Proceedings Paper:
Oliveto, P.S., Sudholt, D. orcid.org/0000-0001-6020-1646 and Witt, C. (2020) A tight lower bound on the expected runtime of standard steady state genetic algorithms. In: GECCO '20: Proceedings of the 2020 Genetic and Evolutionary Computation Conference. GECCO '20: Genetic and Evolutionary Computation Conference, 08-12 Jul 2020, Online conference. ACM Digital Library , pp. 1323-1331. ISBN 9781450371285

https://doi.org/10.1145/3377930.3390212

A Tight Lower Bound on the Expected Runtime of Standard Steady State Genetic Algorithms

Pietro S. Oliveto
Department of Computer Science
University of Sheffield, UK

Dirk Sudholt
Department of Computer Science
University of Sheffield, UK

Carsten Witt
DTU Compute
Technical University of Denmark

ABSTRACT

Recent progress in the runtime analysis of evolutionary algorithms (EAs) has allowed the derivation of upper bounds on the expected runtime of standard steady-state GAs. These upper bounds have shown speed-ups of the GAs using crossover and mutation over the same algorithms that only use mutation operators (i.e., steady-state EAs) both for standard unimodal (i.e., OneMax) and multimodal (i.e., Jump) benchmark functions. These upper bounds suggest that populations are beneficial to the GA as well as higher mutation rates than the default $1/n$ rate. However, making rigorous claims was not possible because matching lower bounds were not available. Proving lower bounds on crossover-based EAs is a notoriously difficult task as it is hard to capture the progress that a diverse population can make. We use a potential function approach to prove a tight lower bound on the expected runtime of the (2+1) GA for OneMax for all mutation rates c/n with $c < 1.422$. This provides the last piece of the puzzle that completes the proof that larger population sizes improve the performance of the standard steady-state GA for OneMax for various mutation rates, and it proves that the optimal mutation rate for the (2+1) GA on OneMax is $(\sqrt{97} - 5)/(4n) \approx 1.2122/n$.

1 INTRODUCTION

Explaining when recombination and mutation based genetic algorithms (GAs) perform better than more traditional general purpose search heuristics that use mutation alone is one of the fundamental problems in evolution-inspired computation. Traditionally proofs showing that crossover is a useful operator relied either on excessively low crossover rates [9, 11] or on some diversity-enforcing mechanism to make recombination effective by increasing the probability that members of the population are different [5, 6, 13, 15, 18]. However, it was never shown whether this enforced diversity was necessary or whether it was an additional requirement for the proofs to hold. Recently some results have appeared proving the superiority of standard steady state GAs[1] over mutation-only algorithms, without the need of any additional diversity enforcing mechanisms. Dang at al. [4] proved that for sufficiently large population sizes the $(\mu+1)$ GA is at least a linear factor faster than the best algorithm using only standard bit mutation for the Jump benchmark function. Hence, they showed that crossover may help algorithms to escape more quickly from local optima. Sutton [19] even proved that for the NP-hard Closest String problem from computational biology, the $(\mu+1)$ GA with sufficiently large population size and restarts is a fixed parameter tractable (FPT) algorithm while if only standard bit mutation is used (i.e., $(\mu+1)$ EA) it is not.

Strikingly, recombination has also been proven to be useful on unimodal functions. Lengler [14] has shown that there exist monotone functions for which the $(\mu+1)$ EA with not too low standard bit mutation rate c/n (i.e., $c > 2.13$) requires exponential runtime with high probability while the $(\mu+1)$ GA with sufficiently large population sizes can solve them in $O(n \log n)$ expected runtime for arbitrary mutation rates i.e., $\Theta(1)/n$. Analyses have revealed that the $(\mu+1)$ GA is faster than the $(\mu+1)$ EA using any standard bit mutation rate and population size, even on unimodal functions where the latter is particularly efficient i.e., OneMax [2, 3]. Furthermore, if the fitness of offspring that are identical to their parents is not unnecessarily re-evaluated, then the algorithm is faster than any unary unbiased black box algorithm for the problem [12], albeit slower than if the diversity is enforced [1, 18]. To prove these results, precise analyses up to the leading constants are required since for OneMax the algorithms have the same asymptotic expected runtime $O(n \log n)$ for moderate population sizes.

An important insight from these analyses is that if diversity is enforced as in Sudholt's work [18], then inevitably there are no advantages of using population sizes greater than $\mu = 2$ for OneMax. On the other hand, the analysis of Corus and Oliveto [3] provides upper bounds that decrease with the population size (up to some sub-logarithmic limit). For large enough population sizes the best derived upper bound is roughly $1.64n \ln n$ while, for $\mu = 2$, Corus and Oliveto only provide a larger upper bound of $\frac{4e^c n \ln n}{c(c+4)} + O(n)$ [2]. Due to a mistake in one probability calculation this turns out to actually be $\frac{9e^c n \ln n}{c(2c+9)} + O(n)$.

Indeed, *all* the positive results summarised above regarding the plain $(\mu+1)$ GA required sufficiently large population sizes. While the comparative statements with the mutation-based algorithms were possible because of the availability of lower bounds on their expected runtime, rigorously showing whether the suggested population sizes are actually necessary requires lower bounds on the expected runtime of the $(\mu+1)$ GA. Proving lower bounds for GAs with crossover is a notoriously hard task. The only available analysis concerning a standard GA is the proof that the simple genetic algorithm (SGA [7]) cannot solve OneMax in polynomial time with overwhelming probability due to the ineffectiveness of the fitness proportional selection operator [16, 17]. There have been recent attempts to generalize proof methods like the family tree technique to crossover-based algorithms [20]; however, these only apply in a specific setting without mutation.

Providing lower bounds on the expected runtime of the $(\mu+1)$ GA for OneMax has turned out to be surprisingly difficult. Sudholt simplified the analysis by considering a "greedy" (2+1) GA that always selects amongst the fittest individuals in the population and is sped-up by automatically achieving the best possible crossover operation between different parents [18]. A less greedy (2+1) GA

[1] Steady state GAs, are those that replace at most a subset of the population in each generation (usually one new individual is created). Typically they use standard bit mutation which flips each bit independently with probability c/n.

was considered by Corus and Oliveto where individuals are only immediately crossed over optimally if the Hamming distance between the parents is larger than 2 [2]. These simplified algorithms allow the analysis to ignore the improvements which may occur in standard GAs when one parent is crossed over with another one of different fitness. However, it was never proven that the algorithms are indeed faster than the standard (2+1) GA, hence that the bounds are also valid for the latter algorithm. In this paper we provide a lower bound for the (2+1) GA with no simplifications that matches its upper bound up to the leading constant, hence providing a rigorous proof that larger populations are beneficial to the GA for OneMax. The preciseness of the results also allows us to derive that the optimal mutation rate c/n for the algorithm is approximately $c = 1.21221445$ for any $c \in (0, 1.422]$.

A major difficulty in proving rigorous lower bounds for populations with crossover is to find a way to aggregate the state of the algorithm such that it accurately captures the current distance from the optimum, but also the potential improvements of the crossover operator. These advancements could be very big if the parents have a large Hamming distance, and our aim is to show that this rarely happens. We solve the aggregation problem for the (2+1) GA by defining a potential function that captures the current fitness and opportunities for easy improvements through crossover. By showing bounds on the expected increase in the potential, we are able to quantify how the distance to the optimum decreases in one generation. The challenge lies in proving this for every possible population, from those with identical individuals to those with a good amount of diversity. Once the potential is appropriately bounded, we can use standard drift analysis arguments to bound the expected time from below.

1.1 Main Contributions

The expected optimisation time of the (2+1) GA is bounded from above as follows.

Theorem 1. *The expected optimisation time of the (2+1) GA with mutation rate c/n, $c > 0$ a constant, on OneMax is at most*

$$\frac{9e^c}{c(2c+9)} \cdot n \ln n + O(n).$$

For $c = 1$ this is $\frac{9}{11} \cdot en\ln(n) + O(n) \approx 2.224n\ln(n) + O(n)$. The upper bound follows from applying the analytical framework in [2] with a corrected transition probability for p_r, using the value $1/(4e)$ instead of $5/(24e)$. It can also be proven with mild adaptations of the proof of [18, Theorem 4].

Our main contribution is the following lower bound that matches the upper bound proven in Theorem 1 up to small-order terms.

Theorem 2. *The expected optimisation time of the (2+1) GA with mutation rate c/n, and $0 < c \leq 1.422$ a constant, on OneMax is at least*

$$\frac{9e^c}{c(2c+9)} \cdot n \ln n - O(n \log\log n).$$

Since the bounds from Theorems 1 and 2 have the same leading constant $\frac{9e^c}{c(2c+9)}$, which is minimised for

$$c = \frac{\sqrt{97} - 5}{4} \approx 1.21221445,$$

we identify this as the optimal mutation rate for the (2+1) GA (up to small-order terms) within the range of rates covered by Theorem 2.

Algorithm 1 (2+1) GA

```
Initialize $P = \{x^1, x^2\}$ by selecting two search points from $\{0,1\}^n$
independently and uniformly at random (u. a. r.).
for $t \leftarrow 1, 2, \ldots$ do
    Select $y^1$ and $y^2$ from $P$ u.a.r. with replacement.
    Create $z$ by applying uniform crossover to $y^1$ and $y^2$.
    Flip each bit in $z$ independently with probability $c/n$.
    Remove the worst of $x^1$, $x^2$ and $z$, breaking ties u. a. r.
end for
```

Theorem 3. *Amongst all mutation rates c/n with $c \in (0, 1.422]$, the choice $c = \frac{\sqrt{97}-5}{4}$ is the optimal mutation rate of the (2+1) GA on OneMax, up to small-order terms. Then the expected optimisation time is $\approx 2.18417n \ln n + O(n)$.*

The best identified mutation rate for the (2+1) GA is lower than the one minimising the upper bound for larger population sizes $\mu \geq 5$ (it is at least $1.425/n$ and increases with μ) always providing upper bounds below $1.7n \ln n$ and decreasing with μ [2]. This implies that the (2+1) GA with mutation rate $\frac{\sqrt{97}-5}{4n}$ is at least 28% slower than any (μ+1) GA with $\mu \geq 5$ and appropriate mutation rates.

In this extended abstract, many proofs are removed because of space constraints.

2 PRELIMINARIES

The (2+1) GA is defined in Algorithm 1. The algorithm initialises the population with two randomly chosen individuals. At each generation it selects two random parents with replacement to be mated via uniform crossover. The operator assigns each bit to the offspring by selecting the corresponding bit from one parent with probability 1/2 and from the other with the same probability. Standard bit mutation is then applied to the offspring by flipping each of its bits independently with probability c/n. Finally, the worst individual amongst the parents and the offspring is removed to select the new population. Ties are broken uniformly at random.

We will analyse the expected runtime of the algorithm to optimise the function $f(x) = \text{OneMax}(x) = \sum_{i=1}^{n} x_i$, which counts the number of 1-bits in a bitstring. We write a population $\{x^1, x^2\}$ in order of monotonically decreasing fitness, that is, $f(x^1) \geq f(x^2)$. Let n_{11} be the number of bit positions where both parents have ones and likewise for n_{00} and the number of zeros. Let n_{10} be the number of positions where x^1 has a 1 and x^2 has a 0 and likewise for n_{01}. Then we have $f(x^1) = n_{11} + n_{10}$ and $f(x^2) = n_{11} + n_{01}$. Since by assumption, $f(x^1) \geq f(x^2)$, we have $n_{10} \geq n_{01}$ and $n_{10} = n_{01}$ is equivalent to the two individuals having equal fitness. In case $n_{10} = 0$, both individuals are identical. Such a population is called *monomorphic* in population genetics, and we use this term here.

Note that the (2+1) GA is an *unbiased* algorithm in the sense of Lehre and Witt [12]. Hence, due to symmetry of bit positions, it suffices to know n_{11}, n_{10} and n_{01} to fully characterise the state of the algorithm. Note that n_{00} can be derived as $n - n_{11} - n_{10} - n_{01}$.

The following lemma characterises probabilities of setting a bit to 1 in the offspring after a crossover of two different parents and a mutation of the result.

Lemma 4. *Consider a crossover of two parents x, y followed by mutation with mutation rate p_m, resulting in an offspring z. For all i,*

$$\mathrm{P}(z_i = 1) = \begin{cases} 1 - p_m & \text{if } x_i = y_i = 1 \\ 1/2 & \text{if } x_i \neq y_i \\ p_m & \text{if } x_i = y_i = 0. \end{cases}$$

Proof. If $x_i = y_i$ then crossover will create an offspring with the same bit value. The statement for $x_i \neq y_i$ holds because of symmetry, or using the following, alternative argument. The offspring has a 1 if crossover creates a 1 and mutation does not flip bit i, or if crossover creates a 0 and mutation does flip bit i. The probability of the former event is $1/2 \cdot (1 - p_m)$ and the probability of the latter event is $1/2 \cdot p_m$. Together, this gives $1/2$. □

Note that differing bits $x_i \neq y_i$ are set to 1 with probability $1/2$, irrespective of the mutation rate. Hence, when two parents are selected, we only need to consider the effect of mutation on the bits where the parents agree. We frequently and tacitly use this fact.

3 A POTENTIAL FUNCTION APPROACH

Our lower bound applies when only considering populations where the number of zeros in the fitter parent is at most $n/\mathrm{polylog}(n)$ and at least $\mathrm{polylog}(n)$. This implies that all probabilities that involve flipping a 0 to 1 are polylogarithmically small.

The main tool for our lower bound is going to be drift analysis, applied to a potential function that captures the current state and potential easy fitness improvements.

Definition 1. *For a population P with values $n_{11}, n_{10}, n_{01}, n_{00}$ we define the potential of P as*

$$\varphi(P) = n_{11} + n_{10} + \frac{n_{01}}{3}.$$

The intuition is that $n_{11} + n_{10}$ describes the current best fitness in the population. The term $n_{01}/3$ adds potential to the best fitness as the population has the potential to exploit the diversity given by the n_{01} 1-bits that only exist in the less fit individual during a successful crossover operation.

The choice of the factor $1/3$ is motivated as follows. We know from previous work [2, 18] that the most helpful populations for improvements are those where two search points have the same number of ones and Hamming distance 2, that is, $n_{10} = n_{01} = 1$. (Larger Hamming distances have the potential for larger fitness improvements, but such populations are rarely reached when the number of zeros becomes reasonably small.)

Assume the current state has $n_{10} = n_{01} = 1$, corresponding to a potential of $n_{11} + n_{10} + 1/3$. The most likely transitions (and, when only $O(n/\mathrm{polylog}(n))$ zeros are left, the only transitions with probability $\Omega(1)$) are (1) collapsing the population to copies of one parent (and potential $n_{11}+n_{10}$) and (2) creating a surplus of one 1-bit by crossover and not flipping anything else (potential $n_{11} + n_{10} + 1$). The probability of the former event is roughly[2] $(1 - p_m)^n/4$, which is the probability of selecting the same parent twice, not flipping any bits and then selecting the other population member for removal plus the probability of selecting different parents, creating one parent by crossover and not flipping any bits. The probability of the latter event is roughly $(1 - p_m)^n/8$, which is the probability of selecting different parents, setting both differing bits to 1 and not flipping any bits in the subsequent mutation.

Comparing these terms, the conditional probability of an improvement via crossover is roughly $1/3$. In case a monomorphic population is reached, the potential reduces by $1/3$ and this happens with conditional probability $1 - 1/3$. In the latter event, the potential increases by $1 - 1/3$ and this happens with conditional probability $1/3$. The net effect of these transitions in the expected change of the potential is $(1 - 1/3)1/3 - 1/3(1 - 1/3) = 0$. So the potential balances out the effects of "volatile" states left quickly.

Obviously, our analysis still needs to account for other, less likely transitions. For populations with $n_{10} = n_{01} > 1$ the conditional transition probabilities change as the probability of creating one of the parents by crossover depends on the Hamming distance $n_{10} + n_{01}$ between parents. For $n_{10} = n_{01} > 1$ the likely progress in a successful crossover may be smaller than $n_{01}/3$. Hence the term $+n_{01}/3$ in Definition 1 is a precise estimate for the likely progress when $n_{01} = 1$ and for larger n_{01} it is an overestimation.

It suffices to restrict our considerations to moderate values of n_{10} and n_{01}. The reason is that the (2+1) GA always has a constant probability of creating a monomorphic population in one generation, regardless of the current population. This means that large values of n_{10} and n_{01} are very unlikely.

Lemma 5. *Let $t \geq \log^2 n$ and $t = n^{O(1)}$. With probability $1 - n^{-\Omega(\log n)}$, all populations within the time interval $[\log^2 n, t]$ have Hamming distance at most $\log^2 n$ between their two individuals.*

Proof. We call a generation that creates a population of two identical individuals a *monomorphic* generation. The crucial idea is to show that monomorphic generations are very frequent so that large Hamming distances are unlikely to occur.

The probability of a monomorphic generation happening is at least $(1/4)(1 - 1/n)^n(1/3) = \Omega(1)$ since it is sufficient to select a fittest parent twice, to clone it and to remove the other parent (which has probability at least $1/3$). For a number $t \geq 0$ of generations after a monomorphic one, let D_t denote the maximum number of bits in which the two parents *ever* have differed during these t generations. The crucial idea is that only mutations can increase this D-value. The total number of bits flipped in t generations is the sum of tn Poisson trials with success probability c/n each. Hence, within t generations following a monomorphic one, the D-value is bounded from above by $2ct$ with probability $1 - 2^{-\Omega(t)}$ according to Chernoff bounds, and clearly the Hamming distance is no larger than the D-value. We set $t := (\log^2 n)/(2c)$ to bound the D-value by $\log^2 n$.

The proof is completed by noting that the probability of not observing a monomorphic generation within $\log^2 n$ generations is $(1 - \Omega(1))^{\log^2 n} = n^{-\Omega(\log n)}$. Together with the failure bound $2^{-\Omega(t)}$, which is $n^{-\Omega(\log n)}$ for $t = (\log^2 n)/(2c)$, and a union bound, this means that in any polynomial number of generations following the first monomorphic one the Hamming distance never exceeds $\log^2 n$ with probability $1 - n^{O(1)} n^{-\Omega(\log n)} = 1 - n^{-\Omega(\log n)}$. □

[2] In this informal discussion we ignore events of smaller probability (e. g. picking the same parent twice and creating the other population member by a lucky mutation).

4 ROADMAP FOR THE ANALYSIS

We give a deliberately informal, high-level view of our analysis, where $\Delta := \varphi(P_{t+1}) - \varphi(P_t)$ denotes the change in potential in one generation. By the law of total probability, Δ can be split up according to the number of zeros flipped by mutation:

$$\begin{aligned}\mathrm{E}(\Delta) = {} & \mathrm{E}(\Delta \mid \text{no zeros flip}) \cdot \mathrm{P}(\text{no zeros flip}) \\ & + \mathrm{E}(\Delta \mid \text{one zero flips}) \cdot \mathrm{P}(\text{one zero flips}) \\ & + \mathrm{E}(\Delta \mid \text{at least 2 zeros flip}) \cdot \mathrm{P}(\text{at least 2 zeros flip}).\end{aligned}$$

The case that no zeros flip does not increase the potential, hence we aim to bound the first line from above by 0. The second line captures the most important case: one zero flips and subsequent progress is made. The second line will be bounded by the dominant term in our claimed lower bound. The third line involves the probability of flipping at least two zeros. If the number of zeros is small, this is unlikely and thus the third line only contributes a small order term.

The above high-level view is not particularly accurate. Firstly, the above estimations need to account for error terms. Secondly, the notion of "i zeros flip" used above is not well-defined. This is because the number of zeros that can flip during mutation depends on the parent selection. The same parents may be selected twice, and then the number of zeros depends on the fitness of the parent. If two different parents are used, we only consider mutations of bits that agree in both parents, as per Lemma 4.

Hence, we need to distinguish between different events from the parent selection. To formalise this, let P_{11}, P_{22}, and P_{12} denote the events that parent selection chooses the first parent twice, the second parent twice and both parents, respectively. We further denote by F_{00} the number of flipping bits amongst the n_{00} bits and likewise for F_{11} and n_{11} bits. We use asterisks to indicate the union of sets: F_{0*} is the number of flipping bits among $n_{01} + n_{00}$ bits and F_{*0} is the number of flipping bits among $n_{10} + n_{00}$ bits. Variables F_{1*} and F_{*1} are defined analogously. Armed with this notation, we express the third line rigorously with a combination of events.

Lemma 6. *For all populations with $n_{11} \geq n - n/\log^3 n$ and $n_{10} + n_{01} \leq \log^2 n$,*

$$\begin{aligned}& \mathrm{E}(\Delta \mid P_{11}, F_{0*} \geq 2)\mathrm{P}(F_{0*} \geq 2) \\ & + \mathrm{E}(\Delta \mid P_{22}, F_{*0} \geq 2)\mathrm{P}(F_{*0} \geq 2) \\ & + \mathrm{E}(\Delta \mid P_{12}, F_{00} \geq 2)\mathrm{P}(F_{00} \geq 2) = O\left(n_{0*}/(n \log n)\right)\end{aligned}$$

Proof. We give one common way of bounding the three lines from the statement. For all $F \in \{F_{0*}, F_{*0}, F_{00}\}$, the drift can only increase by at most $n_{01} + F$ as every flipping 0-bit can only increase the potential by at most 1 and crossover can increase the potential on all n_{01} by at most 1. Also note that $\mathrm{P}(F_{*0} \geq 2)$ has the largest probability amongst all variables F (as the underlying number of zeros is maximal for F_{*0}). Thus, all 3 lines are bounded as

$$\begin{aligned}& \mathrm{E}(\Delta \mid F \geq 2)\mathrm{P}(F_{*0} \geq 2) \\ = {} & \sum_{i=2}^{\infty} \mathrm{E}(\Delta \mid F = i)\mathrm{P}(F_{*0} = i) \leq \sum_{i=2}^{\infty} (n_{01} + i) \cdot \mathrm{P}(F_{*0} = i) \\ \leq {} & \sum_{i=2}^{\infty} (n_{01} + i) \left(\frac{cn_{0*}}{n}\right)^i = \left(\frac{cn_{0*}}{n}\right)^2 \sum_{i=0}^{\infty} (n_{01} + i + 2) \left(\frac{cn_{0*}}{n}\right)^i\end{aligned}$$

Since $n_{0*} \leq n - n_{11} \leq n/\log^3 n$, we get $\left(\frac{cn_{0*}}{n}\right)^2 = O(n_{0*}/(n \log^3 n))$. The sum is bounded by $n_{01} + 2 + \sum_{i=0}^{\infty} i \left(\frac{cn_{0*}}{n}\right)^i$, which is $n_{01} + O(1) = O(\log^2 n)$. Together, this implies the claim. □

The cases of no zeros flipping and one zero flipping are more difficult to handle. Corresponding drift estimates will be derived in the following sections.

5 POTENTIAL DRIFT WHEN NO ZEROS FLIP

We now consider the potential drift when no zeros flip. When the distance to the optimum is $o(n)$, this case is by far the most frequent case. This also means that our drift bounds have to be precise, as even a small error term may have a big impact and spoil the analysis.

We start by considering the drift conditional on selecting the same parent twice.

Lemma 7. *For all populations with $n_{10} = n_{01}$,*

$$\begin{aligned}\mathrm{E}(\Delta \mid P_{11}, F_{0*} = 0)\mathrm{P}(F_{0*} = 0) &= -\frac{n_{01}}{9}\left(1 - \frac{c}{n}\right)^n \\ \mathrm{E}(\Delta \mid P_{22}, F_{*0} = 0)\mathrm{P}(F_{*0} = 0) &= -\frac{n_{01}}{9}\left(1 - \frac{c}{n}\right)^n.\end{aligned}$$

For all populations with $n_{10} > n_{01}$,

$$\begin{aligned}\mathrm{E}(\Delta \mid P_{11}, F_{0*} = 0)\mathrm{P}(F_{0*} = 0) &= -\frac{n_{01}}{3}\left(1 - \frac{c}{n}\right)^n \\ \mathrm{E}(\Delta \mid P_{22}, F_{*0} = 0)\mathrm{P}(F_{*0} = 0) &= 0.\end{aligned}$$

Proof. First assume $n_{10} = n_{01}$ and consider the event P_{11}. Given $F_{0*} = 0$, that is, if no 0-bit is flipped, the offspring can only be accepted if no 1-bit is flipped, i. e., $F_{1*} = 0$. These events happen with probability $\mathrm{P}(F_{0*} = 0)\mathrm{P}(F_{1*} = 0) = (1 - c/n)^n$ and they lead to an offspring that is identical to x^1. Since all search points have equal fitness, x^2 is removed with probability $1/3$. This leads to a monomorphic population and the potential decreases by $n_{01}/3$. Multiplying the above terms proves the claimed equality. The case of P_{22} follows analogously, considering F_{*0} and F_{*1} instead.

For $n_{10} > n_{01}$, if the fitter parent x^1 is selected twice, a copy of it is created with probability $(1 - c/n)^n$ and then x^2 is removed. This decreases the potential by $n_{01}/3$. Multiplying the above terms yields an expectation of $-n_{01}/3 \cdot (1 - c/n)^n$. Note that other operations cannot increase the potential since no 0-bit is being flipped and flipping 1-bits in x^1 does not decrease the potential.

If x^2 is selected twice as parent, the potential cannot increase since no 0-bits are flipped, and it cannot decrease as any 1-bit being flipped will lead to the offspring being rejected. □

Now we consider the drift in the potential when two different parents are chosen. We first deal with the case of both search points having equal fitness, $n_{10} = n_{01}$. The following lemma gives a closed formula for the potential drift in this case.

Lemma 8. *Let $S \sim \mathrm{Bin}(n_{10} + n_{01}, 1/2)$ and $\ell := n_{10} - F_{00} + F_{11}$. Then for all populations with $n_{10} = n_{01}$,*

$$\begin{aligned}& \mathrm{E}(\Delta \mid P_{12}, F_{00}, F_{11}) \\ & = \mathrm{P}(S \geq \ell) \cdot \left(\frac{5}{6} \cdot \mathrm{E}(S \mid S \geq \ell) + F_{00} - \frac{2F_{11}}{3} - n_{10}\right) \\ & \quad + \mathrm{P}(S = \ell) \cdot \frac{n_{10} - F_{00} - F_{11}}{18}\end{aligned}$$

Proof. Consider a step where both parents are selected and F_{00}, F_{11} are known. The number of bits set to 1 by crossover is given by $S \sim \mathrm{Bin}(n_{10} + n_{01}, 1/2)$. By the law of total probability,

$$\mathrm{E}(\Delta \mid P_{12}, F_{00}, F_{11}) = \sum_{s \in \mathbb{Z}} \mathrm{E}(\Delta \mid P_{12}, F_{00}, F_{11}, S = s) \cdot \mathrm{P}(S = s).$$

Since $n_{10} = n_{01}$, the fitness of both parents is $n_{11} + n_{10}$ and the fitness of the offspring is $n_{11} + s + F_{00} - F_{11}$. If $s < \ell$, the latter is less than $n_{11} + n_{10}$ and the offspring will be rejected. If $s > \ell$, the offspring is fitter than the parent, one of the parents will be chosen uniformly at random for removal. If the offspring is as fit as both parents, the offspring is removed with probability $1/3$. Thus,

$$\begin{aligned}
&\mathrm{E}(\Delta \mid P_{12}, F_{00}, F_{11})\\
&= \sum_{s \in \mathbb{Z}} \mathrm{E}(\Delta \mid P_{12}, F_{00}, F_{11}, S = s) \cdot \mathrm{P}(S = s)\\
&= \sum_{s > \ell} \mathrm{E}(\Delta \mid S = s) \cdot \mathrm{P}(S = s) + \frac{2}{3} \cdot \mathrm{E}(\Delta \mid S = \ell) \cdot \mathrm{P}(S = \ell)\\
&= \sum_{s \geq \ell} \mathrm{E}(\Delta \mid S = s) \cdot \mathrm{P}(S = s) - \frac{1}{3} \cdot \mathrm{E}(\Delta \mid S = \ell) \cdot \mathrm{P}(S = \ell).
\end{aligned}$$

Now we estimate $\mathrm{E}(\Delta \mid S = s)\mathrm{P}(S = s)$ for $s \geq \ell$. Let S_{10} be the number of bits among the n_{10} bits that are set to 1 in the offspring and define S_{01} analogously for the n_{01} bits. Note that $S := S_{10} + S_{01}$ where $S_{10} \sim \mathrm{Bin}(n_{10}, 1/2)$ and $S_{01} \sim \mathrm{Bin}(n_{01}, 1/2)$.

Given $S_{10} = s_{10}$ and $S_{01} = s_{01}$, the potential difference Δ is derived as follows. Among the n_{11} bits, F_{11} bits flip to 0, reducing their contribution from 1 to $1/3$ each, leading to a contribution of $-2F_{11}/3$ to the potential difference. All the n_{10} bits contribute 1 to the potential $\varphi(P_t)$. In P_{t+1}, s_{10} bits contribute 1 and the remaining $n_{10} - s_{10}$ bits contribute $1/3$ each. Hence the contribution to the potential difference is $-2(n_{10} - s_{10})/3$. The n_{01} bits contribute $1/3$ each in $\varphi(P_t)$ and in P_{t+1} we have s_{01} bits contributing 1 each and the other $n_{01} - s_{01}$ bits contributing 0. Hence the contribution to the potential difference is $s_{01} - n_{01}/3$. Finally, the contribution of the n_{00} bits to the potential difference is F_{00}. Together,

$$\begin{aligned}
(\Delta \mid S_{10} = s_{10}, S_{01} = s_{01}) &= F_{00} - \frac{2F_{11}}{3} - \frac{2(n_{10} - s_{10})}{3} + s_{01} - \frac{n_{01}}{3}\\
&= F_{00} - \frac{2F_{11}}{3} - n_{10} + s - \frac{1}{3} \cdot s_{10}. \qquad (1)
\end{aligned}$$

By the law of total probability,

$$\begin{aligned}
&\mathrm{E}(\Delta \mid S = s)\\
&= \sum_{s_{10}=0}^{s} \mathrm{E}(\Delta \mid S = s, S_{10} = s_{10})\mathrm{P}(S_{10} = s_{10} \mid S = s)\\
&= \sum_{s_{10}=0}^{s} \left(F_{00} - \frac{2F_{11}}{3} - n_{10} + s - \frac{1}{3} \cdot s_{10}\right)\mathrm{P}(S_{10} = s_{10} \mid S = s)\\
&= F_{00} - \frac{2F_{11}}{3} - n_{10} + s - \frac{1}{3} \sum_{s_{10}=0}^{s} s_{10} \cdot \mathrm{P}(S_{10} = s_{10} \mid S = s)\\
&= F_{00} - \frac{2F_{11}}{3} - n_{10} + s - \frac{1}{3} \cdot \mathrm{E}(S_{10} \mid S = s).
\end{aligned}$$

Now, $(S_{10} \mid S = s)$ follows a hypergeometric distribution with parameters n_{10} (number of red balls), $n_{10}+n_{01}$ (number of balls) and s (number of draws). The expectation is thus $s \cdot n_{10}/(n_{10}+n_{01}) = s/2$. Plugging this in yields

$$\mathrm{E}(\Delta \mid S = s) = F_{00} - \frac{2F_{11}}{3} - n_{10} + \frac{5}{6} \cdot s. \qquad (2)$$

Together, this gives

$$\begin{aligned}
&\mathrm{E}(\Delta \mid P_{12}, F_{00}, F_{11})\\
&= \sum_{s \geq \ell} \mathrm{E}(\Delta \mid S = s) \cdot \mathrm{P}(S = s) - \frac{1}{3} \cdot \mathrm{E}(\Delta \mid S = \ell) \cdot \mathrm{P}(S = \ell)\\
&= \sum_{s \geq \ell} \left(F_{00} - \frac{2F_{11}}{3} - n_{10} + \frac{5}{6} \cdot s\right) \cdot \mathrm{P}(S = s)\\
&\quad - \frac{1}{3} \cdot \left(F_{00} - \frac{2F_{11}}{3} - n_{10} + \frac{5}{6} \cdot \ell\right) \cdot \mathrm{P}(S = \ell).
\end{aligned}$$

Using

$$\begin{aligned}
&\sum_{s \geq \ell} s \cdot \mathrm{P}(S = s) = \sum_{s \geq \ell} s \cdot \mathrm{P}(S = s \wedge S \geq \ell)\\
&= \sum_{s \geq \ell} s \cdot \mathrm{P}(S = s \mid S \geq \ell)\mathrm{P}(S \geq \ell) = \mathrm{E}(S \mid S \geq \ell)\mathrm{P}(S \geq \ell),
\end{aligned}$$

the first terms simplify as

$$\begin{aligned}
&\sum_{s \geq \ell} \left(F_{00} - \frac{2F_{11}}{3} - n_{10} + \frac{5}{6} \cdot s\right) \cdot \mathrm{P}(S = s)\\
&= \left(F_{00} - \frac{2F_{11}}{3} - n_{10}\right) \sum_{s \geq \ell} \mathrm{P}(S = s) + \frac{5}{6} \sum_{s \geq \ell} s \cdot \mathrm{P}(S = s)\\
&= \left(F_{00} - \frac{2F_{11}}{3} - n_{10}\right) \mathrm{P}(S \geq \ell) + \frac{5}{6} \cdot \mathrm{E}(S \mid S \geq \ell)\mathrm{P}(S \geq \ell).
\end{aligned}$$

The last term simplifies as

$$\begin{aligned}
&- \frac{1}{3} \cdot \left(F_{00} - \frac{2F_{11}}{3} - n_{10} + \frac{5}{6} \cdot \ell\right) \cdot \mathrm{P}(S = \ell)\\
&= - \frac{1}{3} \cdot \left(F_{00} - \frac{2F_{11}}{3} - n_{10} + \frac{5}{6} \cdot (n_{10} - F_{00} + F_{11})\right) \cdot \mathrm{P}(S = \ell)\\
&= - \frac{1}{3} \cdot \left(\frac{F_{00}}{6} + \frac{F_{11}}{6} - \frac{n_{10}}{6}\right) \cdot \mathrm{P}(S = \ell)\\
&= - \frac{F_{00} + F_{11} - n_{10}}{18} \cdot \mathrm{P}(S = \ell) = \frac{n_{10} - F_{00} - F_{11}}{18} \cdot \mathrm{P}(S = \ell).
\end{aligned}$$

Together, this proves the claim. □

The bound from Lemma 8 depends on the expected surplus $\mathrm{E}(S \mid S \geq \ell)$ generated by a crossover on the bits that differ between the two parents, where ℓ reflects the fitness threshold above which offspring are accepted. We use the following formula to simplify such expressions. The proof goes back to work by Gruder [8] that is highlighted in a paper by Johnson [10].

Lemma 9. *Let $S \sim \mathrm{Bin}(n, 1/2)$, then for all $\ell \in \mathbb{N}$,*

$$\mathrm{E}(S \mid S \geq \ell)\mathrm{P}(S \geq \ell) = \frac{\ell}{2} \cdot \mathrm{P}(S = \ell) + \frac{n}{2} \cdot \mathrm{P}(S \geq \ell).$$

Using Lemma 9, we obtain the following simplified formula.

Lemma 10. *Let $S \sim \mathrm{Bin}(n_{10} + n_{01}, 1/2)$. Then for all populations with $n_{10} = n_{01}$,*

$$\begin{aligned}&\mathrm{E}(\Delta \mid P_{12}, F_{00}, F_{11})\\ &= \mathrm{P}(S \geq n_{10} - F_{00} + F_{11}) \cdot \left(F_{00} - \frac{2F_{11}}{3} - \frac{n_{10}}{6}\right)\\ &\quad + \mathrm{P}(S = n_{10} - F_{00} + F_{11}) \cdot \frac{17n_{10} - 17F_{00} + 13F_{11}}{36}.\end{aligned}$$

We use Lemma 10 to show that, to get an upper bound on the drift when $F_{00} = 0$, we only need to consider $F_{11} \in \{0, 1\}$ as larger values lead to a non-positive drift. This is not obvious, but follows from a lengthy and trite calculation.

Lemma 11. *For all $n_{10} = n_{01}$ and all $i \geq 2$,*

$$\mathrm{E}(\Delta \mid P_{12}, F_{00} = 0, F_{11} = i) \leq 0.$$

Now we are able to give an upper bound on the potential drift.

Lemma 12. *For every population with $n_{10} = n_{01}$,*

$$\mathrm{E}(\Delta \mid P_{12}, F_{00} = 0) \leq \begin{cases} 0 & \text{if } n_{10} = 0\\ \frac{1}{9} \cdot \mathrm{P}(F_{11} = 0) & \text{if } n_{10} = 1\\ \frac{1}{8} \cdot \mathrm{P}(F_{11} = 0) + \frac{1}{64} \cdot \mathrm{P}(F_{11} = 1) & \text{if } n_{10} \geq 2.\end{cases}$$

For the case of unequal fitness, we use similar arguments as before. This scenario has more involved calculations as we need to distinguish different cases: the offspring may be at least as good as the fitter parent, and then the calculations are similar to the equal-fitness scenario. The offspring may also be worse than the fitter parent and better than the worse parent. In this case, the potential is derived from a different formula as the values for n_{10} and n_{01} in the next generation are still determined according to the fitter parent. In case the offspring's fitness is equal to that of the worse parent, there is a tie and the offspring is only accepted with probability 1/2. The lemma defines two thresholds ℓ_1 and ℓ_2 that reflect the number of bits crossover needs to set to 1 to achieve the fitness of x^1 and x^2, respectively.

Lemma 13. *Let $S \sim \mathrm{Bin}(n_{10} + n_{01}, 1/2)$, $\ell_1 := n_{10} - F_{00} + F_{11}$ and $\ell_2 := n_{01} - F_{00} + F_{11}$. Then for all populations with $n_{10} > n_{01}$,*

$$\begin{aligned}\mathrm{E}(\Delta \mid P_{12}, F_{00}, F_{11}) &= \mathrm{P}(S > \ell_1) \cdot \frac{2F_{00} - 2F_{11} - n_{10} + n_{01}}{3}\\ &\quad + \mathrm{P}(S = \ell_1) \cdot \frac{F_{00} - F_{11} + n_{01}}{3}\\ &\quad + \mathrm{P}(S > \ell_2) \cdot \frac{2F_{00} - n_{01}}{6}\\ &\quad + \mathrm{P}(S = \ell_2) \cdot \frac{F_{00}}{6}.\end{aligned}$$

Considering P_{12} and crossover is easy for $F_{00} = 0$ and $n_{10} > n_{01}$ as the potential drift is always non-positive.

Lemma 14. *For all $n_{10} > n_{01}$ and all $F_{11} \in \mathbb{N}_0$,*

$$\mathrm{E}(\Delta \mid P_{12}, F_{00} = 0, F_{11}) \leq 0.$$

Assembling the previous drift bounds under various conditions, we get the following drift bounds.

Lemma 15. *Assume $c \leq 56/9$, $n_{10} = n_{01}$ and $n_{11} \leq n - c$.*

$$\begin{aligned}&\mathrm{E}(\Delta \mid P_{11}, F_{0*} = 0) \cdot \mathrm{P}(P_{11}, F_{0*} = 0)\\ &+ \mathrm{E}(\Delta \mid P_{22}, F_{*0} = 0) \cdot \mathrm{P}(P_{22}, F_{*0} = 0)\\ &+ \mathrm{E}(\Delta \mid P_{12}, F_{00} = 0) \cdot \mathrm{P}(P_{12}, F_{00} = 0)\\ &\qquad \leq \begin{cases}\frac{2cn_{10}^2}{9n} & \text{if } n_{10} = n_{01}\\ -\frac{n_{01}}{12}\left(1 - \frac{c}{n}\right)^n & \text{if } n_{10} > n_{01}.\end{cases}\end{aligned}$$

Proof. For $n_{10} = n_{01}$, we argue that Lemma 12 implies

$$\mathrm{E}(\Delta \mid P_{12}, F_{00} = 0) \leq \frac{n_{01}}{9}\left(1 - \frac{c}{n}\right)^{n_{11}}.$$

This is obvious for $n_{10} \leq 1$; for $n_{10} \geq 2$ it follows from $\mathrm{P}(F_{11} = 1) = cn_{11}/n \cdot (1 - c/n)^{n-1} = cn_{11}/(n-c) \cdot (1 - c/n)^n \leq c\,(1 - c/n)^n$ and $1/8 + c/64 \leq 2/9 \leq n_{01}/9$ using $c \leq 56/9$.

By Lemma 7 and Lemma 12, along with $\mathrm{P}(P_{12}) = \mathrm{P}(P_{11} \cup P_{22}) = 1/2$ and $\mathrm{P}(F_{00} = 0) = (1 - c/n)^{n_{00}}$, the left-hand side is at most

$$\begin{aligned}&\frac{n_{10}}{18}\left(1 - \frac{c}{n}\right)^{n_{11}+n_{00}} - \frac{n_{10}}{18}\left(1 - \frac{c}{n}\right)^{n}\\ &= \frac{n_{10}}{18}\left(1 - \frac{c}{n}\right)^{n_{11}+n_{00}}\left(1 - \left(1 - \frac{c}{n}\right)^{n_{10}+n_{01}}\right)\\ &\leq \frac{n_{10}}{18}\left(1 - \frac{c}{n}\right)^{n_{11}+n_{00}}\left(1 - \left(1 - \frac{c(n_{10} + n_{01})}{n}\right)\right)\\ &= \frac{n_{10}}{18}\left(1 - \frac{c}{n}\right)^{n_{11}+n_{00}}\frac{c(n_{10} + n_{01})}{n} \leq \frac{2cn_{10}^2}{9n}.\end{aligned}$$

The bound for the case $n_{10} > n_{01}$ follows immediately from Lemmas 7 and 14, along with $\mathrm{P}(P_{11}) = 1/4$. □

6 POTENTIAL DRIFT WHEN ONE ZERO FLIPS

In this section we show that the drift is bounded by a term that yields the leading constant we are aiming for in our main result. Note that here we can afford to include error terms of lower order.

Lemma 16. *For all populations with $n_{10} = n_{01}$,*

$$\mathrm{E}(\Delta \mid P_{11}, F_{0*} = 1) \leq \mathrm{P}(F_{1*} = 0)\left(1 - \frac{n_{01}}{6}\right) + \mathrm{P}(F_{1*} = 1)\left(\frac{2}{9} - \frac{n_{01}}{9}\right)$$

$$\mathrm{E}(\Delta \mid P_{22}, F_{*0} = 1) \leq \mathrm{P}(F_{*1} = 0)\left(1 - \frac{n_{01}}{6}\right) + \mathrm{P}(F_{*1} = 1)\left(\frac{2}{9} - \frac{n_{01}}{9}\right).$$

For all populations with $n_{10} > n_{01}$,

$$\begin{aligned}&\mathrm{E}(\Delta \mid P_{11}, F_{0*} = 1)\\ &\leq \begin{cases}\frac{1}{3} + \frac{2}{3} \cdot \mathrm{P}(F_{1*} = 0) & \text{if } n_{01} = 0\\ \left(1 - \frac{n_{01}}{3}\right) \cdot \mathrm{P}(F_{1*} = 0) + \left(\frac{1}{3} - \frac{n_{01}}{3}\right) \cdot \mathrm{P}(F_{1*} = 1) & \text{otherwise.}\end{cases}\\ &\mathrm{E}(\Delta \mid P_{22}, F_{*0} = 1) \leq \frac{1}{3} \cdot \mathrm{P}(F_{*1} = 0) + \frac{1}{6} \cdot \mathrm{P}(F_{*1} = 1).\end{aligned}$$

Proof. First assume $n_{10} = n_{01}$ and consider the event P_{11}. Given $F_{0*} = 1$, the offspring is accepted with certainty if no 1-bit is flipped. With probability 1/2 the parent survives and then the new potential is at most $n_{11} + n_{10} + 1$. Consequently, the potential changes by at most $1 - n_{01}/3$ due to the loss of diversity. With the remaining probability 1/2, the potential increases by at most 1. The overall expected change in potential in this case is thus at most $1 - n_{01}/6$. If a single 1-bit also flips together with the 0-bit, then the offspring has the same fitness as both parents and the individual to be removed is selected uniformly at random. If the offspring is removed, the potential does not change. If the parent is removed, the potential increases by at most 1/3. If the other population member is removed, the potential increases by at most $1/3 - n_{10}/3$ due to the loss of diversity. The expected change in potential in this case is thus at most $2/9 - n_{10}/9$. If more than one 1-bits flip, then the offspring will have lower fitness than both other members and it will be rejected. Summing up the terms proves the first claim and the case of P_{22} follows analogously, considering F_{*0} and F_{*1} instead.

For $n_{10} > n_{01}$, given $F_{0*} = 1$ and that the fitter parent x^1 is selected twice, we consider separate cases according to the size of n_{01}. If $n_{01} = 0$, then no diversity can be lost. Thus, if no 1-bits flip, the potential increases by 1 because the new offspring has higher fitness than x^1 and x^2 is rejected. If at least one 1-bit flips, then the best fitness does not change and the potential increases by at most 1/3. Overall, for the case when $n_{01} = 0$, the potential changes by $\mathrm{P}(F_{1*} = 0) + 1/3 \cdot \mathrm{P}(F_{1*} > 0) = 1/3 + 2/3 \cdot \mathrm{P}(F_{1*} = 0)$. If $n_{01} > 0$ and no 1-bits flip, then the potential changes by at most $1 - n_{01}/3$ since the diversity is lost because x^2 is removed. If instead at least one 1-bit flips, then the potential changes by at most $1/3 - n_{01}/3$ since the best fitness does not change and the diversity may be lost if x^2 is removed. Since for $n_{01} > 0$ these terms are negative, and the offspring is accepted with probability 1 if $F_{1*} = 1$ the claim follows by summing up the two terms.

If x^2 is selected twice and no 1-bits are flipped, then the potential increases by at most 1/3 (i.e., if an n_{00} bit is flipped) since the parent is removed and the diversity is kept. If a single 1-bit is flipped then the potential increases again by at most 1/3. However, since the offspring has the same fitness as its parent, it is necessary that the parent is removed which happens with probability 1/2. If more than one 1-bits are flipped, then the offspring is rejected. Summing up the terms completes the proof. □

The proof of Lemma 16 has revealed a counterintuitive effect. A population of individuals with very different fitness values $f(x^1) \gg f(x^2)$ can have an advantage over a population where both members have the same fitness $f(x^1)$. This is because, conditioning on a 0-bit flipping, if the fitter parent is chosen twice, a near-arbitrary number of 1-bits can flip at the same time and the outcome may still be accepted. This increases the potential and explains why Lemma 16 contains an unexpectedly large potential drift in the case $n_{10} > n_{01} = 0$.

When $F_{11} \geq 2$, the drift under P_{12} is non-positive for $n_{10} \geq 10$, except for $F_{11} = n_{10-1}$, where it is exponentially small in n_{10}.

Lemma 17. *For all $n_{10} = n_{01}$, $n_{10} \geq 10$ and all $2 \leq i \leq n_{10} - 1$,*

$$\mathrm{E}(\Delta \mid P_{12}, F_{00} = 1, F_{11} = i) \leq \begin{cases} 5n_{10}^3 2^{-2n_{10}}/3 & \text{if } i = n_{10} - 1 \\ 0 & \text{otherwise.} \end{cases}$$

Using Lemmas 16 and 17 and considering the drift under P_{12} separately for $F_{11} \in \{0, 1\}$ and for all F_{11} when $n_{10} \leq 10$, we get:

Lemma 18. *Assume $c \leq 2.71$, $n_{10} + n_{01} \leq \log^2 n$, $n_{11} \geq n - n/\log^3 n$ and $n_{00} \geq n_{10} \log n$.*

$$\begin{aligned} &\mathrm{E}(\Delta \mid P_{11}, F_{0*} = 1) \cdot \mathrm{P}(P_{11}, F_{0*} = 1) \\ &+ \mathrm{E}(\Delta \mid P_{22}, F_{*0} = 1) \cdot \mathrm{P}(P_{22}, F_{*0} = 1) \\ &+ \mathrm{E}(\Delta \mid P_{12}, F_{00} = 1) \cdot \mathrm{P}(P_{12}, F_{00} = 1) \\ &\leq \begin{cases} \frac{c(2c+9)}{9e^c} \cdot \frac{n_{00}}{n} \cdot (1 + O(1/\log n)) & \text{if } n_{10} = n_{01} \\ \frac{c}{e^c}\left(\frac{1}{4} + \frac{c}{48} + \frac{e^c}{4} + O(1/\log n)\right)\frac{n_{00}}{n} & \text{if } n_{10} > n_{01} = 0 \\ \frac{c}{e^c}\left(\frac{1}{4} + \frac{c}{24} + \frac{e^c}{3} + O(1/n)\right)\frac{n_{00}}{n} & \text{if } n_{10} > n_{01} > 0. \end{cases} \end{aligned}$$

7 PUTTING EVERYTHING TOGETHER

Combining results from previous sections, for different numbers of flipping zeros, yields the following unconditional drift bound.

Lemma 19. *If $c \leq 1.422$, $n_{10} + n_{01} \leq \log^2 n$, $n_{11} \geq n - n/\log^3 n$ and $n_{00} \geq \log^5 n$, then*

$$\mathrm{E}(\Delta) \leq \frac{c(2c+9)}{9e^c} \cdot \frac{n_{00}}{n} \cdot (1 + O(1/\log n)).$$

Proof. This follows from adding drift bounds from Lemmas 6, 15 and 18. For $n_{10} = n_{01}$, Lemma 18 gives the stated bound. The terms $O(n_{10}^2/n) = O((\log^4 n)/n)$ from Lemma 15 and $O(n_{0*}/(n \log n)) = O(n_{00}/(n \log n))$ can both be absorbed in the $O(1/\log n)$ term since $n_{00}/n \geq \log^5 n/n$.

For $n_{10} > n_{01} = 0$, the bound from Lemma 18 is at most the bound from the same lemma for $n_{10} = n_{01}$ since

$$\frac{c}{e^c}\left(\frac{1}{4} + \frac{c}{48} + \frac{e^c}{4}\right) \leq \frac{c(2c+9)}{9e^c}$$

for $c \leq 1.422$. Then the claim follows as above.

For $n_{10} > n_{01} > 0$ note that Lemma 15 yields a negative upper bound of $-n_{01}/8 \cdot (1 - c/n)^n = -\Omega(1)$. Since $n_{00} \leq n/\log^3 n$, we obtain a negative drift bound if n is large enough. □

To translate the upper bound from Lemma 19 into a lower bound on the expected runtime, we use the lower-bound version of the multiplicative drift theorem [21, Theorem 2.2].

Theorem 20 (Multiplicative drift, lower bound, e. g., [21]). *Let X_t, $t \geq 0$, be a stochastic process, adapted to a filtration $\mathcal{F}_t$, over a state space $S \subseteq \mathbb{R}^{\geq x_{\min}}$, where $x_{\min} > 0$. Assume that X_t is non-increasing, i. e., $X_{t+1} \leq X_t$ for all $t \geq 0$. Let T be the smallest $t \geq 0$ such that $X_t \leq x_{\min}$. If there exist positive real numbers $\beta, \delta > 0$ such that for all $t < T$ it holds that*

(1) $\mathrm{E}(X_t - X_{t+1} \mid \mathcal{F}_t) \leq \delta X_t$
(2) $\mathrm{P}(X_t - X_{t+1} \geq \beta X_t) \leq \beta\delta/\ln(X_t)$

then

$$\mathrm{E}(T \mid \mathcal{F}_0) \geq \frac{1-\beta}{1+\beta} \frac{\ln(X_0/x_{\min})}{\delta}$$

Theorem 20 requires an upper bound on the drift of the potential function and a sufficiently small probability for large jumps of this value. Such large jumps can occur if the two individuals of the (2+1) GA have a large Hamming distance. Recall that Lemma 5 shows this to be unlikely.

The following lemma shows that a drift of the potential can be translated into a lower bound on the expected optimisation time.

This is not immediate since the potential function is a weighted combination of two quantities.

Lemma 21. *Let N_t denote the number of n_{00}-bits at time t of the (2+1) GA and T the first point in time where $n_{00} \le \log^5 n$. Assume that $n_{01}+n_{10} \le \log^2 n$ for all points in time before T, and $N_0 \ge \log^5 n$. If $\mathrm{E}(\varphi_{t+1} - \varphi_t \mid \varphi_t) \le \delta N_t$ for some $n^{-O(1)} \le \delta < 1$ and all $t < T$, then*

$$\mathrm{E}(T \mid \varphi_0) \ge (1 - O(1/\log n))\frac{\ln n - O(\ln\ln n)}{\delta}.$$

Proof. We introduce a distance function $\overline{\varphi}_t := n - \varphi_t = n_{00} + \frac{2}{3}n_{01}$ as the mirror image of our potential to obtain a function to be minimised, as required in Theorem 20. Moreover, we write $\Delta_t = \overline{\varphi}_t - \overline{\varphi}_{t+1}$. The key idea is to show that the statement on $\mathrm{E}(T \mid \varphi_0)$ holds under the slightly different drift condition

$$\mathrm{E}\big(\Delta_t \mid \overline{\varphi}_t\big) \le \delta\overline{\varphi}_t \tag{3}$$

and then to prove that the actual drift condition $\mathrm{E}(\varphi_{t+1} - \varphi_t \mid \varphi_t) \le \delta N_t$ leads to the same result, up to lower-order terms.

We consider the process from the first point in time where $\overline{\varphi}_t \le n/\log^3 n$, assume (3) to hold for all $t < T$ and estimate the remaining expected time to minimise the distance $\overline{\varphi}_t$. Lemma 5 and the facts that at most $\log n$ bits flip per generation and that the distance does not drop below $n/\log^3 n$ within the first $\log^2 n$ generations (each happening with probability $1 - n^{-\Omega(\log n)}$), imply that $\overline{\varphi}_0 \ge n/(2\log^3 n)$ with respect to our time count. We assume this to happen. By Lemma 5, with probability $1 - n^{-\Omega(\log n)}$ it holds that $n_{10} + n_{01} \le \log^2 n$ for any polynomial number of steps. Assuming this for a sufficiently long period obtained from applying Markov's inequality on $\mathrm{E}\big(T \mid \overline{\varphi}_0\big)$, our assumptions only change the bound on the expected value by a $1 - n^{-\Omega(\log n)}$ factor.

Clearly, since $n_{10} + n_{01} \le \log^2 n$, crossover followed by a neutral mutation can change the $\overline{\varphi}$-value by at most $\log^2 n$. Moreover, each mutation flips k or more bits with probability at most

$$\binom{n}{k}\left(\frac{c}{n}\right)^k \le \frac{c^k}{k!}; \tag{4}$$

in particular it flips at most $\log^2 n$ bits with probability $1-n^{-\Omega(\log n)}$. Adding up these effects, we arrive at $\mathrm{P}\Big(\overline{\varphi}_t - \overline{\varphi}_{t+1} \ge 2\log^2 n\Big) = n^{-\Omega(\log n)}$. The time to minimise the distance function is no smaller than the time to reach a distance of at most $x_{\min} := \log^5 n$ (we stop at this point to fulfill the condition on n_{00} in Lemma 19). Along with $\beta := 2/\log n$ and using $X_t := \overline{\varphi}_t$, we verify the second condition of Theorem 20 by estimating

$$\mathrm{P}(X_t - X_{t+1} \ge \beta X_t) \le$$

$$\mathrm{P}(X_t - X_{t+1} \ge \beta x_{\min}) = \mathrm{P}\Big(X_t - X_{t+1} \ge 2\log^2 n\Big) = n^{-\Omega(\log n)}.$$

Finally, we estimate this bound as $n^{-\Omega(n)} \le \beta\delta/\log n \le \beta\delta/\log(X_t)$ since both β and δ are at least inversely polynomial in n. Hence, $\mathrm{P}(X_t - X_{t+1} \ge \beta X_t) \le \beta\delta/\log(X_t)$, which satisfies the condition. Applying the theorem and recalling our assumption $\overline{\varphi}_0 \ge n/(2\log n)$,

$$\mathrm{E}(T \mid X_0) \ge \frac{1-\beta}{1+\beta}\cdot\frac{\ln(\overline{\varphi}_0/x_{\min})}{\delta} = (1 - O(1/\log n))\frac{\ln(n/\log^4 n)}{\delta}$$

which is $(1 - O(1/\log n))\frac{\ln(n) - O(\ln\ln n)}{\delta}$. Recall that the unconditional expected time is only by a factor $1 - n^{-\Omega(\log n)}$ smaller.

Finally, we relate N_t to the distance function $\overline{\varphi}_t$ and note first that $\overline{\varphi}_t \ge N_t$. Since, for $t < T$, our assumptions imply $\overline{\varphi}_t = n_{00} + (2/3)n_{01} \le (1 + 1/\log n)N_t$, the prerequisite

$$\mathrm{E}(\varphi_{t+1} - \varphi_t \mid \varphi_t) \le \delta N_t$$

along with the fact $\Delta_t = \varphi_{t+1} - \varphi_t$ imply

$$\mathrm{E}\big(\Delta_t \mid \overline{\varphi}_t\big) \le \delta(1 - 1/\log n)\overline{\varphi}_t.$$

Hence, (3) has been established with parameter $\delta(1 - 1/\log n)$ so that

$$\mathrm{E}(T \mid \varphi_0) \ge (1 - O(1/\log n))\frac{\ln(n/\log^4 n)}{(1 - 1/\log n)\delta},$$

which is again $(1 - O(1/\log n))\frac{\ln(n/\log^4 n)}{\delta}$ as claimed. □

We are now ready to prove our main result.

Proof of Theorem 2. With probability $1 - 2^{-\Omega(n)}$ the initial population has a maximum number of $(2/3)n$ one-bits. Since the number of one-bits crossover creates among the n_{10} and n_{01} bits follows a binomial distribution with parameters $n_{10} + n_{01}$ and $1/2$, Chernoff bounds imply that each crossover operation improves the OneMax-value by at most $n^{2/3}$ with probability $1 - 2^{-\Omega(n^{1/3})}$. The same holds for each mutation with constant rate c as seen in (4). Hence, the subsequent $2\log^2 n$ generations do not increase the OneMax-value to more than $(3/4)n$ with probability $1 - 2^{-\Omega(n^{1/3})}$. We consider the first point in time when $n_{11} \ge n - n/\log^3 n$. By the same arguments as before, we then have $n_{11} \le n - n/(2\log^3 n)$ with overwhelming probability. Now Lemma 5 is in force, implying that we can apply Lemma 21 with

$$\delta \le \frac{c(2c+9)}{9e^c n}\cdot(1 + O(1/\log n)).$$

according to Lemma 19. Hence, the expected optimization time is bounded from below by

$$\left(1 - O\left(\frac{1}{\log n}\right)\right)\frac{9e^c n\ln n - O(n\ln\ln n)}{c(2c+9)},$$

which is $\frac{9e^c}{c(2c+9)}\cdot n\ln n - O(n\ln\ln n)$ as claimed. □

8 CONCLUSIONS

Proving lower bounds for crossover-based GAs is a notoriously hard problem. We have provided such a lower bound for the (2+1) GA on OneMax through a careful analysis of a potential function that captures both the current best fitness and the potential for finding improvements through crossover combining different "building blocks" of good solutions. Our lower bound is tight up to small-order terms. This for the first time proves rigorously that populations are provably beneficial for standard steady-state genetic algorithms. We also identified the optimal mutation rate for the (2+1) GA as $c = (\sqrt{97} - 5)/(4n)$ for the considered range of mutation rates c/n.

Our lower bound applies for $c \le 1.422$ and an obvious open question is whether the leading constant in the expected runtime remains at $9e^c/(c(2c+9))$ when this threshold is exceeded.

Acknowledgements. This work was initiated at the Dagstuhl Seminar 19431 and supported by the EPSRC under grant EP/M004252/1.

REFERENCES

[1] E. Carvalho Pinto and C. Doerr. A simple proof for the usefulness of crossover in black-box optimization. In A. Auger, C. M. Fonseca, N. Lourenço, P. Machado, L. Paquete, and D. Whitley, editors, *Parallel Problem Solving From Nature - PPSN XV*, volume 11102 of *LNCS*, pages 29–41. Springer, 2018.

[2] D. Corus and P. S. Oliveto. Standard steady state genetic algorithms can hillclimb faster than mutation-only evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 22(5):720–732, 2018.

[3] D. Corus and P. S. Oliveto. On the benefits of populations for the exploitation speed of standard steady-state genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2019)*, pages 1452–1460, New York, NY, USA, 2019. ACM.

[4] D.-C. Dang, T. Friedrich, T. Kötzing, M. S. Krejca, P. K. Lehre, P. S. Oliveto, D. Sudholt, and A. M. Sutton. Escaping local optima using crossover with emergent diversity. *IEEE Transactions on Evolutionary Computation*, 22(3):484–497, 2018.

[5] D.-C. Dang, T. Friedrich, M. S. Krejca, T. Kötzing, P. K. Lehre, P. S. Oliveto, D. Sudholt, and A. M. Sutton. Escaping Local Optima with Diversity-Mechanisms and Crossover. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2016)*, pages 645–652. ACM Press.

[6] B. Doerr, E. Happ, and C. Klein. Crossover can provably be useful in evolutionary computation. *Theoretical Computer Science*, 425(0):17–33, 2012.

[7] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman, 1989.

[8] O. Gruder. The theory of risk. In *9th International Congress of Actuaries*, volume 2, page 222, 1930.

[9] T. Jansen and I. Wegener. On the analysis of evolutionary algorithms—a proof that crossover really can help. *Algorithmica*, 34(1):47–66, 2002.

[10] N. L. Johnson. A note on the mean deviation of the binomial distribution. *Biometrika*, 44(3-4):532–533, 12 1957.

[11] T. Kötzing, D. Sudholt, and M. Theile. How crossover helps in pseudo-Boolean optimization. In *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference (GECCO 2011)*, pages 989–996. ACM Press, 2011.

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

[13] P. K. Lehre and X. Yao. Crossover can be constructive when computing unique input–output sequences. *Soft Computing*, 15(9):1675–1687, 2011.

[14] J. Lengler. A general dichotomy of evolutionary algorithms on monotone functions. In A. Auger, C. M. Fonseca, N. Lourenço, P. Machado, L. Paquete, and D. Whitley, editors, *Parallel Problem Solving from Nature – PPSN XV*, volume 11102 of *LNCS*, pages 3–15. Springer, 2018.

[15] F. Neumann, P. S. Oliveto, G. Rudolph, and D. Sudholt. On the effectiveness of crossover for migration in parallel evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2011)*, pages 1587–1594. ACM Press, 2011.

[16] P. S. Oliveto and C. Witt. On the runtime analysis of the simple genetic algorithm. *Theoretical Computer Science*, 545:2–19, 2014.

[17] P. S. Oliveto and C. Witt. Improved time complexity analysis of the simple genetic algorithm. *Theoretical Computer Science*, 605:21–41, 2015.

[18] D. Sudholt. How crossover speeds up building-block assembly in genetic algorithms. *Evolutionary Computation*, 25(2):237–274.

[19] A. M. Sutton. Crossover can simulate bounded tree search on a fixed-parameter tractable optimization problem. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2018)*, pages 1531–1538, New York, NY, USA, 2018. ACM.

[20] A. M. Sutton and C. Witt. Lower bounds on the runtime of crossover-based algorithms via decoupling and family graphs. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2019)*, pages 1515–1522. ACM Press, 2019.

[21] C. Witt. Tight bounds on the optimization time of a randomized search heuristic on linear functions. *Combinatorics, Probability and Computing*, 22:294–318, 2 2013.