/3}} \frac{1}{\Delta(i)} + \int_{n^{1/3}}^{X_0} \frac{1}{h(x)}\,\mathrm{d}x\right) \\ &\geq \frac{\beta}{1+\beta}\left(\sum_{i=1}^{n^{1/3}} \frac{1}{\Delta(i)} + \sum_{n^{1/3}+\lceil \ln n\rceil}^{X_0+\lceil \ln n\rceil} \frac{1}{\Delta(i)}\right), \end{aligned}$$

where the equality used that $h(x) := \Delta(\lceil x\rceil)$ for $x \leq n^{1/3}$ and the inequality used the definition of h in the other case along with an index transformation.

We see that the term in parentheses is at least

$$\sum_{i=1}^{X_0} \frac{1}{\Delta(i)} - \sum_{i=1}^{\lceil \ln n \rceil} \frac{1}{\Delta(n^{1/3}+i)} \geq I_0 - \lceil \ln n \rceil \frac{1}{\Delta(n^{1/3})},$$

by the monotonicity of Δ. We also know that $\Delta(n^{1/3}) \geq n^{1/3}\frac{c}{n}(1-c/n)^{n-1} = \Omega(n^{-2/3})$ for any constant choice of c and any $\lambda \geq 1$. Hence

$$\lceil \ln n \rceil \frac{1}{\Delta(n^{1/3})} = O(n^{2/3}(\ln n)) = O(n^{-1/3}(\ln n) I_0),$$

using $I_0 = \Omega(n)$. This altogether proves the desired inequality

$$E(T \mid X_0) \geq (1 - O(n^{-1/3}(\ln n)))I_0. \quad \square$$

4 Approximating the Drift

In the following we will give a closed-form exact formula for the drift at fitness k of the $(1+\lambda)$ EA with mutation rate c/n for constant c and constant λ. Let X_t denote the fitness at time t. Then, $X_{t+1} = X_t + Z$, where Z is the difference of two binomially distributed random variables, namely the number of 1-bits flipped by the mutation and the number of 0-bits flipped by the mutation. If more 1-bits than 0-bits are flipped, i. e. if Z is negative, the algorithm progresses towards the optimum. Hence, we are interested in the exact distribution of the random variable Z.

To this end, we will make use of the ordinary hypergeometric function, also known as Gaussian hypergeometric function ${}_2F_1$. Let for each $a \in \mathbb{Z}$ and $b \in \mathbb{N}$ denote $a^{(b)} = a(a+1)(a+2)\cdots(a+b-1)$ the rising factorial, which is sometimes called the Pochhammer symbol. The hypergeometric function ${}_2F_1$ is defined for $x \in \mathbb{R}$ as the following series:

$${}_2F_1(a,b;c;x) = {}_2F_1\left(\begin{matrix} a, & b \\ & c \end{matrix}; x\right) := \sum_{i=0}^{\infty} \frac{a^{(i)}b^{(i)}}{c^{(i)}} x^i \ .$$

Note that the infinite series terminates if either $a < 0$ or $b < 0$. We refer the reader to [AS64] for more information.

Let $X \sim \mathrm{Bin}(n_1, p_1)$, $Y \sim \mathrm{Bin}(n_2, p_2)$ and let $Z := X - Y$. In the following, we will denote the distribution of Z as $Z \sim \mathrm{BinDiff}(n_1, n_2, p_1, p_2)$.

Theorem 5 *Let $Z \sim \mathrm{BinDiff}(n_1, n_2, p_1, p_2)$. Then, for all $z \in \mathbb{N}$:*

$$\Pr(Z = z) = \begin{cases} p_1^z (1-p_1)^{n_1 - z}(1-p_2)^{n_2} \binom{n_1}{z} \varphi_1, & z \geq 0 \\ p_2^{-z}(1-p_2)^{n_1+z}(1-p_1)^{n_1}\binom{n_2}{-z}\varphi_2 & z < 0 \end{cases},$$

where

$$\varphi_1 = {}_2F_1\left(\begin{matrix}-n_2,\ z-n_1\\ z+1\end{matrix}; \frac{p_1p_2}{(1-p_1)(1-p_2)}\right) ,$$

and $$\varphi_2 = {}_2F_1\left(\begin{matrix}-n_1,\ -(z+n_2)\\ 1-z\end{matrix}; \frac{p_1p_2}{(1-p_1)(1-p_2)}\right) .$$

Proof. Let X, Y and Z be defined as stated in the theorem. We have for all $z \in \mathbb{Z}$

$$\begin{aligned}\Pr(Z=z) &= \Pr(X-Y=z)\\ &= \sum_{\substack{a,b\in\mathbb{Z}\\ a-b=z}} \Pr(X=a)\Pr(X=b) \ . \end{aligned} \tag{2}$$

Furthermore, let $q = p_1p_2/((1-p_1)(1-p_2))$ and $\alpha = p_1^z(1-p_1)^{n_1-z}(1-p_2)^{n_2}$. Consider the case $z \geq 0$. Rewriting the sum from above we get

$$\begin{aligned}&\Pr(Z=z) = \sum_{k=0}^{n_1} \Pr(X=k+z)\Pr(Y=k)\\ &= \sum_{k=0}^{n_1} \binom{n_1}{k+z} p_1^{k+z}(1-p_1)^{n_1-k-z}\binom{n_2}{k}p_2^k(1-p_2)^{n_2-k}\\ &= \alpha\sum_{k=0}^{n_1} \frac{n_1!}{(k+z)!(n_1-k-z)!}\cdot\frac{n_2!}{k!(n_2-k)!}\cdot q^k\\ &= \alpha\sum_{k=0}^{n_1} \frac{(n_1-z-k+1)^{(z+k)}}{z!(z+1)^{(k)}}\cdot\frac{(n_2-k+1)^{(k)}}{k!}\cdot q^k\\ &= \alpha\sum_{k=0}^{n_1} \frac{(-1)^{2k}(-(n_1-z))^{(k)}(n_1-z+1)^{(z)}(-n_2)^{(k)}}{z!(z+1)^{(k)}k!}\cdot q^k \ . \end{aligned}$$

Using the fact that $(n_1-z+1)^{(z)}/z! = \binom{n_1}{z}$ we get

$$\begin{aligned}\Pr(Z=z) &= \alpha\binom{n_1}{z}\sum_{k=0}^{n_1}\frac{(z-n_1)^{(k)}(-n_2)^{(k)}}{(z+1)^{(k)}}\cdot\frac{q^k}{k!}\\ &= \alpha\binom{n_1}{z}{}_2F_1\left(\begin{matrix}-n_2,\ z-n_1\\ z+1\end{matrix};q\right) \ . \end{aligned}$$

The case $z < 0$ can be shown analogously. □

In the previous theorem we have seen that the discrete probability mass function of a BinDiff-distributed random variable can be expressed in terms of hypergeometric functions. The exact terms can also be written as Jacobi polynomials, a class of orthogonal polynomials. This relationship to orthogonal polynomials has been elaborated in connection with the probability distribution of fitness values of a bit string undergoing uniform bit-flip mutation

in [CSWA15]. In that work, another class of orthogonal polynomials, namely Krawtchouk polynomials, were investigated.

In the following we will show that the drift can be computed efficiently. For this purpose, we will assume that the computation of ${}_2F_1$ counts as a single arithmetic operation using some implementation of ${}_2F_1$, similar to counting other special functions like exp.

Corollary 6 *Consider the (1+λ) EA, choosing c and λ constant on* OneMax*. The drift $\Delta(k)$ at fitness k can be computed exactly with $O(k)$ arithmetic operations.*

Proof. Consider a run of the algorithm as stated above and assume that the OneMax-value is k. For all $i \in [\lambda]$ let $X_i \sim \mathrm{Bin}(n-k, c/n)$ and $Y_i \sim \mathrm{Bin}(k, c/n)$, then $Z_i := X_i - Y_i$ is the random variable that denotes the change in fitness of the i-th offspring individual, i. e. $k + \min\{Z_i | i \in [\lambda]\}$ is the fitness of the new parent. Note that the image of each Z_i is $\{z \in \mathbb{Z} | -k \leq z \leq n-k\}$.

Using Theorem 5 we can write the cumulative distribution function of Z_i as $F_{Z_i}(x) = \sum_{y=-k}^{x} \Pr(Z_i = y)$. Let Z^* be defined as the minimum of $Z_1, \ldots, Z_\lambda$, i. e. Z^* is the minimum order statistic. It is known that $F_{Z^*}(z) = 1 - (1 - F_{Z_1}(z))^\lambda$, using Z_1 as an arbitrary representative among all Z_i since all Z_i are independent. Hence, we get that

$$\Pr(Z^* = z) = (1 - F_{Z_1}(z-1))^\lambda - (1 - F_{Z_1}(z))^\lambda \ .$$

Consider the drift at fitness k which is defined as

$$\Delta(k) := E(X_t - X_{t+1} \mid X_t = k).$$

We get

$$\begin{aligned}
\Delta(k) &= -\sum_{z=-k}^{-1} z \Pr(Z^* = z) \\
&= -\sum_{z=-k}^{-1} z \left((1 - F_{Z_1}(z-1))^\lambda - (1 - F_{Z_1}(z))^\lambda \right) \\
&= -\sum_{z=-k}^{-1} z \left(\left(1 - \sum_{y=-k}^{z-1} \Pr(Z_1 = y)\right)^\lambda \right. \\
&\qquad \left. - \left(1 - \sum_{y=-k}^{z} \Pr(Z_1 = y)\right)^\lambda \right) \ .
\end{aligned}$$

The last term involves two nested sums. However, we only need to evaluate $\Pr(Z_1 = y)$ for $y = -k, \ldots, -1$ once, saving all values in a list, in order to compute $F_{Z_1}(z)$ for $y = -k, \ldots, -1$ which requires a single traversal over that list. Computing the actual drift only consists of $O(k)$ arithmetic operations now, since we precomputed the needed values of F_{Z_1}. Hence, the computation of $\Delta(k)$ only needs $O(k)$ arithmetic operations and space $O(k)$. □

As we have seen in Corollary 6, we need $O(k)$ arithmetic computations of the probability mass function of a BinDiff-distributed random variable (see Theorem 5). The computation of the hypergeometric function ${}_2F_1$ proves to be the bottleneck. For example, on an Intel i7-4770S CPU, a single computation of the value of ${}_2F_1(-750, -249; 2; 4.016 \cdot 10^{-6})$, which is a representative invocation of the function in this context, takes around 0.02 seconds using the internal function `hypergeom` from Matlab R2015b. Different parameter settings yield similar computation times. In order to compute the expected runtime of the (1+λ) EA, we need to compute a linear amount of drift values, which can pose a problem for large values of n due to the computational effort on the hypergeometric function. Hence, we are interested in approximating the drift with less computational effort.

The idea is to approximate the BinDiff-distribution. It is well-known that a $\mathrm{Poi}(np)$-distribution yields a good approximation for a $\mathrm{Bin}(n, p)$-distribution for large n and small p. To this end, we will approximate a BinDiff-distribution with the distribution of the difference of the corresponding Poisson approximations. A similar approach to the approximation of the point-wise drift has been pursued by Doerr, Doerr and Yang ([DDY16]). The distribution of the difference of two independent Poisson-distributed random variables is known in the literature as Skellam-distribution. We will state the definition according to [Ske46].

Definition 7 *Let $X \sim \mathrm{Poi}(\mu_1)$ and $Y \sim \mathrm{Poi}(\mu_2)$. The probability mass function of $Z := X - Y$ is given by*

$$\Pr(Z = k) = e^{-(\mu_1+\mu_2)} \left(\frac{\mu_1}{\mu_2}\right)^{\frac{k}{2}} I_k(2\sqrt{\mu_1\mu_2}) \ ,$$

where I_k is the modified Bessel function of the first kind, defined by

$$I_\nu(z) = \left(\frac{z}{2}\right)^\nu \sum_{i=0}^{\infty} \frac{(z/2)^{2i}}{i!(\nu+i)!} \ ,$$

for $\nu \in \mathbb{N}_0$ and $z \in \mathbb{R}$. In the following we will denote the distribution of Z by $Z \sim \mathrm{Skellam}(\mu_1, \mu_2)$.

For more information about the modified Bessel function of the first kind, we refer the reader to [AS64].

Instead of the hypergeometric function in the probability mass function of a BinDiff-distributed random variable, a Skellam-distributed random variable involves the modified Bessel function of the first kind, which is another special function. However, computing for example $I_{250}(1.73)$, takes less than 0.0001 seconds on an Intel i7-4770S CPU using the internal function `besseli` from Matlab R2015b. The computation times are similar for different parameters. This is a big improvement over the time needed for the computation of the hypergeometric function. Computationally, the use of the Skellam-distribution instead of the BinDiff-distribution is therefore justified. In the following we will show that the error from the approximation is small as well. Again, we will assume that the computation of I_ν counts as a single arithmetic operation.

Theorem 8 *Consider the (1+λ) EA with mutation probability c/n and population size λ on* OneMax *where c and λ are constant. Then, the drift $\Delta(k)$ can be approximated up to a factor of $1 \pm O(1/n)$ in $O(k)$ arithmetic operations.*

Proof. We will first bound the relative error of the approximation of a BinDiff-distributed random variable by a Skellam-distributed random variable. Let $X \sim \mathrm{Bin}(n-k, c/n)$ and $Y \sim \mathrm{Bin}(k, c/n)$, then $Z := X - Y \sim \mathrm{BinDiff}(n-k, k, c/n, c/n)$. Let $\tilde{X} \sim \mathrm{Poi}((n-k)c/n)$ and $\tilde{Y} \sim \mathrm{Poi}(kc/n)$ be the according Poisson-approximated random variables. Then, $\tilde{Z} := \tilde{X} - \tilde{Y} \sim \mathrm{Skellam}((n-k)c/n, kc/n)$. The relative error of the Poisson-approximation with respect to X is

$$\left| \frac{\Pr(\tilde{X} = m)}{\Pr(X = m)} - 1 \right| \leq \frac{(e^{(n-k)c/n} - 1)\frac{c}{n}}{m+1} \ .$$

Bounding the exponent by c and omitting the constants, we have $\Pr(\tilde{X} = m) = \Pr(X = m)(1 \pm O(1/(mn)))$ and $\Pr(\tilde{Y} = m) = \Pr(Y = m)(1 \pm O(1/(mn)))$ accordingly (see [Tee07] for details about the relative error bounds).

One can obtain different bounds for the relative error, depending on the actual arithmetic computation of $\Pr(\tilde{Z} = z)$, For example, we have that

$$1 - F_{\tilde{Z}}(z) = 1 - \sum_{\ell=-k}^{z} \Pr(\tilde{Z} = \ell) = \sum_{\ell=z+1}^{n-k} \Pr(\tilde{Z} = \ell) \ .$$

For small z, the second term involves the computation of fewer probabilities than the third term. This can lead to different results on the error bound as a consequence of error propagation. However, due to the triangle inequality we have that the absolute error

$$|\Pr(Z = z) - \Pr(\tilde{Z} = z)| \leq |\Pr(Z = z) - q| + |q - \Pr(\tilde{Z} = z)| \ ,$$

and the last term is 0 for any arithmetic computation q of $\Pr(\tilde{Z} = z)$. Thus, we can compute $\Pr(\tilde{Z} = z)$ in the same way as $\Pr(Z = z)$ in Equation 2 in order to derive a bound on the relative error of $\tilde{Z}$ with respect to Z, instead of using the exact probability mass function. We have for $z \geq 0$ that

$$\begin{aligned}
\Pr(\tilde{Z} = z) &= \sum_{j=0}^{n-z} \Bigg(\Pr(X = j+z) \Pr(Y = z) \\
&\qquad \cdot \left(1 \pm O\left(\frac{1}{(j+z)n}\right)\right)\left(1 \pm O\left(\frac{1}{zn}\right)\right) \Bigg) \\
&= (1 \pm O(1/n)) \sum_{j=0}^{n-z} \Pr(X = j+z) \Pr(Y = z) \\
&= (1 \pm O(1/n)) \Pr(Z = z) \ .
\end{aligned}$$

The case $z < 0$ is analogous and we obtain in total

$$\Pr(\tilde{Z} = z) = (1 \pm O(1/n)) \Pr(Z = z) \ ,$$

for all $z \in \{-k, -k+1, \ldots, n-k\}$.

We can now compute the relative error of the approximated drift $\tilde{\Delta}$. By replacing Z_1 with $\tilde{Z}$ in the computation of the exact drift from Corollary 6 we obtain

$$\begin{aligned}\tilde{\Delta}(k) &= -\sum_{z=-k}^{-1} z \left(\left(1 - \sum_{y=-k}^{z-1} \Pr(\tilde{Z}=y)\right)^{\lambda} \right. \\ &\qquad \left. - \left(1 - \sum_{y=-k}^{z} \Pr(\tilde{Z}=y)\right)^{\lambda} \right) \\ &= -(1 \pm O(1/n))^{\lambda} \sum_{z=-k}^{-1} z \left(\left(\sum_{y=z}^{n-k} \Pr(Z=y)\right)^{\lambda} \right. \\ &\qquad \left. - \left(\sum_{y=z+1}^{n-k} \Pr(Z=y)\right)^{\lambda} \right) \\ &= (1 \pm O(1/n))\Delta(k) \ .\end{aligned}$$

Note that in the last step we exploited the fact that λ is a constant.

Since we can use the same arithmetic computation as in Corollary 6, the number of arithmetic operations in order to compute the approximated drift is $O(k)$ as well. □

5 Computation of Mutation Rates

In the previous section we showed how to approximate the drift with only a small relative error. Using the new drift theorem from Section 3 we are interested in approximating the expected runtime.

Corollary 9 *Consider the (1+λ) EA on* OneMax *with mutation probability c/n, where c and λ are constant. Using the approximation from Section 4 the runtime of the (1+λ) EA can be approximated up to a multiplicative error of $1 \pm O(n^{-1/3}(\ln n))$.*

Proof. Let X_t denote the OneMax-value at time $t > 0$ and let furthermore $\Delta(i) := E(X_t - X_{t+1} \mid X_t = i)$ be the drift at fitness value i.

Theorem 3 states that

$$(1 - O(n^{-1/3}(\ln n))) \cdot I(X_0) \ \leq \ E(T \mid X_0) \ \leq \ I(X_0) \ ,$$

where $I(X_0) = \sum_{i=1}^{X_0} 1/\Delta(i)$.

Plugging in the approximate drift values $\tilde{\Delta}(i)$, as described in Section 4 and by weakening the upper bound we obtain

$$E(T \mid X_0) = (1 \pm O(n^{-1/3}(\ln n))) \sum_{i=1}^{X_0} 1/\tilde{\Delta}(i) \ .$$

Table 1 Approximate optimal c values

	Problem size n					
λ	100	200	500	1000	2000	5000
1	1.19	1.16	1.13	1.11	1.10	1.09
2	1.19	1.16	1.13	1.11	1.10	1.09
3	1.19	1.16	1.13	1.12	1.10	1.09
4	1.20	1.17	1.14	1.12	1.11	1.09
5	1.21	1.17	1.14	1.12	1.11	1.10
6	1.21	1.18	1.14	1.13	1.11	1.10
7	1.22	1.18	1.15	1.13	1.12	1.10
8	1.23	1.19	1.15	1.13	1.12	1.10
9	1.23	1.19	1.16	1.14	1.12	1.11
10	1.24	1.20	1.16	1.14	1.12	1.11

Using a Chernoff bound we can see that the probability that the algorithm initializes in the interval $[n/2 - n^{2/3}, n/2 + n^{2/3}]$ is at least $1 - 2e^{-(2/3)n^{1/3}}$. The expected time to advance from a fitness of $\lceil n/2 + n^{2/3} \rceil$ down to $\lfloor n/2 \rfloor$ is $\sum_{\lfloor n/2 \rfloor}^{\lceil n/2+n^{2/3} \rceil} 1/\tilde{\Delta}(i) \leq n^{2/3}/\tilde{\Delta}(\lfloor n/2 + n^{2/3} \rfloor)$ due to the monotonicity of $\tilde{\Delta}$ by Lemma 4. It holds that $\tilde{\Delta}(\lfloor n/2 + n^{2/3} \rfloor) \leq (n/2 + n^{2/3})(c/n) = O(1)$ and thus, we get $\sum_{\lfloor n/2 \rfloor}^{\lceil n/2+n^{2/3} \rceil} 1/\tilde{\Delta}(i) = \Omega(n^{2/3})$. Similarly, we can show that $\sum_{\lfloor n/2-n^{2/3} \rfloor}^{\lceil n/2 \rceil} 1/\tilde{\Delta}(i) = O(n^{2/3})$. Furthermore, we know that the expected runtime is $\Theta(n \ln n)$, independent from the actual initialization in the given interval. Therefore, the relative error by deviating at most $n^{2/3}$ from $n/2$ is of order $\Theta(n^{2/3}/E(T)) = \Theta((\ln n)n^{-1/3})$, which matches the asymptotic factor.

Conditioning on the event that the algorithm initializes in $[n/2-n^{2/3}, n/2+n^{2/3}]$ yields in total:

$$E(T) = (1 \pm O(n^{-1/3}(\ln n)) \sum_{i=1}^{\lceil n/2 \rceil} 1/\tilde{\Delta}(i) \ , \tag{3}$$

where the error introduced by the Chernoff bound is absorbed by the asymptotic factor. Hence, we can approximate the runtime of the $(1+\lambda)$ EA with only small asymptotic error. □

5.1 Approximating the runtime

We implemented the sum from the right-hand side of Equation 3 in order to compute the approximate expected runtime in Matlab R2015b and computed the approximate expected runtimes for $c = 1.00, 1.01, \ldots, 2.0$ and $\lambda = 1, \ldots, 10$ and problem sizes $n = 100, 200, 500, 1000, 2000, 5000$. The approximated optimal values of c are given in Table 1.

As expected, we can see that for fixed λ, the approximated mutation rate parameter c is decreasing with n, approaching 1 as predicted by the tight analysis in [GW15]. Furthermore, the approximate optimal c for $n = 100$ and $\lambda = 1$ is 1.19, which is only slightly higher than the exact value of 1.17 given in

Table 2 Ratios of the expected runtimes for the approximate optimal mutation rate and the standard mutation rate $1/n$

	Problem size n					
λ	100	200	500	1000	2000	5000
1	0.9865	0.9901	0.9931	0.9946	0.9957	0.9967
2	0.9861	0.9899	0.9929	0.9945	0.9955	0.9965
3	0.9858	0.9896	0.9927	0.9943	0.9954	0.9964
4	0.9854	0.9893	0.9925	0.9941	0.9952	0.9963
5	0.9850	0.9890	0.9923	0.9939	0.9951	0.9962
6	0.9845	0.9886	0.9920	0.9937	0.9949	0.9960
7	0.9840	0.9882	0.9917	0.9935	0.9947	0.9959
8	0.9835	0.9878	0.9914	0.9932	0.9945	0.9957
9	0.9830	0.9874	0.9911	0.9930	0.9943	0.9956
10	0.9824	0.9869	0.9908	0.9927	0.9941	0.9954

[CSWA15], hence justifying that the approximation does not suffer from large constants in the lower order term that might corrupt the approximation for small values of n.

Interestingly, for fixed n the approximate optimal mutation rate grows with λ. However, this behaviour cannot be explained by the bound from [GW15] which means that the reason for this behaviour is hidden in the lower order term. Intuitively, employing a larger population stabilizes the explorative character of allowing a higher mutation rate by reducing the chance that none of the individuals makes progress at all.

In order to evaluate the benefit of setting the mutation rate to the approximate optimal value instead of using the mutation rate $1/n$, we provide the corresponding table of the ratios of the approximated expected runtimes for the optimal mutation rate and $1/n$. As can be seen in Table 2, using the optimal mutation rate compared to $1/n$ does not improve the corresponding expected runtime by more than 2% in the considered ranges of n and λ. This means that a mutation rate of $1/n$ is a sane choice for the parameter ranges that we have examined.

For fixed λ, the ratios increase with n, as expected. Interestingly, for fixed n, the ratios decrease with λ, which means that using the optimal mutation rate has a greater influence on larger population sizes. Another observation from Figure 1 is that for fixed n and λ the expected runtime exhibits a certain robustness in the sense that it does not change by much for values of c in the considered range.

5.2 Comparison with empirical results

As shown in the beginning of this section, the results presented in subsection 5.1 approximate the expected runtime with only small error. However, this error is stated *asymptotically* and the computed approximations of the expected runtimes might differ greatly from the actual expected runtimes due to large constants hidden in the asymptotic term, especially for smaller values

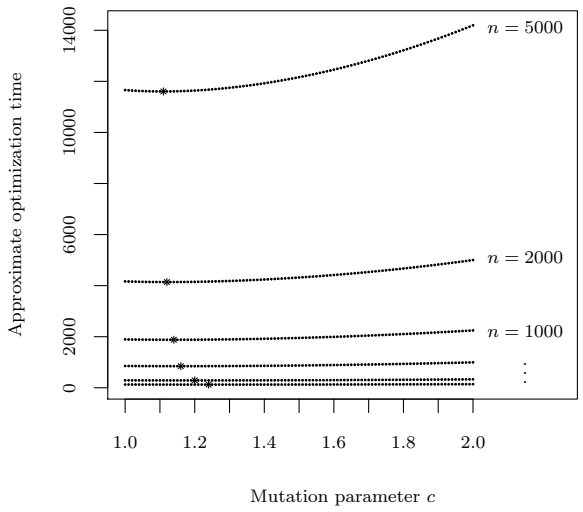

Fig. 1 Approximate expected runtime of the (1+10) EA using the Skellam-approximation for $n = 100, 200, 500, 1000, 2000, 5000$. The minimal c is marked for each n.

of n. However, this is not the case and our approximations turn out to be surprisingly precise, even for smaller values of n.

We performed experiments in order to compare the empirical optimal values of the mutation parameter c to the approximate optimal values of c. For this purpose we implemented the (1+λ) EA in C using the GNU Scientific Library (GSL) for the generation of pseudo-random numbers and to use the GSL implementations of the hypergeometric function and modified Bessel function of the first kind.

The results are shown in Figure 2. The plot displays the empirical optimal values for the mutation parameter c for $n \in \{100, 200, \ldots, 1000, 2000\}$ for the (1+1) EA, (1+50) EA and (1+100) EA, as dots. The empirical optimal values of c are determined by taking the empirical minimum of the corresponding runs of the (1+λ) EA for each pair of n and λ for $c \in \{0.50, 0.52, \ldots, 3.0\}$, averaged over 50000 runs each. The approximated optimal value for c using the Skellam-approximation as described in the beginning of this section is displayed by a dashed line for each setting of n and λ. Additionally, for illustration, the approximated optimal value for c using the exact formula for the BinDiff-distribution (see Corollary 6) is displayed by a continuous line for each setting of n and λ. The approximated values are determined by numerical optimization for both versions. We can see that for each λ both approximations seem to reflect the empirical optimal values very well over the whole range of n, producing only a small absolute error. Note that the empirical optimal

values are subject to some amount of variation, despite the high sample size of 50000 for each data point. This is due to the high variance in the distribution of the optimization time, which can also be observed in Figure 3 where the interquartile range of the experiments is illustrated. Moreover, the Skellam-approximation rapidly approaches the BinDiff-approximation which justifies its use even further.

Due to the fast computation of our approximation we additionally added values for $n = 5000$ and $n = 10000$ in Figure 3 to further illustrate the asymptotic behaviour of the optimal values for c.

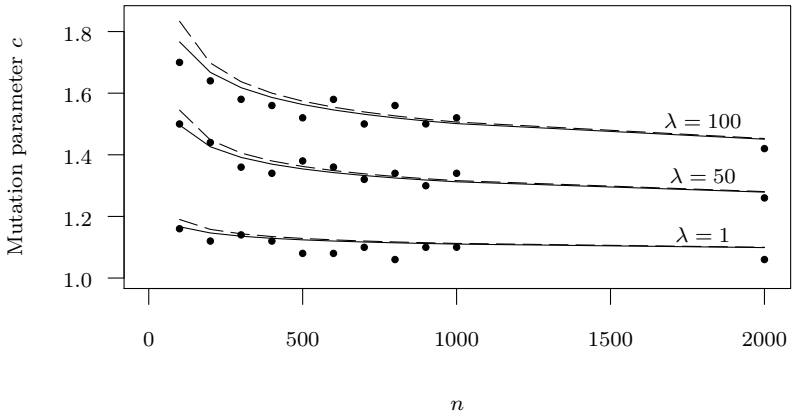

Fig. 2 Empirical optimal and approximate optimal mutation parameter c for the (1+λ) EA for $n = 100, 200, \ldots, 1000, 2000$ and $\lambda = 1, 50, 100$. The empirical optimal values for c are marked by dots. The approximate optimal values for c using the exact formula for the BinDiff-distribution are marked by the continuous line. The approximate optimal values for c using the Skellam-approximation for the BinDiff-distribution are marked by the dashed line.

As already mentioned in section 5.1 the approximate optimal mutation rate grows with λ for fixed n which is due to the lower order term in the expected runtime. To further illustrate the impact of the lower order term we displayed the empirical expected runtimes for several values of λ and fixed n and marked the empirical and approximate optimal values for the mutation parameters c. The results are shown in Figure 4. Both plots display the number of generations needed to optimize the (1+λ) EA for $n = 100$ (left plot) and $n = 2000$ (right plot) for various settings of λ and c. The empirical runtimes are displayed in each plot for $c = 0.5, 0.52, 0.54, \ldots, 3.0$ and $\lambda = 1, 2, 5, 10, 20, 50$ where each data point is averaged over 50000 runs. For illustration of the high variance,

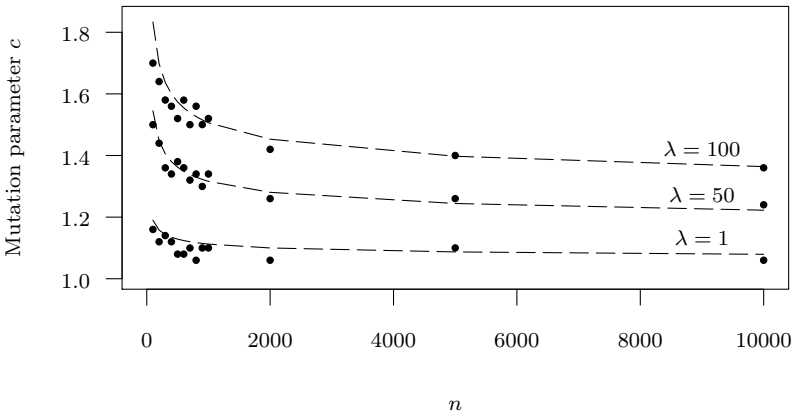

Fig. 3 Empirical optimal and approximate optimal mutation parameter c for the (1+λ) EA for $n = 100, 200, \ldots, 1000, 2000, 5000, 10000$ and $\lambda = 1, 50, 100$. The approximate optimal values for c using the Skellam-approximation for the BinDiff-distribution are marked by the dashed line.

the area between the first and third quartile is shaded. One can easily observe that for higher values of λ both the empirical and the approximate optimal values for c increase for both values of n. Note that higher values of λ lead to lower runtimes.

Conclusions

We have presented an improved variable drift theorem that weakens the requirement that no large steps towards the optimum may occur in the process to a stochastic one. We used this theorem to show that upper and lower bounds on the expected runtime of the (1+λ) EA with mutation probability c obtained from variable drift theorems are at most apart by a small lower order term if the exact drift is known and c and λ are constant. This reduces the analysis of expected optimization time to finding an exact expression for the drift.

Furthermore, we gave an exact closed-form expression for the drift and presented a method for approximating it very efficiently with small error. By applying the new drift theorem and the approximation for the drift, we were able to approximate optimal mutation rates for the (1+λ) EA for various parameter settings of c and λ and also for moderate sizes of n and verified experimentally that these approximations reflect empirical results very precisely.

Our results render the need for costly experiments in order to optimize the parameters unnecessary. Even for moderate n and not too small λ it turns

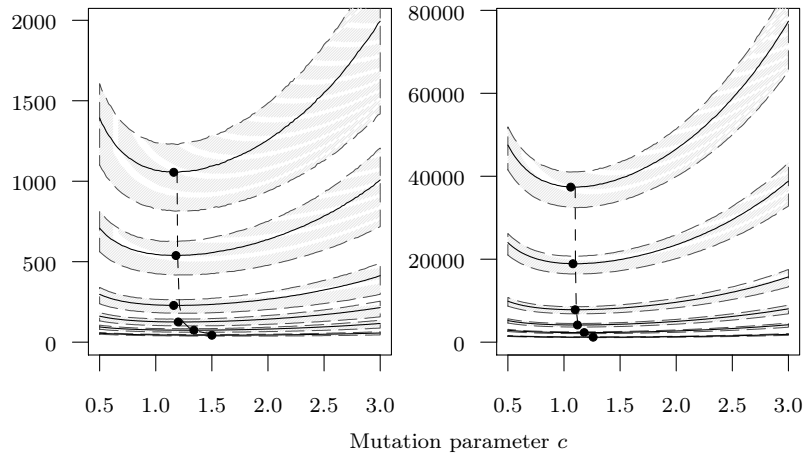

Fig. 4 Empirical expected runtime of the (1+λ) EA for $n = 100$ (left) and $n = 2000$ (right) and $\lambda = 1, 2, 5, 10, 20, 50$ (top to bottom in each plot) over 50000 runs for each setting of n, λ and $c = 0.5, 0.52, 0.54, \ldots, 3.0$. The empirical minimal c is marked by a dot for each n. The area between the first and third quartile is shaded for each λ. The computed approximations of the minimal c are denoted by the dashed line.

out that mutation rates up to 10% larger than the asymptotically optimal rate of $1/n$ minimize the expected runtime. However, the benefit of setting the mutation rate to the optimal value of c, instead of using mutation rate $1/n$ is small with respect to the actual expected runtime.

Acknowledgements

This work was supported by the Danish Council for Independent Research (DFF), grant no. 4002-00542.

References

[AD11] Anne Auger and Benjamin Doerr, editors. *Theory of Randomized Search Heuristics: Foundations and Recent Developments.* World Scientific Publishing, 2011.

[AS64] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* National Bureau of Standards, 1964.

[BDN10] Süntje Böttcher, Benjamin Doerr, and Frank Neumann. Optimal fixed and adaptive mutation rates for the leadingones problem. In *Proc. of Parallel Problem Solving from Nature (PPSN 2010)*, volume 6238, pages 1–10. Springer, 2010.

[BLS14] Golnaz Badkobeh, Per Kristian Lehre, and Dirk Sudholt. Unbiased black-box complexity of parallel search. In *Parallel Problem Solving from Nature - PPSN*

XIII - 13th International Conference, Ljubljana, Slovenia, September 13-17, 2014. Proceedings, pages 892–901, 2014.

[CSWA15] Francisco Chicano, Andrew M. Sutton, L. Darrell Whitley, and Enrique Alba. Fitness probability distribution of bit-flip mutation. *Evolutionary Computation*, 23(2):217–248, 2015.

[DDY16] Benjamin Doerr, Carola Doerr, and Jing Yang. Optimal parameter choices via precise black-box analysis. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference, Denver, CO, USA, July 20 - 24, 2016*, pages 1123–1130, 2016.

[DFW11] Benjamin Doerr, Mahmoud Fouz, and Carsten Witt. Sharp bounds by probability-generating functions and variable drift. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2011)*, pages 2083–2090. ACM Press, 2011.

[DG13] Benjamin Doerr and Leslie Ann Goldberg. Adaptive drift analysis. *Algorithmica*, 65(1):224–250, 2013.

[DJW02] Stefan Droste, Thomas Jansen, and Ingo Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81, 2002.

[DK13] Benjamin Doerr and Marvin Künnemann. Royal road functions and the (1+λ) evolutionary algorithm: Almost no speed-up from larger offspring populations. In *Proc. of the IEEE Congress on Evolutionary Computation (CEC 2013)*, pages 424–431. IEEE Press, 2013.

[DK15] Benjamin Doerr and Marvin Künnemann. Optimizing linear functions with the (1+λ) evolutionary algorithm – different asymptotic runtimes for different instances. *Theoretical Computer Science*, 561:3–23, 2015.

[GW15] Christian Gießen and Carsten Witt. Population size vs. mutation strength for the (1+λ) EA on OneMax. In *Proc. of GECCO '15*, pages 1439–1446. ACM Press, 2015.

[GW16] Christian Gießen and Carsten Witt. Optimal mutation rates for the (1+λ) EA on onemax. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference, Denver, CO, USA, July 20 - 24, 2016*, pages 1147–1154, 2016.

[HPR$^+$] Hsien-Kuei Hwang, Alois Panholzer, Nicolas Rolin, Tsung-Hsi Tsai, and Wei-Mei Chen. Probabilistic analysis of the (1+1)-evolutionary algorithm. To appear in Evolutionary Computation, 2017.

[Jan13] Thomas Jansen. *Analyzing Evolutionary Algorithms - The Computer Science Perspective*. Natural Computing Series. Springer, 2013.

[JJW05] Thomas Jansen, Kenneth A. De Jong, and Ingo Wegener. On the choice of the offspring population size in evolutionary algorithms. *Evolutionary Computation*, 13(4):413–440, 2005.

[Joh10] Daniel Johannsen. *Random combinatorial structures and randomized search heuristics*. PhD thesis, Universität des Saarlandes, Germany, 2010.

[LW14] Per Kristian Lehre and Carsten Witt. Concentrated hitting times of randomized search heuristics with variable drift. In *Proc. of ISAAC '14*, volume 8889 of *Lecture Notes in Computer Science*, pages 686–697. Springer, 2014. Full technical report at `http://arxiv.org/abs/1307.2559`.

[MRC09] Boris Mitavskiy, Jonathan E. Rowe, and Chris Cannings. Theoretical analysis of local search strategies to optimize network communication subject to preserving the total number of links. *International Journal of Intelligent Computing and Cybernetics*, 2(2):243–284, 2009.

[NW10] Frank Neumann and Carsten Witt. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity*. Natural Computing Series. Springer, 2010.

[RS14] Jonathan E. Rowe and Dirk Sudholt. The choice of the offspring population size in the (1, λ) evolutionary algorithm. *Theoretical Computer Science*, 545:20–38, 2014. Preliminary version in Proc. of GECCO 2012.

[Ske46] J. G. Skellam. The frequency distribution of the difference between two poisson variates belonging to different populations. *Journal of the Royal Statistical Society*, 109(3):296–296, 1946.

[Tee07] Kanint Teerapabolarn. A bound on the poisson-binomial relative error. *Statistical Methodology*, 4(4):407–415, 2007.

[Wit13] Carsten Witt. Tight bounds on the optimization time of a randomized search heuristic on linear functions. *Combinatorics, Probability & Computing*, 22(2):294–318, 2013. Preliminary version in Proc. of STACS '12.