Theoretical Computer Science 816 (2020) 144–168

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The impact of lexicographic parsimony pressure for ORDER/MAJORITY on the run time

Benjamin Doerr [a], Timo Kötzing [b], J.A. Gregor Lagodzinski [b], Johannes Lengler [c,*]

[a] Laboratoire d'Informatique (LIX), CNRS, École Polytechnique, Institute Polytechnique de Paris, Palaiseau, France
[b] Hasso Plattner Institute, University of Potsdam, Germany
[c] ETH Zürich, Switzerland

ARTICLE INFO

Article history:
Received 26 July 2018
Received in revised form 8 January 2020
Accepted 9 January 2020
Available online 16 January 2020
Communicated by W. Banzhaf

Keywords:
Genetic programming
Bloat control
Theory
Runtime analysis

ABSTRACT

While many optimization problems work with a fixed number of decision variables and thus a fixed-length representation of possible solutions, genetic programming (GP) works on variable-length representations. A naturally occurring problem is that of bloat, that is, the unnecessary growth of solution lengths, which may slow down the optimization process. So far, the mathematical runtime analysis could not deal well with bloat and required explicit assumptions limiting bloat.

In this paper, we provide the first mathematical runtime analysis of a GP algorithm that does not require any assumptions on the bloat. Previous performance guarantees were only proven conditionally for runs in which no strong bloat occurs. Together with improved analyses for the case with bloat restrictions our results show that such assumptions on the bloat are not necessary and that the algorithm is efficient without explicit bloat control mechanism.

More specifically, we analyzed the performance of the $(1+1)$ GP on the two benchmark functions ORDER and MAJORITY. When using lexicographic parsimony pressure as bloat control, we show a tight runtime estimate of $O(T_{\text{init}} + n \log n)$ iterations both for ORDER and MAJORITY. For the case without bloat control, the bounds $O(T_{\text{init}} \log T_{\text{init}} + n(\log n)^3)$ and $\Omega(T_{\text{init}} + n \log n)$ (and $\Omega(T_{\text{init}} \log T_{\text{init}})$ for $n = 1$) hold for MAJORITY.[1]

1. Introduction

While much work on nature-inspired search heuristics focuses on representing problems with strings of a fixed length (simulating a genome), genetic programming considers trees of variable size. One of the main problems when dealing with a variable-size representation is the problem of *bloat*, meaning an unnecessary growth of representations, exhibiting many redundant parts and slowing down the search.

In this paper we study the problem of bloat from the perspective of run time analysis. We want to know how optimization proceeds when there is no explicit bloat control, which is a setting notoriously difficult to analyze formally: previous works were only able to give results conditional on strong assumptions on the bloat (such as upper bounds on the total

* Corresponding author.
E-mail address: johannes.lengler@inf.ethz.ch (J. Lengler).

[1] An extended abstract of the paper at hand has been published at GECCO 2017 [3].

https://doi.org/10.1016/j.tcs.2020.01.011
0304-3975/

bloat), see [24,16] for overviews. The only exception is the very recent work [12] continuing the line of research presented here.

We use advances from drift theory as well as other tools from the analysis of random walks to bound the behavior and impact of bloat, thus obtaining unconditional bounds on the expected optimization time even when no bloat control is active.

Our focus is on mutation-based genetic programming (GP) algorithms, which has been a fruitful area for deriving run time results in GP. We will be concerned with the problems Order and Majority as introduced in [7]. This is in contrast to other theoretical work on GP algorithms which considered the PAC learning framework [13], the Max-Problem [14], Boolean functions [4,17,22,19,20], the sorting problem for which three ways of bloat control were theoretically investigated [28–30] as well as the generalized Order and Majority versions using weights [24,27].

Individuals for Order and Majority are binary trees, where each inner node is labeled J (short for *join*, but without any associated semantics) and leaves are labeled with literal symbols; we call such trees *GP-trees* . The set of literal symbols is $\{x_i \mid i \leq n\} \cup \{\overline{x}_i \mid i \leq n\}$, where n is the number of variables. In particular, literal symbols are paired (x_i is paired with $\overline{x}_i$). We say that in a GP-tree t a leaf u comes *before* a leaf v if u comes before v in an in-order parse of the tree.

For the Order problem fitness is assigned to GP-trees as follows: we call a variable i *expressed* if there is a leaf labeled x_i and all leaves labeled $\overline{x}_i$ do not come before that leaf. The fitness of a GP-tree is the number of its expressed variables i.

For the Majority problem, fitness is assigned to GP-trees as follows. We call a variable i *expressed* if there is a leaf labeled x_i and there are at least as many leaves labeled x_i as there are leaves labeled $\overline{x}_i$ (the positive instances are in the majority). Again, the fitness of a GP-tree is the number of its expressed variables i. The two functions Order and Majority capture two important aspects of GP: variable length representations and that any given functionality can be achieved by many different representations. However, the tree-structure, typically crucial in GP problems, is completely unimportant for the two problems.

A first run time analysis of genetic programming on Order and Majority was conducted in [6]. This work considered the algorithm $(1+1)$ GP proceeding as follows. A single operation on a GP-tree t chooses a leaf u of t uniformly at random and randomly either relabels this leaf (to a random literal symbol), deletes it (i.e. replacing the parent of u with the sibling of u) or inserts a leaf here (i.e., replaces u with an inner node with one randomly labeled child and u as the other child, in random order). The $(1+1)$ GP is provided with a parameter k which determines how many such operations make up an atomic mutation; in the simplest case with $k=1$, but a random choice of $k = 1 + \text{Pois}(1)$ (where Pois(1) denotes the Poisson distribution with parameter $\lambda = 1$) is also frequently considered. The $(1+1)$ GP then proceeds in generations with a simple mutation/selection scheme (see Algorithm 1).

In this paper we consider a version of bloat control for this algorithm that was introduced in [18] as *lexicographic parsimony pressure*. Here the algorithm always prefers the smaller of two trees, given equal fitness. For this [23] was able to give tight bounds on the optimization time in the case of $k=1$: in this setting no new redundant leaves can be introduced. The hard part is now to give an analysis when $k = 1+\text{Pois}(1)$, where bloat can be reintroduced whenever a fitness improvement is achieved (without fitness improvements, only smaller trees are acceptable). With a careful drift analysis, we show that in this case we get an (expected) optimization time of $O(T_{\text{init}} + n \log n)$, which is tight (see Theorem 4.1). Previously, no bound was known for Majority and the bound of $O(n^2 \log n)$ for Order required a condition on the initialization.

Without such bloat control it is much harder to derive definite bounds. From [6] we have the conditional bounds of $O(nT_{\max})$ for Order using either $k=1$ or $k = 1 + \text{Pois}(1)$, where $T_{\max}$ is an upper bound on the maximal size of the best-so-far tree in the run (thus, these bounds are conditional on these maxima not being surpassed). For Majority and $k=1$ [6] gives the conditional bound of $O(n^2 T_{\max} \log n)$. We focus on improving the bound for Majority and obtain a bound of $O(T_{\text{init}} \log T_{\text{init}} + n \log^3 n)$ for both $k=1$ and $k = 1 + \text{Pois}(1)$ (see Theorem 5.2). The proof of this theorem requires significant machinery for bounding the extent of bloat during the run of the optimization. Thus, our results show that without bloat control the algorithms considered are at most a few logarithmic factors worse than with bloat control. We summarize the best known bounds (as either given by us in this paper or by others) in Table 1.

The remainder of the paper is structured as follows. In Section 2 we will give a short introduction to the studied algorithm. In Section 3 the main tool for the analysis is explained, that is drift analysis. Here we state a selection of known theorems as well as two new ones: Theorem 3.6 gives lower tail bounds in the case of weak drift, in which the expected hitting time is dominated by unlikely events; and Theorem 3.7 gives a lower bound conditional on a multiplicative drift with a bounded step size. In Section 4 we will study the case of bloat control given $k = 1 + \text{Pois}(1)$ operations in each step. Subsequently we will study Majority without bloat control in Section 5. Section 6 concludes this paper.

2. Preliminaries

In this section we formalize the concepts introduced in Section 1. We consider tree-based genetic programming, where a possible solution to a given problem is given by a syntax tree. The inner nodes of such a tree are labeled by function symbols from a set F_S and the leaves of the tree are labeled by terminals from a set T.

We analyze the problems Order and Majority, where the only available function is the join operator (denoted by J). The terminal set X consists of $2n$ literals, where $\overline{x}_i$ is the complement of x_i:

- $F_S := \{J\}$, J has arity 2,

Table 1
Summary of best known bounds. Note that $T_{\max}$ denotes the maximal size of the best-so-far tree in the run until optimization finished (we consider bounds involving $T_{\max}$ as conditional bounds). All bounds are to be understood as complexities, that means that upper bounds hold for arbitrary initializations of the algorithms, while lower bounds are for worst-case initialization.

Problem	Without bloat control	Bloat control
Order, $k = 1$	$O(nT_{\max})$, [6]	$\Theta(T_{\text{init}} + n \log n)$, [23]
Order, $k = 1 + \text{Pois}(1)$	$O(nT_{\max})$, [6]	$\Theta(T_{\text{init}} + n \log n)$, Theorem 4.1
Majority, $k = 1$	$O(T_{\text{init}} \log T_{\text{init}} + n \log^3 n)$, Theorem 5.2, $\Omega(T_{\text{init}} \log T_{\text{init}})$, $n = 1$, Theorem 5.1 $\Omega(T_{\text{init}} + n \log n)$, Theorem 5.1	$\Theta(T_{\text{init}} + n \log n)$, [23]
Majority, $k = 1 + \text{Pois}(1)$	$O(T_{\text{init}} \log T_{\text{init}} + n \log^3 n)$, Theorem 5.2, $\Omega(T_{\text{init}} \log T_{\text{init}})$, $n = 1$, Theorem 5.1 $\Omega(T_{\text{init}} + n \log n)$, Theorem 5.1	$\Theta(T_{\text{init}} + n \log n)$, Theorem 4.1

Given a GP-tree t, mutate t by applying HVL-Prime. For each application, choose uniformly at random one of the following three options.

substitute	Choose a leaf uniformly at random and substitute it with a leaf in X selected uniformly at random.
insert	Choose a node $v \in X$ and a leaf $u \in t$ uniformly at random. Substitute u with a join node J, whose children are u and v, with the order of the children chosen uniformly at random.
delete	Choose a leaf $u \in t$ uniformly at random. Let v be the sibling of u. Delete u and v and substitute their parent J by v.

Fig. 1. Mutation operator HVL-Prime.

- $X := \{x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\}$.

For a given syntax tree t, the value of the tree is computed by parsing the tree in-order and generating the set S of *expressed* variables in this way. For Order a variable i is expressed if a literal x_i is present in t and there is no $\overline{x}_i$ that is visited in the in-order parse before the first occurrence of x_i. For Majority a variable i is expressed if a literal x_i is present in t and the number of literals x_i is at least the number of literals $\overline{x}_i$. We associate with each tree t the complexity C, which denotes the number of nodes t contains. Given a function F, we aim to generate an instance t maximizing F.

In this paper we consider simple mutation-based genetic programming algorithms which use a modified version of the *Hierarchical Variable Length* (HVL) operator ([25], [26]) called *HVL-Prime* as discussed in [6]. HVL-Prime allows to produce trees of variable length by applying three different operations: insert, delete and substitute (see Fig. 1). Each application of HVL-Prime chooses one of these three operations uniformly at random, and several such operations may be applied to create an offspring.

Based on this mutation operator, we consider the $(1 + 1)$ GP as follows. It starts with a given initial tree with T_{init} leaves and tries to improve its fitness iteratively. In each iteration, the number of mutation steps k is chosen according to a fixed distribution; important options for this distribution are (i) constantly 1 and (ii) $1 + \text{Pois}(1)$, where $\text{Pois}(\lambda)$ denotes the Poisson distribution with parameter λ. The choices for k in the different iterations are independent. The $(1 + 1)$ GP then produces an offspring from the best-so-far individual by applying HVL-Prime k times in a row. Importantly, we consider two variants of the selection operator, one without and one with bloat control, where we employ *lexicographic parsimony pressure* [18] as bloat control. The first one replaces the parent by the offspring if the offspring has at least the same fitness. For the second one, an offspring with strictly larger fitness is always accepted and an offspring of strictly smaller fitness is always rejected. If parent and offspring have the same fitness then the offspring is accepted if and only if its complexity C (see Fig. 2) is at most as large as the complexity of the parent. Thus the complexity serves as second order term to break ties. Equivalently, it would be possible to include the complexity as second order term into the fitness function, instead of incorporating it by the selection operator. To summarize, we study the following two algorithms, which differ only if parent and offspring are of equal fitness:

- The $(1 + 1)$ GP without bloat control, which accepts every offspring of equal fitness regardless of the complexity.

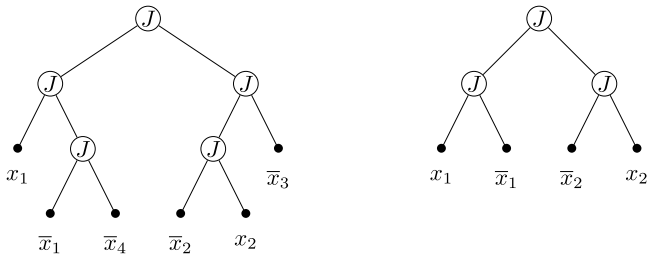

Fig. 2. Two GP-trees with the same fitness. For Order the fitness is 1 since only the first variable occurs with a non-negated literal first. For Majority the fitness is 2, since the variable 1 and 2 have one literal x_i and also one literal $\overline{x}_i$. However, the left one has complexity 11 whereas the other has complexity 7.

- The $(1+1)$ GP with bloat control, which accepts an offspring of equal fitness if and only if it doesn't increase the complexity.

Each of these algorithms comes with two variants, depending on whether we choose $k = 1$ or $1 + \text{Pois}(1)$.

Algorithm 1: The $(1+1)$ GP with bloat control. In the version without bloat control, the If-condition is simply replaced by $f(t') \geq f(t)$.

```
Let t be the initial tree;
while optimum not reached do
    t' ← t;
    Choose k;
    for i = 1 to k do
        t' ← mutate(t');
    if f(t') > f(t) or (f(t') = f(t) ∧ C(t') ≤ C(t)) then t ← t'
```

3. Drift theorems

In this section we collect theorems on stochastic processes that we will use in the proofs. We apply the standard Landau notation $O(\cdot)$, $o(\cdot)$, $\Omega(\cdot)$, $\omega(\cdot)$, $\Theta(\cdot)$ as detailed in [1].

Theorem 3.1 *(Chernoff Bound [5]). Let $X_1, \dots, X_n$ be independent random variables that take values in $[0,1]$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbb{E}[X]$. Then for all $0 \leq \delta \leq 1$,*

$$\Pr[X \leq (1-\delta)\mu] \leq e^{-\delta^2\mu/2},$$

and

$$\Pr[X \geq (1+\delta)\mu] \leq e^{-\delta^2\mu/3}.$$

If instead $X_1, \dots, X_n$ are independent random variables with mean zero that take values in $[-1,1]$, then for all $0 \leq \delta \leq 1$,

$$\Pr[|X| \geq \delta n] \leq 2e^{-\delta^2 n/2}.$$

We will apply a variety of drift theorems to derive the results of this paper. *Drift*, in this context, describes the *expected change* of the best-so-far solution within one iteration with respect to some *potential*. In later proofs we will define potential functions on best-so-far solutions and prove bounds on the drift; these bounds then translate to expected run times with the use of the drift theorems from this section. We use formulations from [15] because they do not require finite search spaces, and they do not require that the potential forms a Markov chain. Instead, we will have random variables Z_t (the current GP-tree) that follow a Markov chain, and the potential is some function of Z_t. We start with a theorem for *additive drift*.

Theorem 3.2 *(Additive Drift [9], formulation of [15]). Let $(Z_t)_{t \in \mathbb{N}_0}$ be random variables describing a Markov process with state space $\mathcal{Z}$, and with a potential function $\alpha : \mathcal{Z} \to S \subseteq [0, \infty)$, and assume $\alpha(Z_0) = s_0$. Let $T := \inf\{t \in \mathbb{N}_0 \mid \alpha(Z_t) = 0\}$ be the random variable that denotes the earliest point in time $t \geq 0$ such that $\alpha(Z_t) = 0$. If there exists $c > 0$ such that for all $z \in \mathcal{Z}$ with $\alpha(z) > 0$ and for all $t \geq 0$ we have*

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

then

$$\mathbb{E}[T] \leq \frac{s_0}{c}.$$

We will use the following *variable drift theorem*, an extension of the variable drift theorem from [10, Theorem 4.6].

Theorem 3.3 *(Variable Drift [15]). Let $(Z_t)_{t\in\mathbb{N}_0}$ be a Markov chain with state space $\mathcal{Z}$ and with a potential function $\alpha : \mathcal{Z} \to S \subseteq \{0\} \cup [s_{\min}, \infty)$ for some $s_{\min} > 0$. Assume $\alpha(Z_0) = s_0$, and let $T := \inf\{t \in \mathbb{N}_0 \mid \alpha(Z_t) = 0\}$ be the random variable that denotes the first point in time $t \in \mathbb{N}$ for which $X_t = 0$. Suppose furthermore that there exists a positive, increasing function $h : [s_{\min}, \infty) \to \mathbb{R}^+$ such that for all $z \in \mathcal{Z}$ with $\alpha(z) > 0$ and all $t \geq 0$ we have*

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

Then

$$\mathbb{E}[T] \leq \frac{1}{h(1)} + \int_1^{s_0} \frac{1}{h(u)} du.$$

The most important special case is for *multiplicative drift*, which was developed in [2]. We again give the version from [15]

Theorem 3.4 *(Multiplicative Drift [15]). Let $(Z_t)_{t\in\mathbb{N}_0}$ be a Markov chain with state space $\mathcal{Z}$ and with a potential function $\alpha : \mathcal{Z} \to S \subseteq \{0\} \cup [s_{\min}, \infty)$ for some $s_{\min} > 0$, and assume $\alpha(Z_0) = n$. Let $T := \inf\{t \in \mathbb{N}_0 : \alpha(Z_t) = 0\}$ be the random variable that denotes the first point in time $t \in \mathbb{N}$ for which $X_t = 0$. Assume that there is $\delta > 0$ such that for all $z \in \mathcal{Z}$ with $\alpha(z) > 0$ and for all $t \geq 0$ we have*

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

Then for all $k > 0$

$$\Pr\left[T > \left\lceil \frac{\log(n/s_{\min}) + k}{\delta} \right\rceil\right] \leq e^{-k},$$

and

$$\mathbb{E}[T] \leq \frac{1 + \log(n/s_{\min})}{\delta}.$$

For bloat estimation we need a lower bound drift theorem in the regime of weak additive drift. To illustrate what we mean by "weak drift", consider a biased random walk on $\mathbb{N}$, starting at n, which makes in each round a step to the left with probability $(1 + \varepsilon)/2$, and a step to the right with probability $(1 - \varepsilon)/2$. This random walk has a drift of ε towards zero, so the hitting time T of zero satisfies $\mathbb{E}[T] = n/\varepsilon$ by additive drift analysis. However, the expectation may not give the full picture. If $\varepsilon \gg 1/n$ (strong drift), then indeed T is concentrated around its expectation, and this generalizes to other random walks (Theorem 3.5 below). But if $\varepsilon \ll 1/n$ (weak drift), then with high probability the biased random walk hits zero after roughly $O(n^2) \ll n/\varepsilon$ steps, since even an unbiased random walk hits zero after roughly $O(n^2)$ steps. Thus the typical length of a random walk is much shorter than its expectation. This is not contradictory, it just means that the expectation is dominated by very unlikely events that contribute high values. Thus the expectation is somewhat misleading in this case. On the other hand, it is not hard to come up with other random walks with weak drift in which T *is* concentrated around its expectation. For example, the deterministic walk which decreases in each round by ε trivially has this property. Thus, in the case of weak drift, knowing the drift is sufficient to derive $\mathbb{E}[T]$, but it is not sufficient to determine typical values of T. In the random walks we encounter, we will need that unlikely events indeed contribute substantially to the expectation.

We start with a theorem (Theorem 3.5) which is tight for strong drift, which follows from Theorem 10 and 12 in [11]. Theorem 3.5 is not directly applicable to our situation, since it does only tight bounds in the regime of weak drift. Nevertheless, we can use it to prove lower bounds on the tail probabilities for the regime of weak drift, see Theorem 3.6 below.

Theorem 3.5 *(Strong Additive Drift, Lower Tail Bound, follows from [11, Theorem 10,12]). Let $(Z_t)_{t\in\mathbb{N}_0}$ be random variables describing a Markov process with state space $\mathcal{Z}$, and with a potential function $\alpha : \mathcal{Z} \to S \subseteq \mathbb{N}$, and assume $\alpha(Z_0) = s_0$. Suppose further that there exist $\delta, \rho, r > 0$ such that for all $z \in \mathcal{Z}$ such that $\alpha(z) > 0$, all $k \in \mathbb{N}_0$, and all $t \geq 0$,*

1. $\Pr[|X_t - X_{t+1}| > k \mid Z_t = z] \leq \frac{r}{(1+\delta)^k}$.

2. $\mathbb{E}[X_t - X_{t+1} \mid Z_t = z] \le \rho$.

Then, for all $x \ge 0$, if $T := \inf\{t \in \mathbb{N}_0 \mid \alpha(Z_t) = 0\}$ is the random variable that denotes the earliest point in time $t \ge 0$ such that $\alpha(Z_t) = 0$.

$$\Pr\left[T \le \frac{s_0 - x}{\rho}\right] \le \exp\left\{-\frac{\delta x}{8} \min\left\{1, \frac{\delta^2 \rho x}{32 r s_0}\right\}\right\}. \tag{2}$$

The next theorem gives a lower bound on hitting times of random walks even if we start close to the goal, provided that the drift towards the goal is weak. We remark that the statement on the expectation is similar to other lower bounds for additive drift [11], but the existing tail bounds are tailored to the regime of strong drift, and are thus not tight in our case. We prove the theorem by martingale theory.

Theorem 3.6 *(Weak Additive Drift, Lower Bounds). For every $\delta, C > 0$ there exists $\varepsilon > 0$ such that the following holds for all $N \ge 1$. Let $(Z_t)_{t \in \mathbb{N}_0}$ be random variables describing a Markov process with state space $\mathcal{Z}$, and with a potential function $\alpha : \mathcal{Z} \to S \subseteq [0, \infty)$. We denote $X_t := \alpha(Z_t)$. Assume $\alpha(Z_0) = s_0$ and that the following conditions hold for all $t \ge 0$ and all $z, z' \in \mathcal{Z}$ such that $\alpha(z) \le N$.*

(i) Weak Drift. $\mathbb{E}[X_t - X_{t+1} \mid Z_t = z] \le C/N$.
(ii) Small Steps. $\Pr[|X_t - X_{t+1}| \ge k \mid Z_t = z'] \le (1+\delta)^{-k}$.
(iii) Initial Increase. $\Pr[X_{t+1} > X_t \mid Z_t = z] \ge \delta$.

Then for every $0 \le x < s_0 \le \varepsilon N$, if $T := \min\{\tau \ge 0 \mid X_t \le x\}$ is the hitting time of $\{0, 1, \dots, x\}$ for X_t, then

$$\mathbb{E}[T] \ge \varepsilon(s_0 - x)N \tag{3}$$

and

$$\Pr[T \ge \varepsilon N^2] \ge \frac{\varepsilon}{N}. \tag{4}$$

Proof. Note that for any constant $N_0 = N_0(\delta, C)$, the statement is trivial for all $N \le N_0$ if ε is sufficiently small. Hence, we may always assume that N is large compared to δ and C.

Without loss of generality, we may assume that $|\mathbb{E}[X_{t+1} - X_t \mid Z_t = z]| \le C/N$, which is stronger than (i). If this does not hold a priori, then we may couple the process X_t to a process X'_t which makes the same steps as X_t (i.e., $X_{t+1} - X_t = X'_{t+1} - X'_t$), with one exception: if $\mathbb{E}[X_t - X_{t+1} \mid Z_t = z] < -C/N$ at any point in time, then with some (additional) probability p_t we choose Z_{t+1} such that X_{t+1} is smaller, thus increasing the drift. More precisely, we choose p_t in such a way that $-C/N \le \mathbb{E}[X_t - X_{t+1} \mid Z_t = z] \le C/N$. Then $X'_t \le X_t$ for all $t \ge 0$, so it suffices to prove the statement for X'_t. To keep notation simple, we will assume that we do not need to modify X_t in the remainder.

We rescale $\tilde{X}_t := X_t - x$ and consider the drift of $\tilde{X}_t^2$. Let $p_i := \Pr[X_{t+1} - X_t = i \mid Z_t = z]$ for all $i \in \mathbb{Z}$. Then

$$\begin{aligned}\mathbb{E}[\tilde{X}_{t+1}^2 - \tilde{X}_t^2 \mid Z_t = z] &= \sum_{i \in \mathbb{Z}} p_i (\tilde{X}_t + i)^2 - \tilde{X}_t^2 = \sum_{i \in \mathbb{Z}} p_i (2\tilde{X}_t i + i^2) \\ &= 2\tilde{X}_t \mathbb{E}[X_{t+1} - X_t \mid Z_t = z] + \sum_{i \in \mathbb{Z}} p_i i^2.\end{aligned}$$

Note that we have $\sum_{i \in \mathbb{Z}} p_i i^2 \ge p_1 \ge \delta$ by (i) and $\sum_{i \in \mathbb{Z}} p_i i^2 \le \sum_{i \in \mathbb{Z}} (1+\delta)^{|i|} i^2 \in O(1)$ by (ii). Together with Condition (i), we have for all $0 \le \tilde{X}_t \le \delta N/(4C)$,

$$\delta/2 \le \mathbb{E}[\tilde{X}_{t+1}^2 - \tilde{X}_t^2 \mid Z_t = z] \le O(1). \tag{5}$$

Let t_0 be the (random) time when the process $\tilde{X}_t$ (started at $\tilde{X}_0 = s_0 - x$) for the first time leaves the interval $I = [1, \delta N/(4C) - x]$ on either side. We note that $t_0 \le T$ holds. Let p_ℓ and p_r be the probabilities that the process leaves the interval on the left (that is, at 0 or lower) and on the right (that is, at $\lfloor \delta N/(4C) - x \rfloor + 1$ or higher), respectively. By (ii) if the process leaves I on the right side, then the expectation of $\tilde{X}_t$ in this case is at most $\delta N/(2C)$; recall that we assumed N to be large. Similarly, if it leaves I on the left, then the expectation of $\tilde{X}_t$ is at least $-D$ for some constant $D > 0$.

By (5) there is a constant $D > 0$ such that the process $Y_t := \tilde{X}_t^2 - Dt$ has a negative drift in the interval I. Hence, using that t_0 is a stopping time we obtain from the optional stopping theorem [8]

$$\begin{aligned}(s_0 - x)^2 = \mathbb{E}[Y_0] \ge \mathbb{E}[Y_{t_0}] &\ge p_r \left(\frac{\delta N}{4C} - x\right)^2 - p_\ell D - D\mathbb{E}[t_0] \\ &\ge p_r \left(\frac{\delta N}{8C}\right)^2 - D - D\mathbb{E}[t_0].\end{aligned} \tag{6}$$

Similarly, we regard the process $U_t = \tilde{X}_t + Ct/N$. By (i) it has a non-negative drift for $t < t_0$. Hence, we obtain

$$s_0 - x = \mathbb{E}[U_0] \le \mathbb{E}[U_{t_0}] \le p_r \frac{\delta N}{2C} + \frac{C\mathbb{E}[t_0]}{N}. \tag{7}$$

This yields a lower bound of $p_r \delta N^2/(2C) \ge (s_0 - x)N - C\mathbb{E}[t_0]$ for p_r. Together with (6) we obtain

$$\mathbb{E}[t_0] \ge \frac{(s_0 - x)\,(\delta N/(2C) - 16(s_0 - x)) - 16D}{16D + \delta/2}, \tag{8}$$

which proves the bound on the expectation (3) since $s_0 - x \le \varepsilon N$.

For the tail bound (4) we reverse the previous argument. By (5) the process $U_t := \tilde{X}_t^2 - \delta t/2$ has a non-negative drift in the interval I. If $\tilde{X}_t$ leaves I on the right side then due to (ii) the expectation of $\tilde{X}_t^2$ is at most $(\delta N/(2C))^2$. Hence, by the optional stopping theorem

$$(s_0 - x)^2 = \mathbb{E}[U_0] \le \mathbb{E}[U_{t_0}] \le p_r \left(\frac{\delta N}{2C}\right)^2 - \frac{\delta}{2}\mathbb{E}[t_0] \overset{(3)}{\le} p_r \left(\frac{\delta N}{2C}\right)^2 - \frac{\delta}{2}\varepsilon(s_0 - x)N.$$

Solving for p_r shows that $p_r \in \Omega(1/N)$ whenever $s_0 - x \le \delta\varepsilon/4N$. Note that we may assume the latter condition by decreasing the ε in the theorem. (Despite the formulation, it is obviously sufficient to prove (3) for ε and (4) for $\varepsilon' := \delta\varepsilon/4$.) Then with probability $\Omega(1/N)$ we have $X_t > \delta N/(4C)$ for some $t \ge 0$. However, starting from this X_t by Theorem 3.5 with probability $\Omega(1)$ we need at least $\Omega(N^2)$ additional steps to return to $x < \varepsilon N$ if $\varepsilon < \delta/(4C)$. This proves (4). □

For our lower bounds we need the following new drift theorem, which allows for non-monotone processes (in contrast to, for example, the lower bounding multiplicative drift theorem from [31]), but requires an absolute bound on the step size.

Theorem 3.7 *(Multiplicative Drift, lower bound, bounded step size). Let $(Z_t)_{t\in\mathbb{N}_0}$ be random variables describing a Markov process with state space $\mathcal{Z}$ with a potential function $\alpha : \mathcal{Z} \to S \subseteq (0, \infty)$, for which we assume $\alpha(Z_0) = s_0$. Let $\kappa > 0$, $s_{\min} \ge \sqrt{2}\kappa$ and let $T := \inf\{t \in \mathbb{N}_0 \mid \alpha(Z_t) \le s_{\min}\}$ be the random variable denoting the earliest point in time $t \ge 0$ such that $\alpha(Z_t) \le s_{\min}$. If there exists a positive real $\delta > 0$ such that for all $z \in \mathcal{Z}$ with $\alpha(z) > s_{\min}$ and all $t \ge 0$ it holds*

1. $|\alpha(Z_t) - \alpha(Z_{t+1})| \le \kappa$, *and*
2. $\mathbb{E}[\alpha(Z_t) - \alpha(Z_{t+1}) \mid Z_t = z] \le \delta\alpha(z)$,

then

$$\mathbb{E}[T] \ge \frac{1 + \ln(s_0) - \ln(s_{\min})}{2\delta + \frac{\kappa^2}{s_{\min}^2 - \kappa^2}}.$$

Proof. We concatenate α with a second potential function g turning the multiplicative bound of the expected drift into an additive bound enabling us to apply the additive drift theorem. Let

$$g(s) := 1 + \ln\left(\frac{s}{s_{\min}}\right)$$

and $g(0) := 0$. Furthermore, let $X_t := \alpha(Z_t)$ and $V_t := g(X_t) = g(\alpha(Z_t))$. It follows that V_t is a stochastic process over the search space $R = g(\alpha(\mathrm{Z})) \cup \{0\}$. We observe that T is also the first point in time $t \in \mathbb{N}$ such that $V_t \le 1$. Since $s_{\min}$ is a lower bound on X_t, $s_{\min} - \kappa$ is a lower bound on X_{t+1}. Thus, $X_{t+1} > 0$ as well as $V_{t+1} > 0$. We derive

$$V_t - V_{t+1} = \ln\left(\frac{X_t}{X_{t+1}}\right).$$

Therefore, due to Jensen's inequality we obtain

$$\mathbb{E}[V_t - V_{t+1} \mid Z_t = z] \le \ln\left(\mathbb{E}\left[\frac{X_t}{X_{t+1}} \,\middle|\, Z_t = z\right]\right).$$

The value of X_{t+1} can only be in a κ-interval around X_t due to the bounded step size. For all $i \ge 0$ let p_i be the probability that $X_t - X_{t+1} = i$ and let q_i be the probability that $X_t - X_{t+1} = -i$. Let $z \in \mathcal{Z}$ and $s := \alpha(z)$. We note that $p_0 = q_0$ and obtain by counting twice the instance of a step size of 0

$$\mathbb{E}\left[\frac{X_t}{X_{t+1}}\,\middle|\, Z_t = z\right] \le \left(\sum_{i=0}^{\kappa} \frac{s}{s-i}\, p_i + \frac{s}{s+i}\, q_i\right) = \left(\sum_{i=0}^{\kappa} s\,\frac{p_i(s+i)+q_i(s-i)}{s^2-i^2}\right)$$
$$\le \left(\sum_{i=0}^{\kappa} s\,\frac{p_i(s+i)+q_i(s-i)}{s^2-\kappa^2}\right) = \left(\frac{s^2}{s^2-\kappa^2} + \sum_{i=0}^{\kappa} \frac{s(ip_i - iq_i)}{s^2-\kappa^2}\right),$$

where the last equality comes from summing all non-zero probabilities for a step size, i.e. $\sum p_i + q_i = 1$. The same holds for X_t since $s_{\min} \ge \sqrt{2}\kappa$. It follows that $X_t^2 - \kappa^2 \ge 1/2 X_t^2$ and this yields

$$\mathbb{E}\left[\frac{X_t}{X_{t+1}}\,\middle|\, Z_t = z\right] \le \left(\frac{s^2}{s^2-\kappa^2} + \frac{2}{s}\sum_{i=0}^{\kappa} ip_i - iq_i\right) = \left(1 + \frac{\kappa^2}{s^2-\kappa^2} + \frac{2}{s}\sum_{i=0}^{\kappa} ip_i - iq_i\right).$$

Since the remaining sum in the log-term is the difference of X_t and X_{t+1} multiplied by the probability for the step size, we obtain

$$\mathbb{E}[V_t - V_{t+1} \mid X_t = s] \le \ln\left(1 + \frac{\kappa^2}{X_t^2-\kappa^2} + 2\mathbb{E}\left[\frac{X_t - X_{t+1}}{X_t}\,\middle|\, Z_t = z\right]\right)$$
$$\le 2\mathbb{E}\left[\frac{X_t - X_{t+1}}{X_t}\,\middle|\, Z_t = z\right] + \frac{\kappa^2}{X_t^2-\kappa^2} \le 2\delta + \frac{\kappa^2}{X_t^2-\kappa^2}.$$

Finally, we apply the additive drift theorem and deduce

$$\mathbb{E}[T] \ge \frac{V_0}{2\delta + \frac{\kappa^2}{s_{\min}^2-\kappa^2}} = \frac{1 + \ln(s_0) - \ln(s_{\min})}{2\delta + \frac{\kappa^2}{s_{\min}^2-\kappa^2}}. \quad \square$$

We conclude this section with the following lemma on the occupation probability of a random walk between two states.

Lemma 3.8. *Let $\delta \ge 0$, and let $r \ge b \ge 0$. Consider a time-discrete random walk $(X_t)_{t\in\mathbb{N}}$ with two states A and B, adapted to some filtration $\mathcal{F}_t$. For any $t \ge 0$, let $S_t := \min\{t' \ge 0 \mid X_{t+t'} = A\}$ be the number of rounds to reach A for the next time after t. Suppose that*

1. $\Pr[X_{t+1} = B \mid \mathcal{F}_t, X_t = A] \ge \delta$ *for all* $t \ge 0$.
2. *There exists $s \ge 0$ such that for all $t \ge 0$,*

$$\Pr[S_t \ge s \mid \mathcal{F}_t, X_t = B, X_{t-1} = A] \ge \frac{b}{s}.$$

Then, if $N_A(r) := |\{1 \le t \le r \mid X_t = A\}|$ denotes how many of the first r round we spend in A, we have

$$\mathbb{E}[N_A(r)] \le \frac{2r}{b\delta},$$

and

$$\Pr\left[N_A(r) > \frac{4r}{b\delta}\right] \le e^{-r/(2s)}.$$

We remark that Condition (2) cannot be replaced by the weaker condition $\mathbb{E}[S_t \mid \mathcal{F}_t, X_t = B, X_{t-1} = A] \ge b$, not even for the statement on the expectation. For example, for $r \gg b \gg 1$ set $S_t := r^2$ with probability b/r^2, and $S_t := 1$ otherwise. Then by a union bound, with probability $\Omega(1)$ we never observe $S_t = r^2$ in the first r rounds, so $\mathbb{E}[N_A(r)] \in \Omega(r)$.

Proof of Lemma 3.8. We first consider the case that $r = s$. We claim that $N_A(s)$ is stochastically dominated by a geometric random variable $\mathrm{Geo}(p)$, where $p := \delta b/s$. Consider the first $r = s$ rounds. By condition (2), whenever we enter B, we spend all the remaining rounds in B with probability at least b/s. We pessimistically assume that we immediately return to A otherwise. Then for $X_t = A$, one of the following three cases will happen.

1. $X_{t+1} = A$, with probability at most $1 - \delta$.
2. $X_{t+1} = B$ and $X_{t+2} = A$, with probability at most $\delta(1 - b/s)$.
3. $X_{t+1} = X_{t+2} = \ldots, X_r = b$, with probability at least $\delta b/s$.

Hence, $N_A(s)$ is stochastically dominated by $\mathrm{Geo}(p)$ as claimed. In particular, $\mathbb{E}[N_A(s)] \le 1/p = s/(b\delta)$.

For the other case $r > s$, we split up the random walk into $k := \lceil r/s \rceil$ phases of length s each, which covers slightly more than r rounds. Then in each phase we know that the expected number of rounds in A is dominated by $\mathrm{Geo}(\delta b/s)$.

Regarding the expectation, the total number of rounds in A is at most $\mathbb{E}[N_A(r)] \leq k \cdot s/(b\delta) \leq 2r/(b\delta)$. For the tail bound, we need to bound the probability $q := \Pr[Y_1 + \ldots + Y_k > 4r/(b\delta)]$, where the Y_i are independent random variables with distribution Geo(p). We equivalently characterize q by $q = \Pr[\mathrm{Bin}(4r/(b\delta), p) < k]$. Since $k < 2r/s = \frac{1}{2}4rp/(b\delta)$, from the Chernoff bound, Theorem 3.1, we deduce $q \leq e^{-(1/2)^2(4r/s)/2} = e^{-r/(2s)}$. □

4. Results with bloat control

In this section we show the following theorem.

Theorem 4.1. *The* $(1+1)$ *GP* with bloat control *choosing* $k = 1 + \mathrm{Pois}(1)$ *on* Order *and* Majority *takes* $O(T_{\mathrm{init}} + n \log n)$ *iterations in expectation for any initial tree of size* T_{init}*, and there are initial trees of size* T_{init} *for which it takes* $\Omega(T_{\mathrm{init}} + n \log n)$ *iterations in expectation.*

4.1. Lower bound

Regarding the proof of the lower bound, let T_{init} and n be given. Let t be a GP-tree which contains T_{init} leaves labeled $\overline{x}_1$. From a simple coupon collector's argument we get a lower bound of $\Omega(n \log n)$ for the run time to insert each x_i. As an optimal tree cannot list any of the leaves in t in addition to the expected number of deletions performed by $(1+1)$ GP in one round being in $O(1)$, we obtain a lower bound of $\Omega(T_{\mathrm{init}})$ from the additive drift theorem (Theorem 3.2).

4.2. Upper bound

This section is dedicated to the proof of the upper bound. Let t be a GP-tree over n variables and denote the number of expressed variables of t by $v(t)$. We call the number of leaves of t the *size* of t and denote it by $s(t)$. For a best-so-far GP-tree of the $(1+1)$ GP we denote the size of the initial GP-tree by T_{init}. Both parameters n and T_{init} are considered to be given. The main difference to the case of only one mutation per iteration of the $(1+1)$ GP is that with more mutations in a single iteration the number of expressed variables can increase together with the introduction of a number of redundant leaves. The increased fitness will hinder the bloat control from rejecting the offspring even though the size could have increased by a large amount.

In order to deal with this behavior we are going to partition the set of leaves by observing the change of fitness when deleting one leaf. For a redundant leaf, the fitness is not affected by deleting it. However, not every non-redundant leaf contributes an expressed variable, since the deletion of a leaf can also increase the fitness if it is a negative literal. Thus, we consider the following sets of leaves.

$R(t)$: *Redundant leaves* v, where the fitness of t is not affected by deleting v.
$C^+(t)$: *Critical positive leaves* v, where the fitness of t decreases by deleting v.
$C^-(t)$: *Critical negative leaves* v, where the fitness of t increases by deleting v.

We denote by $r(t)$, $c^+(t)$ and $c^-(t)$ the cardinality of $R(t)$, $C^+(t)$ and $C^-(t)$, respectively. Thus we obtain

$$s(t) = r(t) + c^+(t) + c^-(t). \tag{9}$$

The general idea of the proof is the following: We are going to construct a suitable potential function g mapping a GP-tree t to a natural number in such a way that the optimum receives a value of 0 and the function displays the fitness with respect to the number of expressed variables and the size in a proper way. For a best-so-far GP-tree t let t' be the offspring of t under the $(1+1)$ GP. By bounding the drift, i.e. the expected change $g(t) - g(t')$ denoted by $\Delta(t)$, we are going to obtain the bound for the optimization time due to Theorem 3.3.

Regarding the bound on the drift we already argued that the case of only one mutation in an iteration is beneficial, since either the amount of expressed variables of parent and offspring are the same or the offspring has exactly one more variable expressed. However, the case of at least two mutations in an iteration is problematic in the above mentioned sense. In order to deal with the negative drift (leading away from the optimum) introduced by the latter case, the positive drift due to the other case has to outweigh the negative drift. Therefore, we need to bound the drift in both cases carefully.

We observe that starting with a very big initial tree the algorithm will delete redundant leaves with a constant probability until most of the occurring variables are expressed. In this second stage the size of the tree is at most linear in n and the algorithm will insert literals, which do not occur in the tree at all, with a probability of at least linear in $1/n$ until all variables are expressed. In order to obtain a better bound on the drift, we will split the second stage in two cases. Finally, by the law of total expectation we will obtain a bound on the drift due to the bounds under the mentioned cases.

In order to deal with critical leaves, we are going to prove upper bounds on the number of these. In fact, there exists a strong correlation between critical and redundant leaves we are going to exploit frequently.

Lemma 4.2. *Let t be a GP-tree, then for* Order *and* Majority *we have*

(i) $c^+(t) \leq r(t) + v(t)$,
(ii) $c^-(t) \leq 2r(t)$.

Proof. We prove both statements by observing the behavior of Order and Majority individually.

(i): Let $opt(t)$ be the number of optimal leaves, i.e. positive leaves x_i, where no additional instances of the variable i are present in t. Obviously $opt(t) \leq v(t) \leq n$ holds. We observe

$$c^+(t) - v(t) \leq c^+(t) - opt(t),$$

thus it suffices to bound the number of non-optimal critical positive leaves.
For Majority a variable i can only contribute such a leaf, if the number of positive literals x_i equals the number of negative literals $\overline{x}_i$. Since every such negative literal is a redundant leaf, we obtain $c^+(t) - opt(t) \leq r(t)$.
For Order a variable i can only contribute such a leaf, if the first occurrence of i is a positive literal x_i and the second occurrence is a negative literal $\overline{x}_i$. In this case the negative literal as well as every additional occurrence of a literal x_i is a redundant leaf. Therefore, we deduce $c^+(t) - opt(t) \leq r(t)$.

(ii): For Majority a variable i can only contribute a critical negative leaf if the number of positive literals x_i is m and the number of negative literals $\overline{x}_i$ is $m+1$ for some $m \geq 1$. In this case each negative literal is a critical negative leaf and each positive literal is a redundant leaf. We obtain $c^-(t) \leq 2r(t)$.
For Order a variable i can only contribute a critical negative leaf if the first occurrence of i is a negative literal and the second occurrence is a positive literal. In this case the first occurrence is a critical negative leaf and every additional occurrence afterwards is a redundant leaf. We obtain $c^-(t) \leq r(t)$. □

In order to construct the mentioned potential function, we want to reward strongly an increase of fitness given by a decrease of the unexpressed variables. Furthermore, we want to reward a decrease of size but without punishing an increase of fitness. Here, we need to be careful with the weights for both changes since a strong reward for a decrease of size might result in a very big negative drift in case of at least two operations. In order to illustrate the choice for the weights, we will fix the weight $m \in \mathbb{R}_{>0}$ for a decrease of unexpressed variables only later on. Thus, we associate with t the potential function

$$g(t) = m(n - v(t)) + s(t) - v(t).$$

This potential is 0 if and only if t contains no redundant leaves and for each $i \leq n$ there is an expressed x_i. Furthermore, by Lemma 4.2 $s(t) - v(t)$ is also 0 since $r(t)$ is 0.

Let $\mathcal{D}_1$ be the event where the algorithm chooses to do exactly one operation in the observed mutation step, and $\mathcal{D}_2$ where the algorithm chooses to do at least two operations in the observed mutation step. Since the algorithm chooses in each step at least one operation, we observe

$$\Pr[\mathcal{D}_1] = \Pr[\mathrm{Pois}(1) = 0] = \frac{1}{e},$$
$$\Pr[\mathcal{D}_2] = 1 - \frac{1}{e}.$$

Now we are going to derive bounds on the negative drift in the case $\mathcal{D}_2$. These are going to be connected with bounds on the positive drift for $\mathcal{D}_1$ by the law of total expectation. Let $\mathcal{E}$ be the event that $v(t') = v(t)$. As argued above, in the case $\mathcal{E}$ the potential cannot increase even if $\mathcal{D}_2$ holds. However, conditional on $\overline{\mathcal{E}}$ the potential can increase yielding a negative drift.

Lemma 4.3. *For the expected negative drift measured by $g(t)$ conditional on $\mathcal{D}_2$ holds*

$$\mathbb{E}[\Delta(t) \mid \mathcal{D}_2] \geq -\frac{1}{e}\left(2e - me + \sum_{i=1}^{m} \frac{m-i}{(i-1)!}\right).$$

In addition, if $s(t) > n/2$ holds, this bound is enhanced to

$$\mathbb{E}[\Delta(t) \mid \mathcal{D}_2] > -\frac{g(t)}{en}\left(\frac{1}{6m} + \frac{2}{3}\right)\left(2e - 5me + \sum_{i=1}^{m} \frac{i(m-i)}{(i-1)!}\right).$$

Proof. Concerning the drift conditional on $\mathcal{D}_2$ we observe

$$\mathbb{E}[\Delta(t) \mid \mathcal{D}_2] \geq -\mathbb{E}[-\Delta(t) \mid \overline{\mathcal{E}}] \Pr[\overline{\mathcal{E}}], \tag{10}$$

since the drift can be negative only in this case. In particular, we observe a drift of at least m for the increase of fitness counteracted by the possible increase of the size. The latter is at most the number of operations the algorithm does in the observed step, because every operation can increase the size by at most 1.

Let $Y \sim \text{Pois}(1) + 1$ be the random variable describing the number of operations in a round. Note that, for all $i \geq 1$,

$$\Pr[Y = i] = \frac{1}{e(i-1)!}.$$

By this probability we obtain for the expected negative drift conditional on $\overline{\mathcal{E}}$

$$\begin{aligned}\mathbb{E}[-\Delta(t) \mid \overline{\mathcal{E}}] &= \sum_{i=0}^{\infty} \mathbb{E}[-\Delta(t) \mid Y = i, \overline{\mathcal{E}}] \Pr[Y = i \mid \overline{\mathcal{E}}] \leq \sum_{i=0}^{\infty} (i-m) \Pr[Y = i \mid \overline{\mathcal{E}}] \\ &\leq \sum_{i=m+1}^{\infty} (i-m) \Pr[Y = i \mid \overline{\mathcal{E}}].\end{aligned}$$

Due to Bayes' theorem we derive

$$\mathbb{E}[-\Delta(t) \mid \overline{\mathcal{E}}] \leq \sum_{i=m+1}^{\infty} (i-m) \Pr[\overline{\mathcal{E}} \mid Y = i] \frac{\Pr[Y = i]}{\Pr[\overline{\mathcal{E}}]},$$

which yields the first bound due to inequality (10) by pessimistically assuming $\Pr[\overline{\mathcal{E}} \mid Y = i] = 1$

$$\mathbb{E}[\Delta(t) \mid \mathcal{D}_2] \geq -\sum_{i=m+1}^{\infty} (i-m) \Pr[Y = i] = -\frac{1}{e}\left(2e - me + \sum_{i=1}^{m} \frac{m-i}{(i-1)!}\right).$$

In order to obtain a better bound on the negative drift, we are going to bound the probability $\Pr[\overline{\mathcal{E}} \mid Y = i]$ by a better bound than the previously applied bound of 1.

The event $\overline{\mathcal{E}}$ requires a non-expressed variable in t to become expressed in t'. There are $n - v(t)$ non-expressed variables in t. These can become expressed by either adding a corresponding positive literal or deleting a corresponding negative literal. There are $2n$ literals in total and due to $n - v(t) \leq g(t)/m$ adding such a positive literal has a probability of at most

$$\frac{n - v(t)}{6n} \leq \frac{g(t)}{6mn}$$

per operation. Regarding the deletion of negative literals, there are at most $s(t) - v(t)$ negative literals. Hence, due to $s(t) - v(t) \leq g(t)$ and $s(t) > n/2$ the probability of deleting a negative literal is at most

$$\frac{s(t) - v(t)}{3s(t)} \leq \frac{2g(t)}{3n}$$

per operation. Let q_l be the probability that the l-th mutation leads an unexpressed variable to become expressed. We can bound the probability that i operations lead to the expression of a previously unexpressed bound by pessimistically assuming that the mutation is going to be accepted. This yields by the union bound

$$\Pr[\overline{\mathcal{E}} \mid Y = i] \leq \bigcup_{l=1}^{i} q_l \leq \sum_{l=1}^{i} q_i = \frac{ig(t)}{n}\left(\frac{1}{6m} + \frac{2}{3}\right).$$

Therefore, we obtain due to inequality (10) an expected drift conditional on $\mathcal{D}_2$ of

$$\begin{aligned}\mathbb{E}[\Delta(t) \mid \mathcal{D}_2] &> -\frac{g(t)}{en}\left(\frac{1}{6m} + \frac{2}{3}\right) \sum_{i=m+1}^{\infty} \frac{i(i-m)}{(i-1)!} \\ &= -\frac{g(t)}{en}\left(\frac{1}{6m} + \frac{2}{3}\right)\left(2e - 5me + \sum_{i=1}^{m} \frac{i(m-i)}{(i-1)!}\right). \quad \square\end{aligned}$$

As a small spoiler for the choice of m, we will give the following Corollary on Lemma 4.2.

Corollary 4.4. *For $m = 10$ we obtain the following bounds*

$$\mathbb{E}[\Delta(t) \mid \mathcal{D}_2] \geq -\frac{1}{e}\left(4 \cdot 10^{-7}\right).$$

In addition, if $s(t) > n/2$ holds, this bound is enhanced to

$$\mathbb{E}[\Delta(t) \mid \mathcal{D}_2] > -\frac{7g(t)}{10en}\left(4 \cdot 10^{-6}\right).$$

We are now going to prove the upper bound by deriving the expected positive drift outweighing the negative drift given by Lemma 4.3.

Case 1: We first consider the case $r(t) \geq v(t)$. Due to Lemma 4.2 and Equation (9) we obtain

$$s(t) = r(t) + c^+(t) + c^-(t) \leq 4r(t) + v(t) \leq 5r(t),$$

thus the algorithm has a probability of at least $1/5$ for choosing a redundant leaf followed by choosing a deletion with probability $1/3$. Since the deletion of a redundant leaf without any additional operations does not change the fitness this contributes to the event $\mathcal{E}$. Hence, we obtain for the event $\mathcal{D}_1$

$$\mathbb{E}[\Delta(t) \mid \mathcal{D}_1,\, \mathcal{E}] \Pr[\mathcal{E}] \geq \frac{1}{15}.$$

Additionally, the drift conditional on $\mathcal{D}_1$ is always positive, which yields

$$\mathbb{E}[\Delta(t) \mid \mathcal{D}_1] \geq \mathbb{E}[\Delta(t) \mid \mathcal{D}_1,\, \mathcal{E}] \Pr[\mathcal{E}] \geq \frac{1}{15}.$$

The drift conditional on $\mathcal{D}_2$ is given by Lemma 4.3. We observe, that the positive drift of $1/15$ outweighs the negative drift for the choice of $m = 10$ given by Corollary 4.4. Overall, we obtain a constant drift in the case of $r(t) \geq v(t)$ due to the law of total expectation

$$\begin{aligned}\mathbb{E}[\Delta(t)] &\geq \mathbb{E}[\Delta(t) \mid \mathcal{D}_1] \Pr[\mathcal{D}_1] + \mathbb{E}[\Delta(t) \mid \mathcal{D}_2] \Pr[\mathcal{D}_2] \geq \frac{1}{15e} - \frac{1}{e}\left(1 - \frac{1}{e}\right)\left(4 \cdot 10^{-7}\right) \\ &\geq \frac{1}{e}\left(\frac{1}{15} - 4 \cdot 10^{-7}\right) \geq \frac{3}{50e}. \end{aligned} \tag{11}$$

Case 2: Suppose $r(t) < v(t)$ and $s(t) \leq n/2$. In particular, we have for at least $n/2$ many $i \leq n$ that there is neither x_i nor $\overline{x}_i$ present in t. The probability to choose such an x_i is at least $1/4$ and the probability that the algorithm chooses an insertion is $1/3$. This insertion will yield a fitness increase of m and since the location of the newly inserted literal is unimportant we obtain

$$\mathbb{E}[\Delta(t) \mid \mathcal{D}_1] \Pr[\mathcal{D}_1] \geq \frac{m}{12e}.$$

For the expected drift in the case $\mathcal{D}_2$ holds we apply again the bound given by Lemma 4.3. Analogue to Case 1 we observe, that the positive drift outweighs the negative drift for the choice of $m = 10$, which yields the following constant drift

$$\mathbb{E}[\Delta(t)] \geq \frac{1}{e}\left(\frac{10}{12} - 4 \cdot 10^{-7}\right) > \frac{8}{10e}.$$

Case 3: Consider now the case that $r(t) < v(t)$ and $s(t) > n/2$. In particular, the tree can contain at most $5n$ leaves due to

$$s(t) \leq 4r(t) + v(t) < 5v(t) \leq 5n,$$

which enables us to bound the probability that an operation chooses a specific leaf v as

$$\frac{1}{5n} \leq \Pr[\text{choose leaf } v] \leq \frac{2}{n}.$$

Let A be the set of i, such that there is neither x_i nor $\overline{x}_i$ in t, and let B be the set of i, such that there is exactly one x_i and no $\overline{x}_i$ in t. Recall that $R(t)$ is the set of redundant leaves in t. For every i in A let $\mathcal{A}_i$ be the event that the algorithm adds x_i somewhere in t. For every j in $R(t)$ let $\mathcal{R}_j(t)$ be the event, that the algorithm deletes j. Finally, let $\mathcal{A}'$ be the event that one of the A_i holds, and $\mathcal{R}'$ the event that one of the $\mathcal{R}_j(t)$ holds.

Conditional on $\mathcal{D}_1$ we observe for every event $\mathcal{A}_i$ a drift of m. For each event $\mathcal{R}_j(t)$ conditional on $\mathcal{D}_1$ we observe a drift of 1 since the amount of redundant leaves decreases by exactly 1. Hence,

$$\mathbb{E}[\Delta(t) \mid \mathcal{A}_i, \mathcal{D}_1] = m,$$
$$\mathbb{E}[\Delta(t) \mid \mathcal{R}_j(t), \mathcal{D}_1] = 1.$$

Regarding the probability for these events we observe that for $\mathcal{A}_i$ the algorithm chooses with probability $1/3$ to add a leaf and with probability $1/(2n)$ it chooses x_i for this. Furthermore, the position of the new leaf x_i is unimportant, hence

$$\Pr[\mathcal{A}_i \mid \mathcal{D}_1] \geq \frac{1}{6n}.$$

Regarding the probability of $\mathcal{R}_j(t)$, with probability at least $1/(5n)$ the algorithm chooses the leaf j and with probability $1/3$ the algorithm deletes j. This yields

$$\Pr[\mathcal{R}_j(t) \mid \mathcal{D}_1] \geq \frac{1}{15n}.$$

In order to sum the events in $\mathcal{A}'$ and $\mathcal{R}'$, we need to bound the cardinality of the two sets A and $R(t)$. For this purpose we will need the above defined set B. First we note that the cardinality of B is at most $v(t)$. In addition

$$|A| + |R(t)| \geq r(t) \tag{12}$$

holds since $R(t)$ is the set of all redundant leaves. Furthermore, we observe that for any variable j, which is not in B or A, there has to exist at least one redundant leaf x_j or $\overline{x}_j$. Since every redundant leaf is included in $R(t)$ we obtain $|A| + |R(t)| + |B| \geq n$ and subsequently

$$|A| + |R(t)| \geq n - v(t). \tag{13}$$

Furthermore, due to Lemma 4.2 we deduce

$$s(t) - v(t) \leq r(t) + c^+(t) + c^-(t) - v(t) \leq 4r(t) \leq 4(|A| + |R(t)|), \tag{14}$$

where the last inequality is due to (12). This inequality (14) in conjunction with (13) yields

$$(m+4)(|A| + |R(t)|) \geq m(n - v(t)) + s(t) - v(t) = g(t). \tag{15}$$

We obtain the expected drift conditional on the event $\mathcal{D}_1$ as for $m \geq 1$

$$\begin{aligned}\mathbb{E}[\Delta(t) \mid \mathcal{D}_1] &\geq \mathbb{E}[\Delta(t) \mid (\mathcal{A}' \vee \mathcal{R}'), \mathcal{D}_1] \Pr[\mathcal{A}' \vee \mathcal{R}' \mid \mathcal{D}_1] \\ &= \sum_{i \in A} \mathbb{E}[\Delta(t) \mid \mathcal{A}_i, \mathcal{D}_1] \Pr[\mathcal{A}_i, \mathcal{D}_1] + \sum_{j \in R(t)} \mathbb{E}[\Delta(t) \mid \mathcal{R}_j(t), \mathcal{D}_1] \Pr[\mathcal{R}_j(t) \mid \mathcal{D}_1] \\ &\geq |A| \frac{m}{6n} + |R(t)| \frac{1}{15n} \geq (|A| + |R(t)|) \frac{1}{15n} \geq \frac{g(t)}{15(m+4)n},\end{aligned}$$

where the last inequality is due to (15). Concerning the expected drift conditional on $\mathcal{D}_2$, the condition for the second bound given by Lemma 4.3 is satisfied in this case. Again, we observe that the positive drift outweighs the negative drift for $m = 10$ given by Corollary 4.4, which justifies the choice of $m = 10$ we are setting from here on. In fact, we could choose any integer $m \geq 5$ in order for the positive drift to outweigh the negative. Summarizing the events $\mathcal{D}_1$ and $\mathcal{D}_2$ we obtain the expected drift

$$\begin{aligned}\mathbb{E}[\Delta(t)] &\geq \mathbb{E}[\Delta(t) \mid \mathcal{D}_1] \Pr[\mathcal{D}_1] + \mathbb{E}[\Delta(t) \mid \mathcal{D}_2] \Pr[\mathcal{D}_2] \\ &\geq \frac{g(t)}{en} \left(\frac{1}{210} - \left(1 - \frac{1}{e}\right) \frac{7}{10} \cdot 4 \cdot 10^{-6} \right) > \frac{g(t)}{250en}.\end{aligned} \tag{16}$$

Summarizing the derived expected drifts (11) and (16), we observe a multiplicative drift in the case of

$$\frac{g(t)}{250en} \leq \frac{3}{50e},$$

which simplifies to $g(t) \leq 15n$. If $g(t) > 15n$, we observe a constant drift. This constant drift is at least $3/50e$ since the expected drift for Case 2 is always bigger than the one for Case 1.

We now apply the variable drift theorem (Theorem 3.3) with $h(x) = \min\{3/(50e), 1x/(250en)\}$, $X_0 = T_{\text{init}} + 10n$ and $X_{\min} = 1$, which yields

$$\mathbb{E}[T \mid g(t)=0] \le \frac{1}{h(1)} + \int_1^{T_{\text{init}}+10n} \frac{1}{h(x)}\,dx$$
$$= 250en + 250en \int_1^{15n} \frac{1}{x}\,dx + \frac{50e}{3} \int_{15n+1}^{T_{\text{init}}+10n} 1\,dx$$
$$= 250en\,(1+\log(15n)) + \frac{50e}{3}\,(T_{\text{init}} - 5n - 1) < 250en\log(15en) + \frac{50e}{3} T_{\text{init}}.$$

This establishes the theorem.

5. Results without bloat control

In this section we show the following theorems.

Theorem 5.1. *The* $(1+1)$ *GP* without bloat control *(choosing* $k=1$ *or* $k=1+\text{Pois}(1)$*) on MAJORITY takes* $\Omega(T_{\text{init}}\log T_{\text{init}})$ *iterations in expectation for* $n=1$ *for some initial trees. For general* $n \ge 1$ *and for some initial trees, it takes* $\Omega(T_{\text{init}} + n\log n)$ *iterations in expectation.*

Theorem 5.2. *The* $(1+1)$ *GP* without bloat control *(choosing* $k=1$ *or* $k=1+\text{Pois}(1)$*) on MAJORITY takes* $O(T_{\text{init}}\log T_{\text{init}} + n\log^3 n)$ *iterations in expectation.*

5.1. Proof of the lower bound

Regarding the proof of Theorem 5.1, let T_{init} be large. Let t_0 be a GP-tree which contains T_{init} leaves labeled $\overline{x}_1$ and no other leaves. From a simple coupon collector's argument we get a lower bound of $\Omega(n\log n)$ for the run time to insert each x_i. It remains to bound the time the algorithm needs to express the x_1.

In order to derive the bound for general $n \ge 1$ we observe, that the algorithm does in expectation 2 operations in each iteration since $\mathbb{E}[1+\text{Pois}(1)] = 2$. Hence, the algorithm needs in expectation at least $T_{\text{init}}/2$ iterations to express the first variable yielding the desired result.

Regarding the bound for the case $n=1$ let t be a GP-tree, let $I_1(t)$ be the set of literals x_1 in t and $I'_1(t)$ be the set of literals $\overline{x}_1$ in t. Additionally, we define $|I_1(t)| = i_1(t)$ and $|I'_1(t)| = i'_1(t)$. We associate with t the potential function $g(t)$ by

$$g(t) = i'_1(t) - i_1(t).$$

In order to express the variable 1, the potential $g(t)$ has to become non-positive at one point. In particular, starting with $g(t_0) = T_{\text{init}}$, the potential has to reach a value of at most $T_{\text{init}}^{2/3}$. Let τ denote the number of iterations until the algorithm encounters for the first time a GP-tree t with $g(t) \le T_{\text{init}}^{2/3}$. We are going to bound the expected value of τ starting with t_0, since this will yield a lower bound for the expected number of iterations until x_1 is expressed.

Let $\mathcal{A}_i$ be the event, that the algorithm performs more than $15\ln(T_{\text{init}})$ operations in the i-th iteration. For a better readability we define z to be $15\ln(T_{\text{init}})$. Regarding the probability of $\mathcal{A}_i$ we obtain due to the Poisson-distributed number of operations

$$\Pr[\mathcal{A}_i] = \sum_{i=z}^{\infty} \frac{1}{e(i-1)!}.$$

Let p_i be the probability, that a Pois(1) distributed random variable is equal to i. We derive $p_{i+1} = p_i/(i+1) \le p_i/2$. Since $\mathcal{A}_i$ is Pois(1)-distributed, this yields

$$\Pr[\mathcal{A}_i] \le p_z \sum_{i=0}^{\infty} \frac{1}{2^i} = \frac{2}{ez!}.$$

By the Stirling bound $n! \ge e(n/e)^n$ we obtain

$$\Pr[\mathcal{A}_i] \le \frac{e^z}{ez^z} \le \frac{T_{\text{init}}^{15}}{z^z} \le T_{\text{init}}^{-15},$$

where the last inequality comes from $z^z \ge e^{2z}$, which holds for $T_{\text{init}} \ge 2$.

Let $\mathcal{A}$ be the event that in T_{init}^2 iterations the algorithm performs at least once more than z operations in a single iteration. By the union bound we obtain for the probability of $\mathcal{A}$

$$\Pr[\mathcal{A}] = \Pr\left[\bigcup_{i=1}^{T_{\text{init}}^2} \mathcal{A}_i\right] \leq \sum_{i=1}^{T_{\text{init}}^2} \Pr[\mathcal{A}_i] \leq T_{\text{init}}^{-13}.$$

Hence, w.h.p. the algorithm will not encounter the event $\mathcal{A}$. By the law of total expectation we deduce

$$\mathbb{E}[\tau] = \mathbb{E}[\tau \mid \mathcal{A}] \Pr[\mathcal{A}] + \mathbb{E}[\tau \mid \overline{\mathcal{A}}] \Pr[\overline{\mathcal{A}}] \geq \mathbb{E}[\tau \mid \overline{\mathcal{A}}] \frac{1}{2}. \tag{17}$$

It remains to bound the expected value of τ under the constraint of $\overline{\mathcal{A}}$.

Let t' be the random variable describing the best-so-far solution in the iteration after t. We are going to derive an upper bound on the drift, i.e. the expected change $g(t) - g(t')$ denoted by $\Delta(t)$, in order to apply the Multiplicative Drift Theorem (Theorem 3.7). The event $\overline{\mathcal{A}}$ assures the condition of a bounded step size. We recall that $g(t) = i_1'(t) - i_1(t)$, where $i_1'(t)$ is the number of literals $\overline{x}_1$ and $i_1(t)$ is the number of literals x_1. If the algorithm chooses an insertion, the probability to insert x_1 is the same as the probability to insert $\overline{x}_1$. Therefore, an insertion will only contribute 0 to the expected drift. The same holds for the literals *introduced* by a substitution. However, for literals *deleted* by a deletion or substitution the probability to choose a literal x_1 or $\overline{x}_1$ is of importance, contrary to the case of insertion.

In order to analyze the drift $\Delta(t)$, we observe that a deletion of a literal in $I_1'(t)$ yields a positive drift of 1, whereas a deletion of a literal in $I_1(t)$ yields a negative drift of -1. For all $j \in [s(t)]$ we define for the j-th literal x in t the indicator variable J_j by

$$J_j = \begin{cases} 0, & \text{if } x \text{ does not get deleted or substituted;} \\ 1, & \text{if } x \text{ is a literal } \overline{x}_1\text{, which gets deleted or substituted;} \\ -1, & \text{if } x \text{ is a literal } x_1\text{, which gets deleted or substituted.} \end{cases}$$

Let $\mathcal{J}_j$ be the event that $J_j \neq 0$. Utilizing these events and the linearity of expectation we deduce

$$\mathbb{E}[\Delta(t)] = \mathbb{E}\left[\sum_{j=1}^{s(t)} \mathcal{J}_j\right] = \sum_{j=1}^{s(t)} \mathbb{E}[\mathcal{J}_j] = \sum_{j \in I_1'(t)} \mathbb{E}[\mathcal{J}_j] - \sum_{j \in I_1(t)} \mathbb{E}[\mathcal{J}_j]. \tag{18}$$

Regarding the probability of $\mathcal{J}_j$ we observe that with a probability of $2/(3s(t))$ the j-th literal will be chosen for a deletion or substitution in one operation. Therefore, with probability $(1 - 2/(3s(t)))^i$ the j-th literal will not be chosen for a deletion or substitution i-times. We deduce, using that k is the random number of operations in a given round, $\Pr[\mathcal{J}_j \mid k = i] = 1 - (1 - 2/3s(t))^i$ and hence

$$\Pr[\mathcal{J}_j] = \sum_{i=1}^{\infty} \Pr[\mathcal{J}_j \mid k = i]\, p_{i-1} = \sum_{i=1}^{\infty} \left(1 - \left(1 - \frac{2}{3s(t)}\right)^i\right) \frac{1}{e(i-1)!}.$$

We observe that $\Pr[\mathcal{J}_j] = \Pr[\mathcal{J}_i]$ for all $i, j \leq s(t)$. The probability for $\mathcal{J}_j$ is at least the probability to have only one operation, which already deletes or substitutes the j-th literal and thus $\Pr[\mathcal{J}_j] \geq 2/(3es(t))$. Regarding an upper bound on the probability we apply a Bernoulli-bound yielding $(1 - 2/(3s(t)))^i \geq 1 - 2i/(3s(t))$ and

$$\Pr[\mathcal{J}_j] \leq \sum_{i=1}^{\infty} \frac{2i}{3s(t)e(i-1)!} = \frac{4}{3s(t)}.$$

Combining both bounds we deduce that $\Pr[\mathcal{J}_j] \in \Theta(1/s(t))$. Hence, there exists a constant c such that we obtain due to (18) and $\Pr[\mathcal{J}_j] = \Pr[\mathcal{J}_i]$ for all $i, j \leq s(t)$

$$\mathbb{E}[\Delta(t)] = i_1'(t) \Pr[\mathcal{J}_j] - i_1(t) \Pr[\mathcal{J}_j] \leq \frac{c}{s(t)} \left(i_1'(t) - i_1(t)\right) = \frac{c}{s(t)}\, g(t).$$

In order to bound the size $s(t)$ we observe that following a standard gambler's ruin argument within $o(T_{\text{init}}^{1.5})$ iterations the size will not shrink by a factor bigger than $1/2$. Therefore, we obtain $s(t) \geq 1/2\ T_{\text{init}}$. Due to the step size bound of $15\ln(T_{\text{init}}) < T_{\text{init}}^{2/3}$ we can apply Theorem 3.7 with $s_{min} = T_{\text{init}}^{2/3}$ and $\delta = 2c/T_{\text{init}}$ and derive

$$\mathbb{E}[\tau \mid \overline{\mathcal{A}},\ X_0 = T_{\text{init}}] \geq \frac{1 + \ln(T_{\text{init}}) - \ln(T_{\text{init}}^{2/3})}{\frac{4c}{T_{\text{init}}} + \frac{(15\ln(T_{\text{init}}))^2}{T_{\text{init}}^{4/3} - (15\ln(T_{\text{init}}))^2}}.$$

In order to simplify this bound we observe $\ln(T_{\text{init}}) \leq 3T_{\text{init}}^{1/3}$, which yields

$$\frac{(15\ln(T_{\text{init}}))^2}{T_{\text{init}}^{4/3} - (15\ln(T_{\text{init}}))^2} \leq \frac{(15\ln(T_{\text{init}}))^2}{T_{\text{init}}^{4/3} - (45T_{\text{init}}^{1/3})^2} \leq \frac{1}{2T_{\text{init}}}.$$

Therefore, we obtain due to (17)

$$\mathbb{E}[\tau] \geq \frac{T_{\text{init}}\ln(T_{\text{init}})}{3(8c+1)}$$

establishing the theorem.

5.2. Proof of the upper bound

5.2.1. Outline

Since the proof of Theorem 5.2 is long and involved, we first give an outline of the proof. The key ingredient is a bound on the bloat, i.e., on the speed with which the tree grows. Roughly speaking, we will show in Theorem 5.4 that if $T_{\text{init}} \geq n\log^2 n$, then the size of the tree grows at most by a constant factor in $O(T_{\text{init}}\log T_{\text{init}})$ rounds.

Before we elaborate on the bloat, let us first sketch how this implies the upper bound. Consider any x_i that is not expressed and let $V'(t_r, i) := \#\{\overline{x}_i\text{-literals}\} - \#\{x_i\text{-literals}\} \geq 1$. For this outline we neglect the case that there are neither $\overline{x}_i$ nor x_i in the string. Then the probability of deleting or substituting an $\overline{x}_i$ is larger than deleting or substituting an x_i, while they have the same probability to be inserted. Computing precisely, and denoting by t_r the GP-tree in round r, we will show an expected drift of

$$\mathbb{E}[V'(t_r, i) - V'(t_{r+1}, i) \mid V(t_r, i) = v] \geq \frac{v}{3eT_{\max}} \tag{19}$$

for the $V'(t_r, i)$, where $T_{\max} \in O(T_{\text{init}})$ is an upper bound on the size of the tree. Using a multiplicative drift theorem, Theorem 3.4, after $O(T_{\text{init}}\log T_{\text{init}})$ rounds we have $V'(t_r, i) = 0$ with very high probability. By a union bound over all i, w.h.p. there is no i left after $O(T_{\text{init}}\log T_{\text{init}})$ rounds for which $V'(t_r, i) < 0$. This proves the theorem modulo the statement on the bloat.

Regarding the bloat, we note that in expectation the offspring has the same size as the parent and the size of the tree does not change significantly by such unbiased fluctuations. However, in some situations bigger offspring are more likely to be accepted or shorter offspring are more likely to be rejected. For example, if the tree contains only positive critical literals then every deletion will be rejected. This results in a positive drift for the size, which we need to bound. Note that since the offspring has in expectation the same size as the parent, the biased drift is caused purely by the selection process, i.e., the drift would be zero if all offspring were accepted. We will show that offspring are rarely rejected and bound the drift of $s(t_r)$ by (essentially) the probability that the offspring is rejected.

Similar to before, for an expressed variable x_i we let $V(t_r, i) := \#\{x_i\text{-literals}\} - \#\{\overline{x}_i\text{-literals}\} \geq 0$. An important insight is that the offspring can only be rejected if there is some expressed variable i such that at least $V(t_r, i) + 1$ mutations touch i, i.e., they insert (by insertion or substitution) or delete (by deletion or substitution) x_i-literals or $\overline{x}_i$-literals.[2] We want to show that this does not happen frequently. The probability to touch x_i-literals or $\overline{x}_i$-literals at least k times falls geometrically in k, as we show in Lemma 5.3. So for this outline we will restrict to the most dominant case $V(t_r, i) = 0$.

Assume that we are in a situation where the size of the tree has grown at most by a constant factor. Similar to before, we may bound the drift of $V(t_r, i)$ in rounds that touch i by

$$\mathbb{E}[V(t_r, i) - V(t_{r+1}, i) \mid V(t_r, i) = v, i \text{ touched in round } r] \leq \frac{Cvn}{T_{\text{init}}} \tag{20}$$

for a suitable constant $C > 0$. The factor n appears because we condition on i being touched in round r, which happens with probability $\Omega(1/n)$.

Equation (20) tells us that the drift towards zero may be positive, but that it is relatively weak. In particular, for $v \leq N := \sqrt{T_{\text{init}}/n}$, the drift is at most $O(1/N)$. Under such circumstances the expected return time to 0 is large. More precisely, it follows from martingale theory (Theorem 3.6) that the expected number of rounds that touch i to reach $V(t_r, i) = 0$ from any starting configuration is at least $\Omega(N)$.[3] In particular, after $V(t_r, i)$ becomes positive for the first time, it needs in expectation $\Omega(N)$ rounds that touch i to return to 0. On the other hand, it only needs $O(1)$ rounds that touch i to leave 0 again. Hence, $V(t_r, i)$ only has value 0 in an expected $O(1/N)$-fraction of all rounds that touch i.[4] However, recall that the drift of $s(t_r)$ is driven by the selection operator, since the drift of $s(t_r)$ would be zero if every offspring would be accepted. If $V(t_r, i)$ is positive then x_i does not influence whether the offspring is accepted. Formally, whenever all variables that are

[2] Some borders cases are neglected in this statement.

[3] Interestingly, we also show that a substantial part of this expectation comes from return times of size $\Omega(N^2)$, which will be important to obtain tail bounds.

[4] This statement is more subtle than it may seem, and it is only true because the return times have a suitable tail distribution.

touched in a round have positive $V(t_r, i)$, then the drift of $s(t_r)$ is zero in this round. Since we have argued that $V(t_r, i)$ is positive in all but a $O(1/N)$ fraction of the rounds, the drift of $s(t_r)$ is also $O(1/N)$.

In particular, if $T_{\text{init}} \geq n \log^2 n$ then in $r_0 \in O(T_{\text{init}} \log T_{\text{init}})$ rounds the drift increases the size of the GP-tree in expectation by at most $r_0/N \in O(T_{\text{init}})$. Hence, we expect the size to grow by at most a constant factor. In fact, we provide strong tail bounds showing that it is rather unlikely to grow by more than a constant factor. The exact statement can be found in Theorem 5.4.

5.2.2. Preparations

We now turn to the formal proof of Theorem 5.2.

Notation. We start with some notation and technical lemmas. For a variable $i \in [n]$, we say that i is *touched* by some mutation, if the mutation inserts, delete or substitutes an x_i- or $\overline{x}_i$-literal, or if it substitutes a literal by x_i or $\overline{x}_i$. We say that a mutation touches i *twice* if it substitutes an x_i-literal into $\overline{x}_i$ or vice versa. Note that a substitution has only probability $O(1/n)$ to touch a literal twice. We call a round an *i-round* if at least one of the mutations in this round touches i. Finally, we say that i is *touched s times* in a round if it is touched exactly s times by the mutations of this round (counted with multiplicity 2 for mutations that touch i twice).

For a GP-tree t, let

$$V(t,i) := \begin{cases} -1, & \text{no } x_i \text{ or } \overline{x}_i \text{ appear in the tree;} \\ -z, & \text{there are } z > 0 \text{ more } \overline{x}_i \text{ than } x_i; \\ z, & i \text{ is expressed, and there are } z \geq 0 \text{ more } x_i \text{ than } \overline{x}_i. \end{cases}$$

In particular, i is expressed if and only if $V(t,i) \geq 0$. Note that $V(t,i) = -1$ may occur either if x_i and $\overline{x}_i$ do not appear at all, or if exactly one more $\overline{x}_i$ than x_i appears. Both cases have in common that i will be expressed after a single insertion of x_i.

Note that a mutation that touches i once can change $V(t,i)$ by at most 1, with one exception: if $V(t,i) = 1$ and there is only a single positive x_i-literal, then $V(t,i)$ may drop to -1 by deleting this literal. Conversely, $V(t,i)$ can jump from -1 to 1 by the inverse operation. In general, if i is touched at most s times and $V(t,i) > s$ then $V(t,i)$ can change at most by s; it can change sign only if $|V(t,i)| \leq s$. We say that a variable i is *critical* in a round if $V(t,i) \geq 0$, and i is touched at least $V(t,i)$ times in this round; we call the variable *non-critical* otherwise. Moreover, we say that a variable is *positive critical* if it is critical and $V(t,i)$ is strictly positive. We say that a round is (positive) critical if there is at least one (positive) critical variable in this round. Note that in a non-critical round, the fitness of the offspring cannot be smaller than the fitness of the parent. Hence, in these rounds every offspring is accepted, and thus the change of size is unbiased in these rounds. Hence the biased drift of $s(t_r)$ comes only from critical rounds.

Many Mutations. We conclude our preparations with a lemma stating that it is exponentially unlikely to have many mutations, even if we condition on some variable to be touched.

Lemma 5.3. *There are constants $C, \delta > 0$ and $n_0 \in \mathbb{N}$ such that the following is true for every $n \geq n_0$, every GP-tree t with $T \geq 2n$ leaves , and every $\kappa \geq 2$. Let $i \in [n]$, and let k denote the number of mutations in the next round. Then:*

1. $\Pr[k \geq \kappa] \leq e^{-\delta\kappa}$.
2. $\Pr[k = 1 \mid i \text{ touched}] \geq \delta$.
3. $\Pr[k \geq \kappa \mid i \text{ touched}] \leq e^{-\delta\kappa}$.
4. $\mathbb{E}[k \mid i \text{ touched}] \leq C$.

Proof. Note that all statements are trivial if the $(1+1)$ GP uses $k = 1$ deterministically. So for the rest of the proof we will assume that k is $1 + \text{Pois}(1)$-distributed. We will use the well known inequality

$$\Pr[\text{Pois}(\lambda) \geq x] \leq e^{-\lambda} \left(\frac{e\lambda}{x}\right)^x \tag{21}$$

for the Poisson distribution [21]. In our case ($\lambda = 1$, $x = \kappa - 1$), and using $e^{-1} \leq 1$, we can simplify to

$$\Pr[\text{Pois}(1) \geq \kappa - 1] \leq \left(\frac{e}{\kappa - 1}\right)^{\kappa - 1}. \tag{22}$$

1: First consider $\kappa \geq 4$. Then, using $\kappa - 1 \geq \kappa/2$ we get from (22):

$$\Pr[k \geq \kappa] = \Pr[\text{Pois}(1) \geq \kappa - 1] \leq (e/3)^{\kappa/2} = e^{\log(e/3)\kappa/2}.$$

Thus 1 is satisfied for $\kappa \geq 4$ with $\delta := \log(e/3)/2$. By making δ smaller if necessary, we can ensure that 1 is also satisfied for $\kappa \in \{2, 3\}$, which proves this property.

2 and 3: Let $T = s(t)$ be the size of t (the number of leaves). Additionally, we define the parameter

$$x := \max\left\{\frac{\#\{\text{i-literals in } t\}}{T}, \frac{1}{n}\right\}.$$

Note that the next mutation has probability at most $2x$ to touch i. Unfortunately, that is not true for subsequent mutations in the same round, which makes the proof considerably more complicated. We claim

$$\Pr[k = 1 \text{ and } i \text{ touched}] \geq \frac{x}{3e}. \tag{23}$$

To see the claim, first note that $\Pr[k = 1] = 1/e$ by definition of the Poisson distribution. First, consider the case that $x = 1/n$. Then we have $\Pr[k = 1 \text{ and } x_i \text{ or } \overline{x}_i \text{ inserted}] = 1/(3en)$, which implies (23). In the other case, the probability that a deletion operation picks a x_i or $\overline{x}_i$ is x, so $\Pr[k = 1 \text{ and } x_i \text{ or } \overline{x}_i \text{ inserted}] = x/(3e)$, which also implies (23). This proves (23) in all cases.

We first prove the simpler case of large x; more precisely, let $x \geq 1/4$. With probability $1/e$ there is only one mutation and with probability at least $x/3 \geq 1/12$ this mutation deletes a x_i or $\overline{x}_i$-literal. Hence,

$$\Pr[k = 1 \text{ and } i \text{ touched}] \geq \frac{1}{12e}.$$

This already implies 2, because

$$\Pr[k = 1 \mid i \text{ touched}] \geq \Pr[k = 1 \text{ and } i \text{ touched}] \geq \frac{1}{12e}.$$

Regarding 3 it suffices to observe that

$$\begin{aligned}\Pr[k \geq \kappa \mid i \text{ touched}] &= \frac{\Pr[k \geq \kappa \text{ and } i \text{ touched}]}{\Pr[i \text{ touched}]} \\ &\leq \frac{\Pr[k \geq \kappa]}{\Pr[k = 1 \text{ and } i \text{ touched}]} \overset{1.}{\leq} 12e \cdot e^{-\delta\kappa},\end{aligned} \tag{24}$$

which implies 3 by absorbing the factor $12e$ into the exponential. This settles the case $x \geq 1/4$.

The case for smaller x basically runs along the same lines, but will be much more involved. In particular, in (24) we cannot use the trivial bounds in the second line. So assume from now on $x < 1/4$ and thus at most one fourth of the literals in t are i-literals. In the following we will bound the probability to have $k > 1$ mutations such that at least one of them touches i. The probability to have $k = \kappa$ mutations is $\Pr[\text{Pois}(1) = \kappa - 1]$. We will first assume $k \leq 1/x$. Note for later reference that $k \leq 1/x \leq n \leq T/2$ in this situation.

So fix some value $k \leq 1/x$. Let us refer to the mutations by $M_1, \ldots, M_k$ and let $\kappa_i := \min\{1 \leq \kappa \leq k \mid M_\kappa \text{ touches } i\}$ be the index of the first mutation that touches i. If none of $M_1, \ldots, M_k$ touches i then we set $\kappa_i := \infty$. We claim that for all $k \leq 1/x$ and all $1 \leq \kappa \leq k$,

$$\Pr[\kappa_i \geq \kappa + 1 \mid k, \kappa_i \geq \kappa] \geq 1 - 3x \geq e^{-6x}, \tag{25}$$

where the last inequality holds since $x < 1/4$.

In order to see the first inequality of (25) we distinguish two cases. If $x = 1/n$, then the number of i-literals in t is at most $Tx = T/n$. Since we condition on $\kappa_i \geq \kappa$, the number of i-literals is still at most T/n after the first $\kappa - 1$ operations. The number of leaves after $\kappa - 1 < n$ operations is at least $T - n \geq T/2$. Hence, the probability to pick one of these leaves for deletion or substitution is at most $(2/3)(T/n)/(T/2) < 2/n$. On the other hand, the probability to insert an i-literal or to substitute a leaf with x_i or $\overline{x}_i$ is at most $1/n$. By the union bound, the probability to touch i is at most $3/n$. This proves (25) if $x = 1/n$.

The other case is very similar only involving different numbers. The number of i-literals in t is Tx. Since $k \leq 1/x \leq T/2$, after $\kappa \leq k$ operations the size of the remaining tree is at least $T/2$. Therefore, the probability that M_κ picks an i-literal for deletion or substitution is at most $(2/3)xT/(T/2) \leq 2x$. On the other hand, the probability to insert an i-literal or to substitute a leaf with x_i or $\overline{x}_i$ is at most $1/n \leq x$. By the union bound, the probability to touch i is at most $3x$. This proves (25) if $x = \#\{i\text{-literals}\}/T$.

We can expand (25) to obtain the probability of $\kappa_i = \infty$. For $2 \leq k \leq 1/x$,

$$\Pr[\kappa_i = \infty \mid k] = \prod_{i=1}^{k} \Pr[\kappa_i \geq \kappa + 1 \mid k, \kappa_i \geq \kappa] \geq e^{-6kx},$$

and consequently, for all $1 \leq k \leq 1/x$,

$$\Pr[i \text{ touched} \mid k] = 1 - \Pr[\kappa_i = \infty \mid k] \le 1 - e^{-6kx} \le 6kx.$$

For $k > 1/x$ we will use the bound $\Pr[i \text{ touched} \mid k] \le 1$. To ease notation, we will assume in our formulas that $1/x$ is an integer. Then we may bring both cases together, and bound

$$\begin{aligned}\Pr[k \ge 2 \text{ and } i \text{ touched}] &\le \sum_{\kappa=2}^{1/x} \Pr[k=\kappa]\Pr[i \text{ touched} \mid k=\kappa] + \sum_{\kappa=1+1/x}^{\infty} \Pr[k=\kappa] \\ &\overset{1.}{\le} \sum_{\kappa=2}^{1/x} e^{-\delta\kappa} 6\kappa x + \sum_{\kappa=1+1/x}^{\infty} e^{-\delta\kappa} \le x \sum_{\kappa=2}^{\infty} (6\kappa + \tfrac{1}{x} e^{-\delta/x}) e^{-\delta\kappa} \\ &\le Cx\end{aligned}$$

for a suitable constant $C > 0$, since the function $\frac{1}{x}e^{-\delta/x}$ is upper bounded by a constant for $x \in (0,1]$. Together with (23), we get

$$\begin{aligned}\frac{1}{\Pr[k=1 \mid i \text{ touched}]} &= 1 + \frac{\Pr[k \ge 2 \text{ and } i \text{ touched}]}{\Pr[k=1 \text{ and } i \text{ touched}]} \\ &\le 1 + \frac{Cx}{x/(3e)} = 1 + 3eC.\end{aligned}$$

This proves 2 for $\delta := 1/(1+3Ce)$. For 3 we compute similar as before

$$\begin{aligned}\Pr[k \ge \kappa \text{ and } i \text{ touched}] &\le \sum_{\kappa'=\kappa}^{1/x} \Pr[k=\kappa']\Pr[i \text{ touched} \mid k=\kappa'] + \sum_{\kappa'=\max\{\kappa,1+1/x\}}^{\infty} \Pr[k=\kappa'] \\ &\le \sum_{\kappa'=\kappa}^{1/x} e^{-\delta\kappa'} 6\kappa' x + \sum_{\kappa'=\max\{\kappa,1+1/x\}}^{\infty} e^{-\delta\kappa'} \\ &\le x e^{-\delta\kappa/2} \sum_{\kappa'=1}^{\infty} (6\kappa' + \tfrac{1}{x} e^{-\delta/x}) e^{-\delta\kappa'/2} \le Cx e^{-\delta\kappa/2}\end{aligned}$$

for a suitable constant $C > 0$. Therefore, as before,

$$\begin{aligned}\frac{1}{\Pr[k \ge \kappa \mid i \text{ touched}]} &= 1 + \frac{\Pr[k < \kappa \text{ and } i \text{ touched}]}{\Pr[k \ge \kappa \text{ and } i \text{ touched}]} \ge 1 + \frac{\Pr[k=1 \text{ and } i \text{ touched}]}{\Pr[k \ge \kappa \text{ and } i \text{ touched}]} \\ &\ge 1 + \frac{x/(3e)}{Cxe^{-\delta\kappa/2}} \ge \frac{1}{3eC} e^{\delta\kappa/2}.\end{aligned}$$

This proves 3, since we may decrease δ in order to swallow the constant factor $3eC$ by the term $e^{\delta\kappa/2}$.
4: This follows immediately from 3, because

$$\mathbb{E}[k \mid i \text{ touched}] = \sum_{\kappa \ge 1} \Pr[k \ge \kappa \mid i \text{ touched}] \le 1 + \sum_{\kappa \ge 2} e^{-\delta\kappa},$$

and the latter sum is bounded by an absolute constant. □

5.2.3. Bloat estimation

The main part of the proof is to study how the size of the GP-tree increases. We show that it increases by only a little more than a constant factor within roughly $T_{\text{init}} \log T_{\text{init}}$ rounds if $T_{\text{init}} \in \omega(n \log^2 n)$. However, we need explicit tail bounds on the error probability, which are provided by the following theorem.

Theorem 5.4. *There is $\varepsilon > 0$ such that the following holds. Let $f = f(n) \in \omega(1)$ be any growing function with $f(n) \in o(n)$. Let $T_{\min} := \max\{T_{\text{init}}, f(n)\, n \log^2 n\}$. Then for sufficiently large n, with probability at least $1 - \exp(-\varepsilon\sqrt{f(n)})$, within the next $r_0 := \varepsilon f(n) T_{\min} \log T_{\min}$ rounds the tree has never more than $T_{\max} := \sqrt{f(n)} T_{\min}$ leaves.*

The proof of Theorem 5.4 is the most technical part of the proof and this whole subsection is devoted to it. First, we provide an outline of the basic ideas, adding some actual numbers to the general outline presented in Section 5.2.1. We will couple the size of the GP tree to a different process $S = (S_r)_{r \ge 0}$ on $\mathbb{N}$ which is easier to analyze. The key idea is that we only have a non-trivial drift in rounds in which the offspring is rejected. As we will see later, this event does not

happen often. Formally, we define S by a sum $S_r = T_{\min} + \sum_{j=1}^{r}(X'_j + X_j)$, where X'_j are independent random variables with zero drift, and X_j are only non-zero in critical rounds. Thus S can be regarded as a decomposition into terms which are unbiased, and terms which account for the (small) positive drift of the size of the GP-tree.

The most difficult part is to bound the contribution of the X_j, i.e., to show that most rounds are non-critical. To this end, we will show that the random variables $V(t,i)$, once they are non-negative, follow a random walk with weak drift as described in Theorem 3.6, with parameter $N := \sqrt{T_{\min}/n} \geq \sqrt{f(n)}\log T_{\min}$. For the purpose of this outline we consider only rounds in which at most one variable $i \in [n]$ with $V(t,i) = 0$ is critical. This (almost) covers the case when the number k of mutations in a round is constantly one, but similar arguments transfer to the case when k is $1 + \mathrm{Pois}(1)$-distributed. Whenever i is touched in such a round then $V(t,i)$ has probability $\Omega(1)$ to increase, so the state $V(t,i) = 0$ will only persist for $O(1)$ rounds that touch i. On the other hand, after being increased, it needs in expectation $\Omega(N)$ i-rounds to return to zero. Intuitively, this means that in a random i-round, the probability to encounter $V(t,i) = 0$ is $O(1/N)$. Note that this intuition is not quite correct, but we can use Lemma 3.8 for the formal argument. Since each round touches only $O(1)$ variables, and each of them has only probability $O(1/N)$ to be critical, there are only $O(r_0/N) \in O(\varepsilon\sqrt{f(n)}T_{\min})$ critical rounds within r_0 rounds. Thus the size of the GP-tree grows roughly by at most a constant factor in $T_{\min}\log T_{\min}$ rounds.

Proof of Theorem 5.4. We will prove the theorem under the assumption that the size of the GP-tree never falls below $T_{\min}$. This is justified because we can track the process until either r_0 rounds have passed or the size of the GP-tree falls below $T_{\min}$ in some round $r \leq r_0$. In the former case we are done, in the latter case we apply the same argument again starting in the next round in which the size of the GP-tree exceeds $T_{\min}$.[5]

Let t be the GP-tree in round j, let k be the number of mutations in this round, and let t' be the tree resulting from these mutations. We set $X'_{j+1} := s(t') - s(t)$, and

$$X_{j+1} := \begin{cases} k, & \text{if round } j \text{ is positive critical;} \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

As mentioned in the outline, we define $S_r := T_{\min} + \sum_{j=1}^{r}(X'_j + X_j)$.

$\mathbf{s(t_r) \leq S_r}$. We first show that the size of the GP-tree after r rounds is at most S_r. The fitness of t' can only be smaller than the fitness of t if there is at least one index i for which $V(t,i)$ changes from non-negative to negative, which can only happen in positive critical rounds. In particular, in the second case of (26) we have $f(t') \geq f(t)$, and hence the GP-tree t' is accepted. Thus, in this case we have $S_{r+1} - S_r = X'_{r+1} + X_{r+1} = s(t') - s(t)$, so S_j and the size of the GP-tree both change by the same amount. For the first case of (26), we have $S_{r+1} - S_r = k + s(t') - s(t) \geq \max\{0, s(t') - s(t)\}$. Since the size of the GP-tree changes either by $s(t') - s(t)$ (if t' is accepted) or by 0 (if t' is rejected), the increase of S_r is at least the increase of the size of the GP-tree. Since this is true for all cases, the size of the GP-tree is at most S_r, as claimed. We will derive upper bounds on S_r in the following.

In order to bound $S_r = \sum_{j=1}^{r}(X_j + X'_j)$ we will prove separately that each of the bounds $\sum_{j=1}^{r} X'_j \leq T_{\max}/3$ and $\sum_{j=1}^{r} X_j \leq T_{\max}/3$ hold with probability at least $1 - \exp\{-\Omega(\sqrt{f(n)})\}$. By the union bound, it will follow that *both* bounds together hold with probability at least $1 - \exp\{-\Omega(\sqrt{f(n)})\}$. The two bounds will imply that the size of the GP-tree is at most $T_{\min} + 2T_{\max}/3 \leq T_{\max}$, thus proving the theorem. Recall that we need to consider the range $1 \leq r \leq r_0 = \varepsilon f(n) T_{\min}\log T_{\min}$.

Bounding $\mathbf{X'_j}$. First we bound X'_j. For $\sum_{j=1}^{r} X'_j$, note that each X'_j is the sum of k Bernoulli-type random variables (with values $+1$ for insertion, -1 for deletion, and 0 for substitution), where k is either constantly 1 or $1 + \mathrm{Pois}(1)$-distributed, depending on the algorithm. Let us denote by K_r the total number of Bernoulli-type variables (i.e., the total number of mutations in r rounds). In the case where we always choose $k = 1$, we have trivially $K_r = r$. In the case $k \sim 1 + \mathrm{Pois}(1)$ we have $K_r \sim r + \mathrm{Pois}(r)$ since the sum of independent Poisson distributed random variables is again Poisson distributed. Since $\mathrm{Pois}(r)$ is dominated by $\mathrm{Pois}(r_0)$, we have

$$\Pr[K_r \geq 3r_0] \leq \Pr[\mathrm{Pois}(r_0) \geq 2r_0] \overset{(21)}{\leq} \frac{e^{-r_0}(er_0)^{2r_0}}{(2r_0)^{2r_0}} = \left(\frac{e}{4}\right)^{r_0}$$

for each $r \leq r_0$. Note that this estimate holds also for the case that all k are one, because then the probability on the left is zero. Taking a union bound over all $1 \leq r \leq r_0$ we see that with exponentially high probability[6] $K_r \leq 3r_0$ also holds uniformly for all $1 \leq r \leq r_0$. For each mutation the probability of insertion, deletion, and substitution is 1/3 each, i.e., each of the K_r Bernoulli-type random variables contributes $+1$, -1, or 0, with probability 1/3 each. Thus we may use the Chernoff

[5] We are slightly cheating here, because for $k \sim 1 + \mathrm{Pois}(1)$, the size of the GP-tree may jump to something strictly larger than $T_{\min}$ in one step. However, our proof also works if we start with any GP-tree of size at most $2T_{\min}$, and the probability to increase the size of the GP-tree by more than $T_{\min}$ in one step is negligibly small.

[6] that means with probability $1 - e^{-\Omega(r_0)}$.

bound, Theorem 3.1, with $\delta = r_0^{-1/4}$ to infer that with sufficiently high probability $\sum_{j=1}^{r} X'_j \le r_0^{3/4} < T_{\min}/3$ holds uniformly for all $1 \le r \le r_0$. In particular, this probability is $1 - \exp\{-\Omega(\sqrt{f(n)})\}$.

Bounding X_j: Setup. It remains to bound $\sum_{j=1}^{r} X_j$. Recall that X_j is either zero or the number of mutations applied in the j-th round. Therefore, the sum is non-decreasing in r and it suffices to bound the sum for $r = r_0$. Then the same bound will follow for all $r \le r_0$.

We fix some $i \in [n]$ and consider the random walk of the variable $V(t_r, i)$. Recall that we assume the size of the GP-tree t_r to be at least $T_{\min}$. Since $V(t_r, i)$ can only change in i-rounds, it makes sense to study the random walk by only considering i-rounds. We will apply Theorem 3.6 with $N := \sqrt{T_{\min}/n}$ to this random walk. To this end, in the following paragraphs we prove that the random walk that $V(t_r, i)$ performs in i-rounds satisfies the conditions of Theorem 3.6.

Bounding X_j: Computing the drift. Now we are ready to compute the drift of X_j.
Let us first consider $v \ge 1$, and compute the drift

$$\Delta_{v,i} := \mathbb{E}[V(t_{r+1}, i) - V(t_r, i) \mid V(t_r, i) = v, r \text{ is } i\text{-round}].$$

We mind the reader to not confuse this drift with the drift of S_r, which is a very different concept. The notation $\Delta_{v,i}$ is slightly abusive because the drift does depend on t_r, too. However, we will derive lower bounds on the drift which are independent of t_r, thus justifying the abuse of notation. In fact, we will compute the drift of

$$\Delta'_{v,i} := \mathbb{E}[V(t'_r, i) - V(t_r, i) \mid V(t_r, i) = v, r \text{ is } i\text{-round}],$$

where t'_r is the offspring of t_r. In other words, we ignore whether the offspring is accepted or not. Note that this can only decrease the drift, since a mutation that causes t'_r to be rejected can not increase $V(t_r, i)$. Hence, any lower bound on $\Delta'_{v,i}$ is also a lower bound on $\Delta_{v,i}$.

Let $\mathcal{E}_r$ be the event that r is an i-round. Note that

$$\Pr[\mathcal{E}_r] \in \Omega(1/n), \tag{27}$$

since we always have probability $1/(3n)$ to touch i with an insertion.

Consider any round r conditioned on $\mathcal{E}_r$ and let M be a mutation in round r. If M does not touch i, then M does not change $V(t_r, i)$ and the contribution to the drift is zero. Next we consider the case that M is an insertion of either x_i or $\overline{x}_i$. Both cases are equally likely and the case that M is an insertion contributes zero to the drift. By the same argument, the cases that M substitutes a non-i-literal by x_i or $\overline{x}_i$ cancel out and together contribute zero to the drift.

Next consider deletions of x_i or $\overline{x}_i$. This case is not symmetric, since there are $v \ge 1$ more x_i than $\overline{x}_i$. So we can describe the number of x_i by $x + v$, and the number of $\overline{x}_i$ by x, for some $x \ge 0$. Consider the first x occurrences of x_i. Then the probability that a deletion M picks one of these first x_i equals the probability that M picks one of the $\overline{x}_i$. As before, both cases are equally likely. Therefore, the contribution to the drift from either picking one of the first x occurrences of x_i or any occurrence of $\overline{x}_i$, cancel out. For the remaining v literals x_i the unconditional probability that a deletion picks one of them is $v/s(t_r) \le v/T_{\min}$, where $s(t_r) \ge T_{\min}$ is the current size of the GP-tree. Thus the conditional probability (on $\mathcal{E}_r$) to pick one of them is at most $O(vn/T_{\min})$ by (27). Since the conditional expected number of deletions is $\mathbb{E}[\#\text{ deletions} \mid \mathcal{E}_r] \in O(1)$ by Lemma 5.3, the deletions contribute $-O(vn/T_{\min})$ to the drift $\Delta_{v,i}$. By the same argument we also get a contribution of $-O(vn/T_{\min})$ for substitutions of x_i-literals or $\overline{x}_i$-literals.

Summarizing, the only cases contributing to $\Delta'_{v,i}$ are deletions and substitutions of i-literals, and they contribute not less than $-O(vn/T_{\min})$, which is $-O(\sqrt{n/T_{\min}})$ for $v \le N = \sqrt{T_{\min}/n}$. All other cases contribute zero to $\Delta'_{v,i}$. Therefore, the random walk of $V(t_r, i)$ (where we only consider rounds which touch i) satisfies the first condition of Theorem 3.6 with $N = \sqrt{T_{\min}/n}$.

Bounding X_j: Step sizes and initial increase. Now we check the other two conditions of Theorem 3.6, on step sizes and the initial increase of X_j. The second condition (small steps) of Theorem 3.6 follows from Lemma 5.3. Finally, for the third condition (initial increase) we show that for every $v \le N$, where $N = \sqrt{T_{\min}/n}$ and every n sufficiently large, with probability at least δ the next non-stationary step increases $V(t_r, i)$ by exactly one. Note that by Lemma 5.3, an i-round has probability $\Omega(1)$ to have exactly one mutation. Now we distinguish two cases: if there are less than $s(t_r)/n$ occurrences of x_i then the probability to touch i in any way is $O(1/n)$ and the probability of inserting an x_i-literal is $\Omega(1/n)$. Hence, conditioned on touching i, with probability $\Omega(1)$ the only mutation in this round is an insertion of x_i.

For the other case, assume there are more than $s(t_r)/n \ge T_{\min}/n \in \omega(1)$ occurrences of i-literals. Additionally, assume that $v \le \sqrt{T_{\min}/n} < (1/3)s(t_r)/n$, where the last inequality holds for n large enough since then $T_{\min}/n$ is large enough. Then $\overline{x}_i$ occurs at least half as often as x_i, and thus the probability of deleting or substituting a $\overline{x}_i$-literal is at least half as big as the probability to delete or substitute an x_i-literal. Therefore, a mutation that touches i is with probability $\Omega(1)$ a deletion of $\overline{x}_i$. So in both cases the first mutation that touches i increases $V(t_r, i)$ with probability $\Omega(1)$. This proves that the third condition of Theorem 3.6 is satisfied.

Bounding X_j: Putting everything together. So far, we have shown that $V(t_r, i)$ performs a random walk that satisfies the conditions of Theorem 3.6. Hence, for $0 < v < \varepsilon' N = \varepsilon' \sqrt{T_{\min}/n}$ the expected hitting time of $\{[0, 1, \ldots, v]\}$ when starting at

any value larger than v is $\Omega(\sqrt{T_{\min}/n})$, for a suitable constant $\varepsilon' > 0$. Moreover, with probability $\Omega(1/N)$ the hitting time is at least $\Omega(N^2)$.

Now we have all ingredients to bound the expected number of positive critical rounds. We fix a variable i and some $v \geq 0$ and aim to bound the number of rounds in which $V(t_r, i) = v$ and i is a critical variable. For $v \geq \varepsilon' N \geq \varepsilon' \sqrt{f(n)} \log T_{\min}$, with probability at least $1 - e^{-\Omega(N)} \geq 1 - \exp\{-\Omega(\sqrt{f(n)})\}/T_{\min}$ this does not happen in a specific round by Lemma 5.3. By a union bound, with probability $1 - \exp\{-\Omega(\sqrt{f(n)})\}$ it never happens for any variable i and any of r_0 rounds, with room to spare. So we may assume $0 \leq v < \varepsilon N$. We use Lemma 3.8 to estimate how many i-rounds occur with $V(t_r, i) = v$ before for the first time $V(t_r, i) > v$. For this purpose we check the conditions of Lemma 3.8. In each i-round with $V(t_r, i) = v$, with probability $\Omega(1)$ the value of $V(t_r, i) = v$ increases strictly by Lemma 5.3. On the other hand, once $V(t_r, i) > v$ it takes in expectation at least $\Omega(\sqrt{T_{\min}/n})$ i-rounds before the interval $[0, 1, \dots, v]$ is hit again, and it takes at least $\Omega(T_{\min}/n)$ i-rounds with probability at least $\Omega(\sqrt{n/T_{\min}})$. Thus we are in the situation of Lemma 3.8 with $\delta \in \Omega(1)$ and $s \in \Theta(\sqrt{T_{\min}/n})$.

Let E_i denote the number of i-rounds and let $E_{i,v}$ be the number of i-rounds with $V(t_r, i) = v$. Note that we can only apply Lemma 3.8 if $E_i \geq s$. However, in each round we have probability at least $1/(3n)$ to insert an i-literal. Hence, $\mathbb{E}[E_i] \geq r_0/(3n) \in \Omega(f(n) \log n)$. In particular, by the Chernoff bound, Theorem 3.1, $\Pr[E_i < r_0/(6n)] \in e^{-\Omega(f(n)\log n)} \ll (1/n)e^{-\Omega(f(n))}$. Hence, after a union bound over all i, we observe that with probability $1 - e^{-\Omega(f(n))}$ we have $E_i \geq r_0/(6n)$ for all $1 \leq i \leq n$, and we will assume this henceforth. In particular, $E_i \geq r_0/(6n) \geq s$. Thus we may apply Lemma 3.8 with $r = E_i$ and obtain

$$\mathbb{E}[E_{i,v}] \leq C\sqrt{\frac{n}{T_{\min}}}\mathbb{E}[E_i]$$

for a suitable constant $C > 0$. Moreover, by the tail bound in Lemma 3.8,

$$\begin{aligned}\Pr\left[E_{i,v} \leq 2C\sqrt{\frac{n}{T_{\min}}}E_i\right] &\geq 1 - e^{-r_0/(12ns)} \in 1 - e^{-\Omega(\sqrt{f(n)}\log T_{\min})} \\ &\geq 1 - \frac{1}{nN}e^{-\Omega(\sqrt{f(n)})}. \end{aligned} \tag{28}$$

By a union bound over all i and v we see that with probability $1 - \exp\{-\Omega(\sqrt{f(n)})\}$ the bound $E_{i,v} \leq 2C\sqrt{n/T_{\min}}E_i$ from (28) holds for all $1 \leq i \leq n$ and all $1 \leq v \leq \sqrt{N}$. So again we may assume this from now on.

An i-round with $V(t_r, i) = v$ has probability $e^{-\Omega(v)}$ for i to be critical by Lemma 5.3. Therefore, the expected number of critical rounds within the first r_0 rounds is at most

$$\mathbb{E}[\#\{\text{critical rounds}\}] \leq \sum_{\substack{i\in[n]\\0\leq v\leq \varepsilon N}} e^{-\Omega(v)} \cdot \mathbb{E}[E_{i,v}] \in O\left(\sqrt{\frac{n}{T_{\min}}}\right)\sum_{i\in[n]}\mathbb{E}[E_i]. \tag{29}$$

The bound $e^{-\Omega(v)}$ that an i-round with $V(t_r, i) = v$ is critical holds independently of all previous rounds. Therefore, as before we can use the Chernoff bound to amend (29) by the corresponding tail bound and obtain with probability at least $1 - e^{-\Omega(\sqrt{f(n)})}$ that

$$\#\{\text{critical rounds}\} \leq C'\sqrt{\frac{n}{T_{\min}}}\sum_{i\in[n]}E_i \tag{30}$$

for a suitable constant $C' > 0$.

We bound the sum further by observing that in each round only $O(1)$ literals are touched in expectation and the number of touched literal drops at least exponentially. Therefore, $\sum_{i\in[n]}\mathbb{E}[E_i] \in O(r_0)$ and by standard concentration bounds [11, Theorem 11] with probability $1 - \exp\{-\Omega(\sqrt{f(n)})\}$ the expectation is not exceeded by more than a constant factor. Moreover, by assumption we have $T_{\min} \geq f(n)n\log^2 n$, which implies $T_{\min} \geq (1/2)f(n)n\log^2 T_{\min}$ for sufficiently large n. Hence, with probability $1 - \exp\{-\Omega(\sqrt{f(n)})\}$

$$\begin{aligned}\#\{\text{critical rounds}\} &\in O\left(r_0\sqrt{\frac{n}{T_{\min}}}\right) \in O\left(\frac{r_0}{\sqrt{f(n)}\log T_{\min}}\right) \\ &\leq \tfrac{1}{12}\sqrt{f(n)}T_{\min},\end{aligned}$$

where the last step follows from $r_0 = f(n)\varepsilon T_{\min}\log T_{\min}$ if $\varepsilon > 0$ is sufficiently small. Since X_j is zero in non-critical rounds and is bounded by $1 + \text{Pois}(1)$ in critical rounds, as before we may use [11, Theorem 11] to get the following tail bound.

$$\Pr\left[\sum_{j=1}^{r_0}X_j \leq \tfrac{1}{3}\sqrt{f(n)}T_{\min}\right] \in 1 - e^{-\Omega(\sqrt{f(n)})}.$$

Thus we have shown that with sufficiently large probability $\sum_{j=1}^{r_0} X_j \le \frac{1}{3}\sqrt{f(n)}T_{\min} = T_{\max}/3$. This proves the desired bound on S_r and thus concludes the proof of Theorem 5.4. □

5.2.4. Run Time Bound

For technical reasons, before we prove Theorem 5.2 on the expected runtime, we first need to prove a rather technical statement that holds with high probability.

Lemma 5.5. *There is $\varepsilon > 0$ such that the following holds for any growing function $f(n) \in \omega(1)$ with $f(n) \in o(n)$. Let $T_{\min} := \max\{T_{\text{init}}, f(n)n\log^2 n\}$. If n is sufficiently large, then for any starting tree, with probability at least $1 - \exp\{-f(n)^{1/4}\}$ the $(1+1)$ GP without bloat control on* Majority *finds a global optimum within $r_0 := \varepsilon f(n)T_{\min}\log T_{\min}$ rounds, and the size of the GP-tree never exceeds $T_{\max} = \sqrt{f(n)}T_{\min}$.*

Proof. We already know by Theorem 5.4 that with probability $1 - \exp\{-\Omega(\sqrt{f(n)})\}$ the size of the GP-tree does not exceed $T_{\max}$ within r_0 rounds. We fix a variable i, which is not expressed at the beginning, and consider $V'(t_r, i) := \max\{-V(t_r, i), 0\}$. We claim that $V'(t_r, i)$ has a multiplicative drift,

$$\mathbb{E}[V'(t_r, i) - V'(t_{r+1}, i) \mid V'(t_r, i) = v] \ge \frac{v}{3eT_{\max}}, \tag{31}$$

for all $v \ge 0$, as long as i is not expressed. In order to prove (31) we first consider insertions. It is equally likely to insert x_i (which decreases $V'(t_r, i)$) and $\overline{x}_i$ (which increases $V'(t_r, i)$). Moreover, whenever the offspring is accepted after inserting $\overline{x}_i$, it is also accepted after inserting x_i. Therefore, the contribution to the drift from insertions is at least zero. Analogously, substituting a non-i-literal by an i-literal contributes at least zero to the drift. For deletions, with probability at least $1/(3e)$ we have exactly one mutation, and this mutation is a deletion. In this case, the probability to delete a $\overline{x}_i$-literal is exactly by $v/s(t_r) \ge v/T_{\max}$ larger than the probability to delete an x_i-literal. Since we always accept deleting a single $\overline{x}_i$-literal, this case contributes no less than $-v/(3eT_{\max})$ to the drift. For all the other cases (several deletions, substitutions of one or several i-literals), it is always more likely to pick an $\overline{x}_i$-literal for deletion/substitution than an x_i-literal and it is more likely to accept the offspring if an $\overline{x}_i$-literal is deleted/substituted. Therefore, these remaining cases contribute at least zero to the drift. This proves (31).

We next show that for $V(t_r, i) = 0$ in the next i-round with probability $\Omega(1)$ the literal x_i is expressed in the offspring and no other literal becomes unexpressed. We call such a round *i-fixing*. Note that the number of expressed literals can never decrease, so x_i can only become unexpressed if a literal x_j becomes expressed in the same round. In this case we can just swap the roles of i and j for the remainder of the argument. So we may assume that after an i-fixing round the literal x_i stays expressed forever. Then it suffices to show that for every i, if i is not expressed for a sufficient number of rounds, then there is an i-fixing round.

Note that a sufficient condition for an i-fixing round is that there is only a single mutation which inserts a new x_i-literal or deletes an $\overline{x}_i$-literal. The probability to insert a new x_i-literal equals the probability to insert a new $\overline{x}_i$-literal, to create an x_i-literal by substitution or to create an $\overline{x}_i$-literal by substitution. On the other hand, the probability to delete an $\overline{x}_i$-literal equals the probability to delete an x_i-literal (since $V(t_r, i) = 0$), to substitute an x_i-literal and to substitute an $\overline{x}_i$-literal. Thus, the probability that an i-round with only a single mutation is i-fixing is at least $1/3$. Moreover, an i-round has probability $\Omega(1)$ to consist of a single mutation by Lemma 5.3. This proves that for $V(t_r, i) = 0$ the next i-round has probability $\Omega(1)$ to be i-fixing.

By the Multiplicative Drift Theorem 3.4, $V'(t_r, i)$ reaches 0 after at most $r_{\text{init}} := 3eT_{\max}(k + \log T_{\max})$ steps with probability at least $1 - e^{-k}$, for a parameter $k > 0$ that we fix later. Moreover, once at 0 the next i-round is i-fixing with probability $\Omega(1)$. If it is not i-fixing, then $V'(t_r, i)$ may jump from 0 to a positive value. This value will be at most k with probability at least $1 - e^{-\Omega(k)}$ by Lemma 5.3, and again by the Multiplicative Drift Theorem $V'(t_r, i)$ will return to 0 after $r_{\text{return}} := 3eT_{\max}(k + \log\log k + O(1)))$ steps with probability at least $1 - e^{-\Omega(k)}$. Assume this pattern repeats up to $C\log k$ times, for a sufficiently large constant $C > 0$. Then the probability that there is an i-fixing round with $V'(t_r, i) = 0$ is at least $1 - e^{-\Omega(k)}$. It remains to estimate the number of rounds spent in the state $V'(t_r, i) = 0$. Since each round has probability at least $1/(3n)$ to be an i-round, among any $r_{\text{fix}} := 6Cn\log k$ rounds there will be at least $C\log k$ i-rounds with probability at least $1 - e^{-\Omega(k)}$. In particular, if we spend $6Cn\log k$ rounds in the state $V'(t_r, i) = 0$, then with probability at least $1 - e^{-\Omega(k)}$ at least $C\log k$ of them will be i-rounds. By a union bound, the probability that there is an i-fixing round with $V'(t_r, i) = 0$ within $r_{\text{total}} := r_{\text{init}} + C\log k r_{\text{return}} + r_{\text{fix}}$ rounds is $1 - O(e^{-\Omega(k)}\log k) \ge 1 - e^{-\Omega(k)}$, where the latter bound holds if k is sufficiently large.

By a union bound over all i, with probability $1 - ne^{-\Omega(k)}$ all indices will be fixed after at most $r_{\text{total}} \in O(T_{\max}k\log k)$ steps. Choosing $k = f^{1/3}\log T_{\min}/(\log f(n) + \log\log T_{\min})$ gives $ne^{-\Omega(k)} \le \exp\{-f(n)^{1/4}\}$ and $r_{\text{total}} \le r_0$, both with room to spare. This proves the lemma. □

Finally we are ready to prove Theorem 5.2.

Proof of Theorem 5.2. The theorem essentially follows from Lemma 5.5 by using restarts. Let $f(n) \in \omega(1)$ be a growing function such that $f(n) \le n$. We define a sequence $(T_i)_{i\ge 0}$ recursively by $T_0 := T_{\min} = \max\{T_{\text{init}}, n\log^2 n\}$ and $T_{i+1} :=$

$\sqrt{f(n)}T_i$. Moreover, we define $r_i := \varepsilon f(n) T_i \log T_i$, where $\varepsilon > 0$ is the constant from Lemma 5.5. Note that T_i and r_i are chosen such that when we start with any GP-tree of size T_i, then with probability at least $1 - \exp\{-f(n)^{1/4}\}$ a global optimum is found within the next r_{i+1} rounds without exceeding size T_{i+1}.

By Lemma 5.5 there is a high chance to find an optimum in r_0 rounds without increasing the size of the GP-tree too much. In this case, the optimization time is at most r_0. For the other case, the probability that either the global optimum is not found or the size of the GP-tree exceeds T_1 is at most $p := \exp\{-f(n)^{1/4}\}$. Let t_1 be the GP-tree at the first point in time where something goes wrong, i.e., we set t_1 to be the first GP-tree of size larger than T_1, if this happens within the first r_0 rounds; otherwise we set t_1 to be the GP-tree after r_0 rounds. In either case, t_1 is a GP-tree of size at most T_1. Then we do a restart, i.e., we apply Lemma 5.5 again with t_1 as the starting tree. Similar as before, there is a high chance to find an optimum in r_1 rounds without blowing up the GP-tree too much. Otherwise (with probability at most p), we define t_2 to be the first GP-tree with size at least T_2, if such a tree exists before round $r_0 + r_1$; otherwise, we let t_2 be the tree at time $r_0 + r_1$. Repeating this argument, the expected optimization time T_{opt} is at most

$$\mathbb{E}[T_{\text{opt}}] \le r_0 + p\,(r_1 + p\,(r_2 + p\,(\ldots))) = \sum_{i=0}^{\infty} p^i r_i = \varepsilon f(n) \sum_{i=0}^{\infty} p^i T_i \log T_i$$

By the recursive definition we see that $T_i = f(n)^{i/2} T_{\min}$. In particular, using that $p\sqrt{f(n)} < 1/2$ for sufficiently large n we obtain

$$\begin{aligned}
\mathbb{E}[T_{\text{opt}}] &\le \varepsilon f(n) \sum_{i=0}^{\infty} 2^{-i} T_{\min} \log\left(f(n)^{i/2} T_{\min}\right) \\
&= \varepsilon f(n) T_{\min} \left(\log(T_{\min}) \sum_{i=0}^{\infty} 2^{-i} + \log(f(n)) \sum_{i=0}^{\infty} 2^{-i} \frac{i}{2}\right) \\
&\overset{f(n)<n<T_{\min}}{\le} 3\varepsilon f(n) T_{\min} \log T_{\min}.
\end{aligned}$$

This shows that for every arbitrarily slowly growing function $f(n)$ we have $\mathbb{E}[T_{\text{opt}}] \le 3\varepsilon f(n) T_{\min} \log T_{\min}$. We claim that we may replace the function $f(n)$ by a constant, i.e., that $\mathbb{E}[T_{\text{opt}}] \le 3\varepsilon C T_{\min} \log T_{\min}$ for a suitable constant $C > 0$. Assume otherwise for the sake of contradiction, i.e., assume that for every constant $C > 0$ there are arbitrarily large n_C and GP-trees t_C of size T_C such that $\mathbb{E}[T_{\text{opt}} \mid t_{\text{init}} = t_C] > 3\varepsilon C T_C \log T_C$. Then we choose a growing sequence C_i (for instance $C_i = i$). Since for each C_i there are arbitrarily large counterexamples n_{C_i}, t_{C_i}, we may choose a growing sequence $n_{C_1} < n_{C_2} < n_{C_3} < \ldots$ of counterexamples. Now we define $f(n) := \min\{i \mid n_{C_i} > n\} \in \omega(1)$ and obtain a contradiction, since we have an infinite sequence of counterexamples for which $\mathbb{E}[T_{\text{opt}}] > 3\varepsilon f(n) T_{\min} \log T_{\min}$. Hence we have shown for a suitable constant $C > 0$ that $\mathbb{E}[T_{\text{opt}}] \le 3\varepsilon C T_{\min} \log T_{\min}$. This proves the theorem, since $T_{\min} \log T_{\min} \in \Theta(\max\{T_{\text{init}} \log T_{\text{init}}, n \log^3 n\})$. □

6. Conclusion

We considered a simple mutational genetic programming algorithm, the $(1+1)$ GP, and studied the two simple problems Order and Majority. It turns out that for these problems, optimization is efficient in spite of the possibility of bloat: except for logarithmic factors, all run times are linear. However, bloat and the variable length representations were not easily analyzed, but required rather deep insights into the optimization process and the growth of the GP-trees.

For optimization preferring smaller GP-trees we observed a very efficient optimization behavior: whenever there is a significant number of redundant leaves, these leaves are pruned. Whenever only few redundant leaves are present, the algorithm easily increases the fitness of the GP-tree.

For optimization without bloat control, we were able to show that the extent of bloat is not too excessive during the optimization process, meaning that the tree is only larger by at most multiplicative polylogarithmic factors. Since this is an upper bound, the real bloat might be smaller. While polylogarithmic factors are not a major obstacle for a theoretical analysis, a solution which is not even linear in the optimal solution might not be desirable from a practical point of view. So if bloat does occur then for obtaining small solutions, some kind of bloat control should be used.

From our analysis we witnessed an interesting option for bloat control: by changing the probabilities such that deletions are more likely than insertions we would observe in the presented drift equations a bias towards shorter solutions. Overall, this would lead to faster optimization.

Declaration of competing interest

The authors have no conflict of interest with respect to this manuscript.

References

[1] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 2. edition, MIT Press, 2001.
[2] Benjamin Doerr, Leslie Ann Goldberg, Adaptive drift analysis, Algorithmica 65 (1) (2013) 224–250.
[3] Benjamin Doerr, Timo Kötzing, J.A. Gregor Lagodzinski, Johannes Lengler, Bounding bloat in genetic programming, in: Proc. of GECCO'17, ACM, 2017, pp. 921–928.
[4] Benjamin Doerr, Andrei Lissovoi, Pietro S. Oliveto, Evolving boolean functions with conjunctions and disjunctions via genetic programming, in: Proc. of GECCO'19, ACM, 2019, pp. 1003–1011.
[5] Devdatt P. Dubhashi, Alessandro Panconesi, Concentration of Measure for the Analysis of Randomized Algorithms, Cambridge University Press, 2009.
[6] Greg Durrett, Frank Neumann, Una-May O'Reilly, Computational complexity analysis of simple genetic programming on two problems modeling isolated program semantics, in: Proc. of FOGA'11, 2011, pp. 69–80.
[7] David E. Goldberg, Una-May O'Reilly, Where does the good stuff go, and why? How contextual semantics influences program structure in simple genetic programming, in: Proc. of EuroGP'98, 1998, pp. 16–36.
[8] Geoffrey Grimmett, David Stirzaker, Probability and Random Processes, Oxford University Press, 2001.
[9] Jun He, Xin Yao, A study of drift analysis for estimating computation time of evolutionary algorithms, Nat. Comput. 3 (1) (2004) 21–35.
[10] Daniel Johannsen, Random Combinatorial Structures and Randomized Search Heuristics, PhD thesis, Universität des Saarlandes, 2010.
[11] Timo Kötzing, Concentration of first hitting times under additive drift, Algorithmica 75 (3) (2016) 490–506.
[12] Timo Kötzing, J.A. Gregor Lagodzinski, Johannes Lengler, Anna Melnichenko, Destructiveness of lexicographic parsimony pressure and alleviation by a concatenation crossover in genetic programming, in: Proc. of PPSN'18, Springer, 2018, pp. 42–54.
[13] Timo Kötzing, Frank Neumann, Reto Spöhel, PAC learning and genetic programming, in: Proc. of GECCO'11, 2011, pp. 2091–2096.
[14] Timo Kötzing, Andrew M. Sutton, Frank Neumann, Una-May O'Reilly, The Max problem revisited: the importance of mutation in genetic programming, in: Proc. of GECCO'12, 2012, pp. 1333–1340.
[15] Johannes Lengler, Angelika Steger, Drift analysis and evolutionary algorithms revisited, Comb. Probab. Comput. 27 (4) (2018) 643–666.
[16] Andrei Lissovoi, Pietro S. Oliveto, Computational complexity analysis of genetic programming, in: Benjamin Doerr, Frank Neumann (Eds.), Theory of Evolutionary Computation: Recent Developments in Discrete Optimization, Springer International Publishing, Cham, 2020, pp. 475–518, Also available at https://arxiv.org/abs/1811.04465.
[17] Andrei Lissovoi, Pietro Simone Oliveto, On the time and space complexity of genetic programming for evolving boolean conjunctions, in: Proc. of AAAI'18, 2018, pp. 1363–1370.
[18] Sean Luke, Liviu Panait, Lexicographic parsimony pressure, in: Proc. of GECCO'02, 2002, pp. 829–836.
[19] Andrea Mambrini, Luca Manzoni, A comparison between geometric semantic GP and cartesian GP for Boolean functions learning, in: Proc. of GECCO'14, 2014, pp. 143–144.
[20] Andrea Mambrini, Pietro Simone Oliveto, On the analysis of simple genetic programming for evolving Boolean functions, in: Proc. of EuroGP'16, 2016, pp. 99–114.
[21] Michael Mitzenmacher, Eli Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis, Cambridge University Press, New York, NY, USA, 2005.
[22] Alberto Moraglio, Andrea Mambrini, Luca Manzoni, Runtime analysis of mutation-based geometric semantic genetic programming on Boolean functions, in: Proc. of FOGA'13, 2013, pp. 119–132.
[23] Frank Neumann, Computational complexity analysis of multi-objective genetic programming, in: Proc. of GECCO'12, 2012, pp. 799–806.
[24] Anh Nguyen, Tommaso Urli, Markus Wagner, Single- and multi-objective genetic programming: new bounds for weighted ORDER and MAJORITY, in: Proc. of FOGA'13, 2013, pp. 161–172.
[25] Una-May O'Reilly, An Analysis of Genetic Programming, PhD thesis, Carleton University, Ottawa, Canada, 1995.
[26] Una-May O'Reilly, Franz Oppacher, Program search with a hierarchical variable length representation: Genetic programming, simulated annealing and hill climbing, in: Proc. of PPSN'94, 1994, pp. 397–406.
[27] Tommaso Urli, Markus Wagner, Frank Neumann, Experimental supplements to the computational complexity analysis of genetic programming for problems modelling isolated program semantics, in: Proc. of PPSN'12, 2012, pp. 102–112.
[28] Markus Wagner, Frank Neumann, Parsimony pressure versus multi-objective optimization for variable length representations, in: Proc. of PPSN'12, 2012, pp. 133–142.
[29] Markus Wagner, Frank Neumann, Single- and multi-objective genetic programming: new runtime results for sorting, in: Proc. of CEC'14, 2014, pp. 125–132.
[30] Markus Wagner, Frank Neumann, Tommaso Urli, On the performance of different genetic programming approaches for the sorting problem, Evol. Comput. 23 (4) (2015) 583–609.
[31] Carsten Witt, Tight bounds on the optimization time of a randomized search heuristic on linear functions, Comb. Probab. Comput. 22 (2) (2013) 294–318.