Bounding Bloat in Genetic Programming

Benjamin Doerr
École Polytechnique
Laboratoire d'Informatique (LIX)
Paris-Saclay, France
doerr@lix.polytechnique.fr

Timo Kötzing
Hasso Plattner Institute
Potsdam, Germany
timo.koetzing@hpi.de

J. A. Gregor Lagodzinski
Hasso Plattner Institute
Potsdam, Germany
gregor.lagodzinski@hpi.de

Johannes Lengler
ETH Zürich
Zürich, Switzerland
johannes.lengler@inf.ethz.ch

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 (unnecessary growth of solutions), slowing down optimization. Theoretical analyses could so far not bound bloat and required explicit assumptions on the magnitude of bloat.

In this paper we analyze bloat in mutation-based genetic programming for the two test functions ORDER and MAJORITY. We overcome previous assumptions on the magnitude of bloat and give matching or close-to-matching upper and lower bounds for the expected optimization time.

In particular, we show that the (1+1) GP takes (i) $\Theta(T_{\text{init}} + n \log n)$ iterations with bloat control on ORDER as well as MAJORITY; and (ii) $O(T_{\text{init}} \log T_{\text{init}} + n(\log n)^3)$ and $\Omega(T_{\text{init}} + n \log n)$ (and $\Omega(T_{\text{init}} \log T_{\text{init}})$ for $n = 1$) iterations without bloat control on MAJORITY.

CCS CONCEPTS

•Theory of computation → Optimization with randomized search heuristics; Genetic programming; Theory of randomized search heuristics;

KEYWORDS

Genetic Programming, Mutation, Theory, Run Time Analysis

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 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, a setting notorious for being 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 bloat), see [NUW13] for an overview.

We use recent advances from drift theory, as well as 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 MAJORTIY as introduced in [GO98]. This is in contrast to other theoretical work on GP algorithms which considered the PAC learning framework [KNS11] or the Max-Problem [KSNO12], as well Boolean functions [MMM13, MM14, MO16].

Individuals for ORDER and MAJORTIY are binary trees, where each inner node is labeled J (short for *join*, but without any associated semantics) and leaves are labeled with variable symbols; we call such trees *GP-trees*. The set of variable symbols is $\{x_i \mid i \leq n\} \cup \{\overline{x}_i \mid i \leq n\}$, for some n. In particular, variable 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 symbol x_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 variable symbols x_i.

For the MAJORITY problem, fitness is assigned to GP-trees as follows. We call a variable symbol x_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 variable symbols x_i.

A first analysis of genetic programming on ORDER and MAJORITY was made in [DNO11]. 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

GECCO'17, Berlin, Germany

DOI: 10.1145/nnnnnnn.nnnnnnn

Table 1: Summary of best known bounds. Note that $T_{\max}$ denotes the maximal size of the best-so-far tree in the run (bounds involving $T_{\max}$ are conditional bounds).

Problem	k	Without Bloat Control	With Bloat Control
ORDER	1	$O(nT_{\max})$, [DNO11]	$\Theta(T_{\text{init}} + n \log n)$, [Neu12]
	$1 + \text{Pois}(1)$	$O(nT_{\max})$, [DNO11]	$\Theta(T_{\text{init}} + n \log n)$, Theorem 4.1
MAJORITY	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	$\Theta(T_{\text{init}} + n \log n)$, [Neu12]
	$1 + \text{Pois}(1)$	$O(T_{\text{init}} \log T_{\text{init}} + n \log^3 n)$, 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

relabels this leaf (to a random variable 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 uses a parameter k which determines how many such operations make up an atomic mutation; in the simplest case, $k = 1$, 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).

A straightforward version of bloat control for this algorithm is to always prefer the smaller of two trees, given equal fitness. Introducing this simple bloat control, [Neu12] 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 a (expected) optimization time of $\Theta(T_{\text{init}} + n \log n)$ (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 [DNO11] 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 $k = 1$, [DNO11] gives the conditional bound of $O(n^2 T_{\max} \log n)$. We focus on improving the bound for MAJORITY and get, for both $k = 1$ and $k = 1 + \text{Pois}(1)$, a bound of $O(T_{\text{init}} \log T_{\text{init}} + n \log^3 n)$ (see Theorem 5.2). The proof of this theorem requires significant machinery for bounding the extent of bloat during the run of the optimization.

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 the analysis of drift. Here we state a selection of known theorems as well as a new one (Theorem 3.3), which 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. Due to space restrictions, some proofs have been moved to the appendix.

2 PRELIMINARIES

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, whose only function is the join operator (denoted by J). The terminal set X consists of $2n$ variables, where $\overline{x}_i$ is the complement of x_i:

- $F_S := \{J\}$, J has arity 2,
- $X := \{x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\}$.

For a given syntax tree tree t the value of the tree is computed by parsing the tree inorder and generating the set S of *expressed* literals in this way. For ORDER a literal i is expressed if a variable x_i is present in t and there is no $\overline{x}_i$ that is visited in the inorder parse before the first occurrence of x_i. For MAJORITY a literal i is expressed if a variable x_i is present in t and the number of variables x_i is at least the number of variables $\overline{x}_i$.

In this paper we consider simple mutation-based genetic programming algorithms which use the operator HVL-Prime as discussed in [DNO11]. HVL-Prime allows to produce trees of variable length by applying three different operations: insert, delete and substitute (see Figure 1). Each application of HVL-Prime chooses one of these three operations uniformly at random, whereas k denotes the number of applications of HVL-Prime we allow for each mutation.

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.

Given a GP-tree t, mutate t by applying HVL-Prime k times. 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.

Figure 1: Mutation operator HVL-Prime

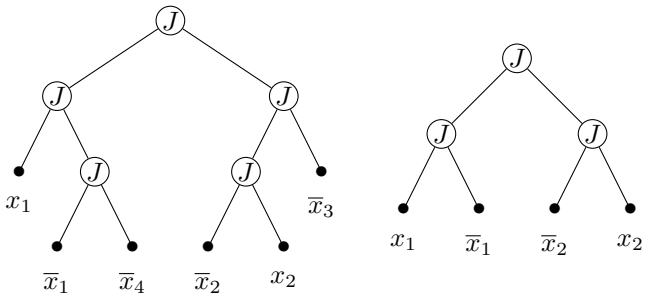

Figure 2: Two GP-trees with the same fitness. For ORDER the fitness is 1, since only the first literal occurs with a non-negated variable first. For MAJORITY the fitness is 2, since the literal 1 and 2 have one variable x_i and also one variable $\overline{x}_i$. However, the left one has complexity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

We consider two problems. The first one is the single problem of computing a tree t which maximizes F. During an optimization run we can use the complexity C to generate an order for solutions with the same fitness by preferring solutions with smaller complexity (see Figure 2). This gives us a way of breaking ties between solutions with the same fitness. Hence, the second problem consists of maximizing the multi-objective function given by F and C.

Consequently, we study the following problems:

- ORDER and MAJORITY without bloat control, which consist of maximizing the given function without studying the complexity.
- ORDER and MAJORITY with bloat control, which consist of maximizing the given function and preferring solutions with smaller complexity, if two solutions have the same function value.

To solve these problems we study the (1+1) GP proceeding 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 is (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 i.i.d. The (1+1) GP then produces an offspring from the best-so-far individual by applying mutation k times in a row; the offspring is discarded if its fitness is worse than the best-so-far, otherwise it is kept to replace the previous best-so-far. Algorithm 1 states the (1+1) GP more formally.

Algorithm 1: (1+1) GP

```
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) then t ← t';
```

3 DRIFT THEOREMS

We will use a variety of drift theorems to derive the theorems 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 start with a theorem for *additive drift*.

Theorem 3.1 (Additive Drift [HY04]). *Let $(X_t)_{t\geq 0}$ be random variables describing a Markov process over a finite state space $S \subseteq \mathbb{R}$. Let T be the random variable that denotes the earliest point in time $t \geq 0$ such that $X_t = 0$. If there exist $c > 0$ such that*

$$E(X_t - X_{t+1} \mid T > t) \leq c,$$

then

$$E(T \mid X_0) \geq \frac{X_0}{c}.$$

[Joh10, Theorem 4.6] gives a *variable drift theorem* for situations where the drift is not uniform across the search space; frequently one can find a uniform lower bound and use the additive drift theorem, but using the variable drift theorem will typically give much better bounds. The version of this theorem that we use is due to [RS12], which removes the restriction of h being differentiable (in our application of the variable drift theorem, this restriction is in fact *not* met).

Theorem 3.2 (Variable Drift [RS12]). *Let $(X_t)_{t\geq 0}$ be random variables describing a Markov process over a finite state space $S \subseteq \mathbb{R}_0^+$ and let $x_{\min} := \min\{x \in S \mid x > 0\}$. Furthermore, let T be the random variable that denotes the first point in time $t \in \mathbb{N}$ for which*

$X_t = 0$. *Suppose that there exists a monotone increasing function* $h : \mathbb{R}^+ \to \mathbb{R}^+$ *such that* $1/h$ *is integrable and*

$$E(X_t - X_{t+1} \mid X_t) \geq h(X_t)$$

holds for all $t < T$. *Then,*

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

For our lower bounds we need the following new drift theorem.

Theorem 3.3 (Multiplicative Drift, bounded step size). *Let* $(X_t)_{t \geq 0}$ *be random variables describing a Markov process over a finite state space* $S \subseteq \mathbb{R}_+$ *with minimum* 1. *Let* $s_{\min} \geq \sqrt{2}\kappa$ *and* T *be the random variable that denotes the earliest point in time* $t \geq 0$ *such that* $X_t \leq s_{\min}$. *If there exist positive reals* κ, $\delta > 0$ *such that for all* $s > s_{\min}$ *and* $t \geq 0$ *with* $P(X_t = s) > 0$ *it holds that*

(1) $|X_t - X_{t+1}| \leq \kappa$, *and*
(2) $E(X_t - X_{t+1} \mid X_t = s) \leq \delta s$,

then, for all $s_0 \in S$ *with* $P(X_0 = s_0) > 0$,

$$E(T|X_0 = s_0) \geq \frac{1 + \ln(s_0) - \ln(s_{\min})}{2\delta + \frac{\kappa^2}{s_{\min}^2 - \kappa^2}}.$$

4 RESULTS WITH BLOAT CONTROL

In this section we show the following theorem.

Theorem 4.1. *The (1+1) GP* with bloat control *on ORDER and MAJORITY takes* $\Theta(T_{\text{init}} + n \log n)$ *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, and the expected number of deletions performed by (1+1) GP is O(1), we get a lower bound of T_{init} from the additive drift theorem (Theorem 3.1).

4.2 Upper Bound

The following subsection is dedicated to the proof of the upper bound. Let T_{init}, and n be given. We want to apply a theorem concerning variable drift (Theorem 3.2). In order to construct a suitable potential function, we partition the leaves of a GP-tree t into three pairwise disjoint sets:

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

We denote the number of expressed literals of t by $v(t)$. Additionally, we denote by $r(t)$, $c^+(t)$ and $c^-(t)$ the cardinality of $R(t)$, $C^+(t)$ and $C^-(t)$, respectively. Before proving the upper bound on the expected optimization time we are going to proof upper bounds on the number of critical leaves.

Lemma 4.2. *Let* t *be a GP-tree, then for ORDER and MAJORITY holds*

$$c^-(t) \leq r(t) + v(t).$$

Lemma 4.3. *Let* t *be a GP-tree, then for ORDER and MAJORITY holds*

$$c^+(t) \leq 2r(t).$$

Given the best-so-far GP-tree t, then $v(t)$ is the number of expressed variables x_i, for $i \leq n$. Let $s(t)$ be the number of leaves of t (we call it the *size* of t). Due to the above defined sets we obtain

$$s(t) = r(t) + c^+(t) + c^-(t). \tag{1}$$

Concerning the construction of a potential function we want to strongly reward 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. Thus, we associate with t the potential function

$$g(t) = 10(n - v(t)) + s(t) - v(t).$$

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, since $r(t)$ is 0, by Lemma 4.2 and Lemma 4.3 $s(t) - v(t)$ is also 0.

Let t be a GP-tree, the best-so-far tree in a run of the (1+1) GP. Let t' be the random variable describing the best-so-far solution in the next iteration. We are going to derive a bound on the expected drift $g(t) - g(t')$, which we denote by $\Delta(t)$.

For this purpose we will distinguish between the case D_1, where the algorithm chooses to do exactly one operation in the observed mutation step, and 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

$$P(D_1) = P(\text{Pois}(1) = 0) = \frac{1}{e},$$
$$P(D_2) = 1 - \frac{1}{e}.$$

Furthermore, let E be the event that $v(t') = v(t)$. Note that, conditional on E, the potential cannot increase since the number of leaves can only decrease. However, conditional on $\overline{E}$, the potential can decrease since every addition of a leaf is accepted as long as the fitness increases.

Lemma 4.4. *For the expected negative drift measured by* $g(t)$ *conditional on* D_2 *holds*

$$E(\Delta(t) \mid D_2) \geq -\frac{1}{e}\left(4 \cdot 10^{-7}\right).$$

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

$$E(\Delta(t) \mid D_2) \geq -\frac{7g(t)}{10en} \sum_{i=11}^{\infty} i(i-10)\frac{1}{(i-1)!}.$$

Proof. First we note that the drift conditional on D_1 is always positive, since with one operation the algorithm cannot increase the size by more than 1. However, this is necessary for an accepted offspring with improved fitness and increased number of redundant leaves.

Concerning the drift conditional on D_2 we observe

$$E(\Delta(t) \mid D_2) \geq -E(-\Delta(t) \mid \overline{E})P(\overline{E}),$$

since the drift can be negative in this case. In particular, we observe a drift of at least 10 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, since 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 $i \geq 1$,

$$P(Y = i) = \frac{1}{e(i-1)!}.$$

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

$$\begin{aligned} E(-\Delta(t) \mid \overline{E}) &= \sum_{i=0}^{\infty} E(-\Delta(t) \mid Y = i, \overline{E}) P(Y = i \mid \overline{E}) \\ &\leq \sum_{i=0}^{\infty} (i-10) P(Y = i \mid \overline{E}) \\ &\leq \sum_{i=11}^{\infty} (i-10) P(Y = i \mid \overline{E}). \end{aligned}$$

Due to Baye's theorem we derive

$$E(-\Delta(t) \mid \overline{E}) \leq \sum_{i=11}^{\infty} (i-10) P(\overline{E} \mid Y = i) \frac{P(Y = i)}{P(\overline{E})},$$

which yields the first bound by pessimistically assuming $P(\overline{E} \mid Y = i) = 1$

$$\begin{aligned} E(\Delta(t) \mid D_2) &\geq -\sum_{i=11}^{\infty} (i-10) P(Y = i) \\ &= -\frac{1}{e}\left(2e - 10e + \sum_{i=1}^{10} \frac{10-i}{(i-1)!}\right) \\ &\geq -\frac{1}{e}\left(4 \cdot 10^{-7}\right). \end{aligned} \tag{2}$$

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

The event $\overline{E}$ 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. Due to $n - v(t) \leq g(t)/10$ adding such a positive literal has a probability of at most

$$\frac{n - v(t)}{6n} \leq \frac{g(t)}{60n}$$

per operation. Regarding the deletion of negative literals, there are at most $s(t) - v(t)$ negative literals. Hence, due to $s(t) \leq g(t)$ 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 p_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

$$\begin{aligned} P(\overline{E} \mid Y = i) &\leq \bigcup_{l=1}^{i} q_l \leq \sum_{l=1}^{i} q_i = \frac{ig(t)}{n}\left(\frac{1}{60} + \frac{2}{3}\right) \\ &< \frac{ig(t)}{n}\frac{7}{10}. \end{aligned}$$

Therefore, we obtain an expected drift conditional on D_2 of

$$\begin{aligned} E(\Delta(t) \mid D_2) &\geq -E(-\Delta \mid \overline{E}) P(\overline{E}) \\ &\geq -\frac{7g(t)}{10en} \sum_{i=11}^{\infty} i(i-10)\frac{1}{(i-1)!}. \end{aligned}$$

□

We are now going to proof the upper bound by deriving the expected positive drift matching the negative drift given by Lemma 4.4. To do so, 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 literals 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 literals are expressed. In order to apply the second bound given by Lemma 4.4, we will split the second stage in two cases.

Case 1: We first consider the case $r(t) \geq v(t)$. Due to Lemma 4.2, Lemma 4.3 and Equation (1) 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 E. Hence, we obtain for the event D_1

$$E(\Delta(t) \mid D_1, E) P(E) \geq \frac{1}{15}.$$

Additionally, the drift conditional on D_1 is always positive, which yields

$$E(\Delta(t) \mid D_1) \geq E(\Delta(t) \mid D_1, E) P(E) \geq \frac{1}{15}.$$

The drift conditional on D_2 is given by Lemma 4.4. Overall, we get a constant drift in the case of $r(t) \geq v(t)$ due to the law of total expectation

$$\begin{aligned} E(\Delta(t)) &\geq E(\Delta(t) \mid D_1) P(D_1) + E(\Delta(t) \mid D_2) P(D_2) \\ &\geq \frac{1}{15e} - \frac{1}{e}\left(1 - \frac{1}{e}\right)\left(4 \cdot 10^{-7}\right) \\ &\geq \frac{1}{e}\left(\frac{1}{15} - 4 \cdot 10^{-7}\right) \\ &\geq \frac{3}{50e}. \end{aligned}$$

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 x_i is at least $n/4$ and the probability that the algorithm chooses an insertion is $1/3$. Since the location of the newly inserted literal is unimportant we obtain

$$E(\Delta(t) \mid D_1) P(D_1) \geq \frac{10}{12e}.$$

For the expected drift in the case D_2 holds, we apply again the bound given by Lemma 4.4, which yields a constant drift analogously to Case 1

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

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

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

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

$$\frac{1}{5n} \leq q \leq \frac{2}{n}.$$

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 A_i be the event that the algorithm adds x_i somewhere in t. For every j in $R(t)$ let $R_j(t)$ be the event, that the algorithm deletes j. Finally, let A' be the event, that one of the A_i holds, and R' the event that one of the $R_j(t)$ holds.

Conditional on D_1 we observe for every event A_i a drift of 10. For each event $R_j(t)$ conditional on D_1 we observe a drift of 1, since the amount of redundant leaves decreases by exactly 1. Hence,

$$\begin{aligned} E(\Delta(t) \mid A_i, D_1) &= 10, \\ E(\Delta(t) \mid R_j(t), D_1) &= 1. \end{aligned}$$

Regarding the probability for these events, we observe that for 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

$$P(A_i \mid D_1) \geq \frac{1}{6n}.$$

Regarding the probability of $R_j(t)$, with probability at least $1/(5n)$ the algorithm chooses the leaf j and with probability $1/3$ the algorithm deletes j. This yields

$$P(R_j(t) \mid D_1) \geq \frac{1}{15n}.$$

In order to sum the events in A' and 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

$$|A| + |R(t)| \geq r(t)$$

holds, since $R(t)$ is the set of all redundant leaves. Furthermore, we observe that for any literal 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

$$|A| + |R(t)| \geq n - v(t). \tag{3}$$

Furthermore, due to (3) we deduce

$$\begin{aligned} s(t) - v(t) &\leq r(t) + c^+(t) + c^-(t) - v(t) \\ &\leq 4r(t) \leq 4(|A| + |R(t)|). \end{aligned} \tag{4}$$

This inequality (4) in conjunction with (3) yields

$$(10+4)(|A| + |R(t)|) \geq 10(n - v(t)) + s(t) - v(t) = g(t).$$

We obtain the expected drift conditional on the event D_1 as

$$\begin{aligned} &E(\Delta(t) \mid D_1) \\ &\geq E(\Delta(t) \mid (A' \vee R'), D_1) P(A' \vee R' \mid D_1) \\ &= \sum_{i \in A} E(\Delta(t) \mid A_i, D_1) P(A_i, D_1) \\ &\quad + \sum_{j \in R(t)} E(\Delta(t) \mid R_j(t), D_1) P(R_j(t) \mid D_1) \\ &\geq |A| \frac{10}{6n} + |R(t)| \frac{1}{15n} \geq (|A| + |R(t)|) \frac{1}{15n} \\ &\geq \frac{g(t)}{15(m+4)n}. \end{aligned}$$

Concerning the expected drift conditional on D_2, the condition for the second bound given by Lemma 4.4 is satisfied in this case. Summarizing the events D_1 and D_2 we obtain the expected drift

$$\begin{aligned} E(\Delta(t)) &\geq E(\Delta(t) \mid D_1) P(D_1) + E(\Delta(t) \mid D_2) P(D_2) \\ &\geq \frac{g(t)}{en}\left(\frac{1}{210} - \left(1 - \frac{1}{e}\right)\frac{7}{10}\sum_{i=11}^{\infty} i(i-10)\frac{1}{(i-1)!}\right) \\ &= \frac{g(t)}{en}\left(\frac{1}{210} - \left(1 - \frac{1}{e}\right)\frac{7}{10}\left(\sum_{i=1}^{10} \frac{i(10-i)}{(i-1)!} - 15e\right)\right) \\ &> \frac{2g(t)}{500en}. \end{aligned}$$

Summarizing the derived expected drifts, we observe a multiplicative drift in the case of

$$\frac{2g(t)}{500en} \leq \frac{3}{50e},$$

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 3.2) with $h(x) = \min\{3/(50e), 2x/(500en)\}$, $X_0 = T_{\text{init}} + 10n$ and $x_{min} = 1$, which yields

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

This establishes the theorem.

5 RESULTS MAJORITY

In this section we show the following theorem.

Theorem 5.1. *The (1+1) GP* without bloat control *on MAJORITY takes* $\Omega(T_{\text{init}} \log T_{\text{init}})$ *iterations in expectation for* $n = 1$. *For general* $n \geq 1$ *it takes* $\Omega(T_{\text{init}} + n \log n)$ *iterations in expectation.*

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

5.1 Lower Bound

We are going to give a rough proof for the lower bound, which will leave out in-depth analysis of standard arguments but gives non-standard details.

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 literal 1. To derive the second bound we observe, that the algorithm does in expectation 2 operations in each iteration. Hence, the algorithm needs in expectation $\Omega(T_{\text{init}})$ iterations to express the first literal yielding the desired result.

Regarding the first bound let t be a GP-tree, let $I_1(t)$ be the number of variables x_1 in t and $I_1'(t)$ be the number of variables $\overline{x}_1$ in t. We associate with t the potential function $g(t)$ by

$$g(t) = I_1'(t) - I_1(t).$$

In order to express the literal 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 $\sqrt{T_{\text{init}}}$. Let τ denote the number of iterations until the algorithm encounters for the first time a GP-tree t with $g(t) \leq \sqrt{T_{\text{init}}}$. 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 the literal 1 is expressed.

Let A be the event that in T_{init}^2 iterations the algorithm performs at least once more than $15\ln(T_{\text{init}})$. With high probability the algorithm will not encounter the event A. This yields

$$E(\tau) = E(\tau \mid A)P(A) + E(\tau \mid \overline{A})P(\overline{A}) \geq E(\tau \mid \overline{A})\frac{1}{2}.$$

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

Let t' be the random variable describing the best-so-far solution in the iteration after t. We are going to bound the expected drift $g(t) - g(t')$, which we denote by $\Delta(t)$. Let $g(t) = k - j$, where k is the number of variables $\overline{x}_i$ and j is the number of variables x_i. We observe that the variables *introduced* by an insertion or substitution only yield a drift of 0.

Let B be the event, that the algorithm chooses at least once a variable x_i for a substitution or deletion in this iteration. The probability of B is at least the probability for the algorithm to do exactly one operation: a deletion or substitution of a variable x_i. Let $s(t)$ be the amount of leaves of t. We deduce

$$P(B) \geq \frac{1}{e} \cdot \frac{2}{3} \cdot \frac{j}{s(t)}.$$

Furthermore, the expected negative drift of $g(t)$ can be bounded by this event B, which yields

$$E(\Delta \mid B) = -1.$$

Regarding the positive drift, let C_i be the event, that in this iteration the algorithm chooses to do i operations, which are either substitutions or deletions of variables $\overline{x}_i$. We observe that each deletion of a variable $\overline{x}_i$ reduces $s(t)$ and $I_1'(t)$ by 1. Each substitution of a variable $\overline{x}_i$ reduces only $s(t)$ by 1. Therefore, we can bound the probability for a substitution by at most the probability of a deletion. Let p_i be the probability, that a Pois(1) distributed random variable is equal to i. This yields for $k < s(t)$

$$P(C_i) \leq \frac{2}{3^i} \cdot p_{i-1} \cdot \frac{k!(s(t)-i)!}{s(t)!(k-i)!} \leq \frac{2}{3^i} \cdot p_{i-1} \cdot \frac{k}{2s(t)}.$$

Hence, we obtain the expected drift for $\overline{B}$

$$E(\Delta(t) \mid \overline{B})P(\overline{B}) \leq \frac{k}{es(t)} \sum_{i=1}^{\infty} \frac{i}{3^i(i-1)!} = \frac{4k}{9e^{2/3}s(t)}.$$

Summarizing, we obtain by the law of total expectation

$$E(\Delta(t)) \leq \frac{4k}{9e^{2/3}s(t)} - \frac{2j}{3es(t)} \leq \frac{2g(t)}{3es(t)}$$

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 $15\ln(T_{\text{init}}) \leq \sqrt{T_{\text{init}}}$ we can apply Theorem 3.3 and derive

$$\begin{aligned} E(\tau \mid \overline{A}, X_0 = T_{\text{init}}) &\geq \frac{1 + \ln(T_{\text{init}}) - \ln(T_{\text{init}}^{1/2})}{\frac{2}{3eT_{\text{init}}} + \frac{T_{\text{init}}}{T_{\text{init}}^2 - T_{\text{init}}}} \\ &\geq \frac{1.5e\,T_{\text{init}}\ln(T_{\text{init}})}{2 + 6e}. \end{aligned}$$

Therefore, we obtain the desired result

$$E(\tau) \geq \frac{1.5e\,T_{\text{init}}\ln(T_{\text{init}})}{4 + 12e}.$$

5.2 Upper Bound

Since the proof is long and involved, we give a very rough sketch of the proof. The key ingredient is a bound on the bloat, i.e., on the speed with which the tree grows. More precisely, we will show that if $T_{\text{init}} \geq n\log^2 n$, then in $O(T_{\text{init}}\log T_{\text{init}})$ rounds, the size of the tree grows at most by a constant factor. 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 relabelling a $\overline{x}_i$ is larger than deleting or relabelling a x_i, while they have the same probability to be inserted. Computing precisely, denoting t_r the GP-tree in round r, we get a drift

$$\mathbb{E}[V'(t_{r+1}, i) - V'(t_r, i) \mid V(t_r, i) = v] \leq -\frac{v}{3eT_{\max}} \tag{5}$$

for the $V'(t_r, i)$, where $T_{\max} = O(T_{\text{init}})$ is the maximal length of the string. Using a multiplicative drift theorem [DG13], 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, with high probability 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.

To control the bloat, 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 longer offsprings are more likely to be accepted, or shorter offsprings are more likely to be rejected. This results in a positive drift for the length, which we need to bound. Note that the biased drift comes purely from the selection process. We will show

that offsprings are rarely rejected, and bound the drift of $|t_r|$ by (essentially) the probability that the offspring is rejected.

For an index $i \in [n]$, we say that i is *touched* by some mutation, if the mutation inserts, delete or changes a x_i or $\overline{x}_i$ variable, or if it changes a variable into x_i or $\overline{x}_i$. We call a round an *i-round* if at least one of the mutations in this round touches i. First we show some elementary properties of rounds that touch i.

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, 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) $\mathbb{E}[k \mid i \text{ touched}] \leq C$.

Similar as 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 x_i such that i is touched by at least $V(t_r, i) + 1$ mutations.[1] So we want to show that this does not happen often. The probability to touch i at least k times falls geometrically in k by Lemma 5.3, so in the following we will restrict to the most dominant case $V(t_r, i) = 0$.

Similar as before, we may bound the drift of $V(t_r, i)$ in rounds that touch i by

$$\mathbb{E}[V(t_{r+1}, i) - V(t_r, i) \mid V(t_r, i) = v, r \text{ is } i\text{-round}] \geq -\frac{Cvn}{T_{\text{init}}} \quad (6)$$

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

Equation (6) tells us that the drift is negative, but relatively weak. We prove that under such circumstances, the expected return time to 0 is large. More precisely, with martingale theory we show the following lemma.

Lemma 5.4. *Consider a sequence of random variables $(X_t)_{t \geq 0}$ taking values in the non-negative integers. Let $\mathcal{F}_{t,x}$ be the set of all filtrations $\mathcal{F} = (X_0, \dots, X_t)$ with $X_t = x$ of this process. Assume there are $\delta, C, N > 0$ such that the following four conditions hold for all $t \geq 0$.*

(i) Weak Drift. $\mathbb{E}[X_{t+1} - X_t \mid \mathcal{F}] \geq -Cx/N$ *for all $x \geq 1$ and all $\mathcal{F} \in \mathcal{F}_{t,x}$;*
(ii) Small Steps. $\Pr[|X_{t+1} - X_t| \geq k \mid \mathcal{F}] \leq (1+\delta)^{-k}$ *for all $\mathcal{F} \in \bigcup_{x \geq 1} \mathcal{F}_{t,x}$;*
(iii) Initial Increase. $\Pr[X_{t+1} > X_t \mid \mathcal{F}] \geq \delta$ *for all $\mathcal{F} \in \bigcup_{x \leq \sqrt{N}} \mathcal{F}_{t,x}$.*

Then there is $\varepsilon = \varepsilon(\delta, C) > 0$ which does not depend on N such that the following holds for every $1 \leq x \leq \frac{1}{2}\sqrt[4]{N}$, and every $x' \geq x$. If $T := \min\{\tau \geq 0 \mid X_t < x\}$ is the hitting time of $\{0, 1, \dots, x-1\}$ then

$$\mathbb{E}[T \mid X_0 = x'] \geq \varepsilon\sqrt{N}.$$

By (6), we may apply Lemma 5.4 with $N := T_{\text{init}}/n$ and obtain a return time of $\Omega(\sqrt{T_{\text{init}}/n})$. In other words, after $V(t_r, i)$ becomes positive for the first time, it needs in expectation $\Omega(\sqrt{T_{\text{init}}/n})$ i-rounds to return to 0. On the other hand, it only needs $O(1)$ i-rounds to leave 0 again. Hence, it is only in 0 in a $O(\sqrt{n/T_{\text{init}}})$-fraction of all i-rounds. Thus the drift of $|t_r|$ is also $O(\sqrt{n/T_{\text{init}}})$.

In particular, if $T_{\text{init}} \geq n \log^2 n$ then in $r_0 = O(T_{\text{init}} \log T_{\text{init}})$ rounds the drift increases the size of the GP-tree in expectation by at most $r_0\sqrt{n/T_{\text{init}}} = O(T_{\text{init}})$. Hence we expect the size to grow by at most a constant factor. With Markov's inequality, after working out the details, the exact (rather technical) statement is the following.

Theorem 5.5. *There is $\varepsilon > 0$ such that the following holds. Let $f = f(n) = \omega(1)$ be any growing function. Assume $T_{\text{init}} \geq f(n) \cdot n \log^2 n$. Then with probability at least $1 - 1/(2\sqrt{f(n)})$, within the next $r_0 := \varepsilon f(n) T_{\text{init}} \log T_{\text{init}}$ rounds the tree has never more than $T_{\max} := \frac{1}{4}\sqrt{f(n)} T_{\text{init}}$ leaves.*

Note that if $f(n)$ grows very slowly, Theorem 5.5 says roughly that after $\approx T_{\text{init}} \log T_{\text{init}}$ rounds the size of the tree is still $\approx T_{\text{init}}$. It is still technical to derive the upper bound in Theorem 5.2 from Theorem 5.5, but it is possible with the ideas sketched at the beginning of this section. This concludes the proof outline.

6 CONCLUSION

We considered a simple mutational genetic programming algorithm, the (1+1) GP. We saw that, for the two simple problems ORDER and MAJORITY, 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 saw a very efficient optimization behavior: whenever there is a significant number of redundant leaves, these leaves are being pruned. Whenever only few redundant leaves are present, the algorithm easily increases the fitness of the GP-tree.

For optimization without consideration of the size of the GP-trees, 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 multiplicative polylogarithmic factors. While such factors are not a big problem 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. For actually getting small solutions, some kind bloat control should be used.

From our work we see an interesting option for bloat control: make deletions more likely than insertions. In the drift equations in this paper we would see a bias towards shorter solutions, overall leading to faster optimization.

REFERENCES

[DG13] Benjamin Doerr and Leslie Ann Goldberg. Adaptive drift analysis. *Algorithmica*, 65(1):224–250, 2013.
[DNO11] Greg Durrett, Frank Neumann, and Una-May O'Reilly. Computational complexity analysis of simple genetic programming on two problems modeling isolated program semantics. In *Proc. of FOGA'11*, pages 69–80, 2011.
[GO98] David E. Goldberg and 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*, pages 16–36, 1998.
[HY04] Jun He and Xin Yao. A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, 3(1):21–35, 2004.
[Joh10] Daniel Johannsen. *Random Combinatorial Structures and Randomized Search Heuristics*. PhD thesis, Universität des Saarlandes, 2010. Available

[1] This statement is not literally true, since it neglects some borders cases.

online at http://scidok.sulb.uni-saarland.de/volltexte/2011/3529/pdf/Dissertation_3166_Joha_Dani_2010.pdf.

[KNS11] Timo Kötzing, Frank Neumann, and Reto Spöhel. PAC learning and genetic programming. In *Proc. of GECCO'11*, pages 2091–2096, 2011.

[KSNO12] Timo Kötzing, Andrew M. Sutton, Frank Neumann, and Una-May O'Reilly. The max problem revisited: the importance of mutation in genetic programming. In *Proc. of GECCO'12*, pages 1333–1340, 2012.

[MM14] Andrea Mambrini and Luca Manzoni. A comparison between geometric semantic GP and cartesian GP for boolean functions learning. In *Proc. of GECCO'14*, pages 143–144, 2014.

[MMM13] Alberto Moraglio, Andrea Mambrini, and Luca Manzoni. Runtime analysis of mutation-based geometric semantic genetic programming on boolean functions. In *Proc. of FOGA'13*, pages 119–132, 2013.

[MO16] Andrea Mambrini and Pietro Simone Oliveto. On the analysis of simple genetic programming for evolving boolean functions. In *Proc. of EuroGP'16*, pages 99–114, 2016.

[Neu12] Frank Neumann. Computational complexity analysis of multi-objective genetic programming. In *Proc. of GECCO'12*, pages 799–806, 2012.

[NUW13] Anh Nguyen, Tommaso Urli, and Markus Wagner. Improved computational complexity results for weighted ORDER and MAJORITY. In *Proc. of FOGA'13*, 2013. To appear.

[RS12] Jonathan E. Rowe and Dirk Sudholt. The choice of the offspring population size in the $(1, \lambda)$ ea. In *Proc. of GECCO'12*, pages 1349–1356, 2012.

7 APPENDIX

7.1 Proof Multiplicative Drift, Bounded Step Size

We are going to map the multiplicative bound of the expected drift to a function, which will enable us to apply the additive drift theorem. Let

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

and $g(0) := 0$. Furthermore, let $Z_t := g(X_t)$ denote the associated stochastic process of X_t over the finite search space $R = g(S) \cup \{0\}$. We observe, that T is also the first point in time $t \in \mathbb{N}$ such that $Z_t \leq 1$. 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 $Z_{t+1} > 0$. We observe

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

Therefore, due to Jensen's inequality we obtain

$$E(Z_t - Z_{t+1} \mid X_t = s) \leq \ln\left(E\left(\frac{X_t}{X_{t+1}} \middle| X_t = s\right)\right).$$

Due to the bounded stepsize, the value of X^{t+1} can only be in a κ-interval around X_t. For all $i \geq 0$ let x_i be the probability that $X_t - X_{t+1} = i$ and let y_i be the probability that $X_t - X_{t+1} = -i$. We note, that $x_0 = y_0$ and obtain by counting twice the instance of a step size of 0

$$\begin{aligned} E\left(\frac{X_t}{X_{t+1}} \middle| X_t = s\right) &\leq \left(\sum_{i=0}^{\kappa} \frac{s}{s-i} x_i + \frac{s}{s+i} y_i\right) \\ &= \left(\sum_{i=0}^{\kappa} s \frac{x_i(s+i) + y_i(s-i)}{s^2 - i^2}\right) \\ &\leq \left(\sum_{i=0}^{\kappa} s \frac{x_i(s+i) + y_i(s-i)}{s^2 - \kappa^2}\right) \\ &= \left(\frac{s^2}{s^2-\kappa^2} + \sum_{i=0}^{\kappa} \frac{s(ix_i - iy_i)}{s^2-\kappa^2}\right), \end{aligned}$$

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

$$\begin{aligned} E\left(\frac{X_t}{X_{t+1}} \middle| X_t = s\right) &\leq \left(\frac{s^2}{s^2-\kappa^2} + \frac{2}{s}\sum_{i=0}^{\kappa} ix_i - iy_i\right) \\ &= \left(1 + \frac{\kappa^2}{s^2-\kappa^2} + \frac{2}{s}\sum_{i=0}^{\kappa} ix_i - iy_i\right). \end{aligned}$$

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

$$\begin{aligned} E(Z_t - Z_{t+1} \mid X_t = s) &\leq \ln\left(1 + \frac{\kappa^2}{X_t^2 - \kappa^2} \right. \\ &\quad \left. + 2E\left(\frac{X_t - X_{t+1}}{X_t} \middle| X_t = s\right)\right) \\ &\leq 2E\left(\frac{X_t - X_{t+1}}{X_t} \middle| X_t = s\right) + \frac{\kappa^2}{X_t^2 - \kappa^2} \\ &\leq 2\delta + \frac{\kappa^2}{X_t^2 - \kappa^2}. \end{aligned}$$

Finally, we apply the additive drift theorem and obtain

$$\begin{aligned} E(T \mid X_0 = s_0) = E(T \mid Z_0 = g(s_0)) &\geq \frac{g(s_0)}{2\delta + \frac{\kappa^2}{s_{\min}^2 - \kappa^2}} \\ &= \frac{1 + \ln(s_0) - \ln(s_{\min})}{2\delta + \frac{\kappa^2}{s_{\min}^2 - \kappa^2}}. \end{aligned}$$

7.2 Bloat Control Upper Bound

Proof Lemma 4.2. Let $opt(t)$ be the number of optimal leaves, i.e. positive leaves x_i, where no additional instances of the literal 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 we have to bound the number of non-optimal critical positive leaves.

For MAJORITY a literal 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 literal 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 variable x_i is a redundant leaf. Therefore, we deduce $c^-(t) - opt(t) \leq r(t)$. □

Proof Lemma 4.3. We split the proof in two parts, one for ORDER and one for MAJORITY.

For MAJORITY a literal i can only contribute a critical negative leaf if the number of positive variables x_i is m and the number of negative variables $\overline{x}_i$ 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 literal 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 $c^+(t) \leq r(t)$. □

7.3 MAJORITY Lower Bound

7.4 Lower Bound

Regarding the proof of the lower bound, 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 literal 1. To derive the second bound we observe, that the algorithm does in expectation 2 operations in each iteration since $E(1 + \text{Pois}(1)) = 2$. Hence, the algorithm needs in expectation at least $T_{\text{init}}/2$ iterations to express the first literal yielding the desired result.

Regarding the first bound let t be a GP-tree, let $I_1(t)$ be the number of variables x_1 in t and $I_1'(t)$ be the number of variables $\overline{x_1}$ in t. We associate with t the potential function $g(t)$ by

$$g(t) = I_1'(t) - I_1(t).$$

In order to express the literal 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 $g(t) \leq T_{\text{init}}^{2/3}$. 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 the literal 1 is expressed.

Let 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 A_i we obtain due to the Poisson-distributed number of operations

$$P(A_i) = \sum_{i=z}^{\infty} \frac{1}{e(i-1)!}.$$

Let p_i be the probability, that a Pois(1) distributed random variable is equal to i. We derive

$$p_{i+1} = p_i \frac{1}{i+1} \leq p_i \frac{1}{2}.$$

Since A_i is Pois(1) distributed, this yields

$$P(A_i) \leq p_z \sum_{i=0}^{\infty} \frac{1}{2^i} = \frac{2}{ez!}.$$

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

$$P(A_i) \leq \frac{e^z}{ez^z} \leq \frac{T_{\text{init}}^{15}}{z^z} \leq T_{\text{init}}^{-15},$$

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

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

$$P(A) = P(\bigcup_{i=1}^{T_{\text{init}}^2} A_i) \leq \sum_{i=1}^{T_{\text{init}}^2} P(A_i) \leq T_{\text{init}}^{-13}.$$

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

$$E(\tau) = E(\tau \mid A)P(A) + E(\tau \mid \overline{A})P(\overline{A}) \geq E(\tau \mid \overline{A})\frac{1}{2}.$$

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

Let t' be the random variable describing the best-so-far solution in the iteration after t. We are going to bound the expected drift $g(t) - g(t')$, which we denote by $\Delta(t)$. Let $g(t) = k - j$, where k is the number of variables $\overline{x}_i$ and j is the number of variables x_i. If the algorithm chooses an insertion, the probability to insert x_i is the same as the probability to insert $\overline{x}_i$. Therefore, an insertion will only contribute a 0 to the expected drift. The same holds for the variables *introduced* by a substitution. However, for variables *deleted* by a deletion or substitution the probability to choose a variable x_i or $\overline{x}_i$ is of importance contrary to an insertion.

Let B be the event, that the algorithm chooses at least once a variable x_i for a substitution or deletion in this iteration. The probability of B is at least the probability for the algorithm to do exactly one operation: a deletion or substitution of a variable x_i. Let $s(t)$ be the amount of leaves of t. We deduce

$$P(B) \geq \frac{1}{e} \cdot \frac{2}{3} \cdot \frac{j}{s(t)}.$$

Furthermore, the expected negative drift of $g(t)$ can be bounded by this event B, which yields

$$E(\Delta \mid B) = -1.$$

Regarding the positive drift, let C_i be the event, that in this iteration the algorithm chooses to do i operations, which are either substitutions or deletions of variables $\overline{x}_i$. Again, the algorithm chooses with probability 1/3 to do either a substitution. Additionally, the algorithm chooses to do i operations with probability p_{i-1} with p_i as defined above. However, the probability to choose a variable $\overline{x}_i$ changes with each operation. Each deletion of a variable $\overline{x}_i$ reduces $s(t)$ and I_1' by 1. Each substitution of a variable $\overline{x}_i$ reduces $s(t)$ by 1 and I_1' stays the same. Therefore, we can bound the probability for a substitution by at most the probability of a deletion. This yields for $k < s(t)$

$$P(C_i) \leq \frac{2}{3^i} \cdot p_{i-1} \cdot \frac{k!(s(t)-i)!}{s(t)!(k-i)!} \leq \frac{2}{3^i} \cdot p_{i-1} \cdot \frac{k}{2s(t)}.$$

Hence, we obtain the expected drift for $\overline{B}$

$$E(\Delta(t) \mid \overline{B})P(\overline{B}) \leq \frac{k}{es(t)} \sum_{i=1}^{\infty} \frac{i}{3^i(i-1)!} = \frac{4k}{9e^{2/3}s(t)}.$$

Summarizing, we obtain by the law of total expectation

$$E(\Delta(t)) \leq \frac{4k}{9e^{2/3}s(t)} - \frac{2j}{3es(t)} \leq \frac{2g(t)}{3es(t)}$$

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 3.3. and derive

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

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

$$E(\tau) \geq \frac{3e\, T_{\text{init}} \ln(T_{\text{init}})}{8 + 12e},$$

which yields the desired result.

7.5 MAJORITY Upper Bound

7.5.1 Preparations. We now turn to the formal proof of the upper bound in Theorem 5.2.

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

Our first lemma states that it is exponentially unlikely to have many mutations, even if we condition on some variable to be touched.

Lemma 7.1. *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, 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

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

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

$$\Pr[\text{Pois}(1) \geq \kappa - 1] \leq \left(\frac{e}{\kappa - 1}\right)^{\kappa - 1}. \tag{7}$$

1. First consider $\kappa \geq 4$. Then, using $\kappa - 1 \geq \kappa/2$ we get from (7):

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

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 $\kappa \in \{2, 3\}$. This proves 1.

2. and 3. Let $T = |t|$ be the number of leaves in t, and assume $T \geq 2n$. Moreover, we define the parameter

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

Then we claim

$$\Pr[k = 1 \text{ and } i \text{ touched}] \geq \frac{x}{3e}. \tag{8}$$

To see the claim, first note that $\Pr[k = 1] = 1/e$ by definition of the Poisson distribution. Consider first 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 (8). 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 (8). This proves the claim.

We first give the proof in the much simpler case of large x. For concreteness, 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,

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

This already implies 2., since

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

For 3., it suffices to observe that

$$\begin{aligned} \Pr[k \geq \kappa \mid i \text{ touched}] &= \frac{\Pr[k \geq \kappa \text{ and } i \text{ touched}]}{\Pr[i \text{ touched}]} \\ &\leq \frac{\Pr[k \geq \kappa]}{\Pr[k = 1 \text{ and } i \text{ touched}]} \\ &\overset{1.}{\leq} 12e \cdot e^{-\delta\kappa}, \end{aligned} \tag{9}$$

which easily implies 3. by absorbing the factor $12e$ into the exponential.

The case for smaller x will basically run along the same lines, but will be much more involved. In particular, in (9) we cannot use the trivial bounds in the second line. So assume from now on $x < 1/4$, so at most one forth 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 call the mutations $M_1, \ldots, M_k$, 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 $1 \leq \kappa \leq k$,

$$\Pr[\kappa_i \geq \kappa + 1 \mid k, \kappa_i \geq \kappa] \geq 1 - 3x \geq e^{-6x}, \tag{10}$$

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

To see the the first inequality of (10), 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 relabelling is at most $\frac{2}{3}(T/n)/(T/2) < 2/n$. On the other hand, the probability to insert an i-literal, or to relabel 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 (10) if $x = 1/n$.

The other case is very similar, only with 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 relabelling is at most $\frac{2}{3}xT/(T/2) \leq 2x$. On the other hand, the probability to insert an i-literal, or to relabel 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 (10) if $x = \#\{i\text{-literals}\}/T$.

We can expand (10) to obtain the probability of $\kappa_i = \infty$. For $2 \le k \le 1/x$,

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

and consequently

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

For $k > 1/x$ we will use the trivial 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 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 in the interval $(0, 1]$. Together with (8), 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 may 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,

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

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., since

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

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

For a GP-tree t, let

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

In particular, x_i is expressed if and only if $V(t, i) \ge 0$. 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 x_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 $|V(t, i)| \le s$. We say that an index i is *critical* in a round if $V(t, i) \ge 0$, and i is touched at least $V(t, i)$ times in this round; we call the index *non-critical* otherwise. Note that in a non-critical round, the fitness of the GP-tree cannot decrease. We say that a round is critical if there is at least one critical index in this round.

We end this section with a lemma which 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.

Lemma 7.2. *Consider a sequence of random variables $(X_t)_{t \ge 0}$ taking values in the non-negative integers. Let $\mathcal{F}_{t,x}$ be the set of all filtrations $\mathcal{F} = (X_0, \dots, X_t)$ with $X_t = x$ of this process. Assume there are $\delta, C, N > 0$ such that the following four conditions hold for all $t \ge 0$.*

- *(i)* Weak Drift. $\mathbb{E}[X_{t+1} - X_t \mid \mathcal{F}] \ge -Cx/N$ *for all* $x \ge 1$ *and all* $\mathcal{F} \in \mathcal{F}_{t,x}$;
- *(ii)* Small Steps. $\Pr[|X_{t+1} - X_t| \ge k \mid \mathcal{F}] \le (1 + \delta)^{-k}$ *for all* $\mathcal{F} \in \bigcup_{x \ge 1} \mathcal{F}_{t,x}$;
- *(iii)* Initial Increase. $\Pr[X_{t+1} > X_t \mid \mathcal{F}] \ge \delta$ *for all* $\mathcal{F} \in \bigcup_{x \le \sqrt{N}} \mathcal{F}_{t,x}$.

Then there is $\varepsilon = \varepsilon(\delta, C) > 0$ which does not depend on N such that the following holds for every $1 \le x \le \frac{1}{2}\sqrt[4]{N}$, and every $x' \ge x$. If $T := \min\{\tau \ge 0 \mid X_t < x\}$ is the hitting time of $\{0, 1, \dots, x - 1\}$ then

$$\mathbb{E}[T \mid X_0 = x'] \ge \varepsilon\sqrt{N}.$$

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

Without loss of generality, we may assume that, stronger than (i), we have $|\mathbb{E}[X_{t+1} - X_t \mid \mathcal{F}]| \le Cx/N$ for all $x \ge 1$ and all $\mathcal{F} \in \mathcal{F}_{t,x}$. If this does not hold for some t and x, we may just alter the process slightly by, in addition, with some probability adding an O(1) step to the left. It is clear that the modified process is dominated by the original one, and hence rather has a smaller time to reach $\{0, \dots, x - 1\}$.

We first consider the drift of X_t^2. Let $p_i := \Pr[X_{t+1} - X_t = i \mid \mathcal{F}_t]$ for all $i \in \mathbb{Z}$. Then

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

Consequently, by our assumption on the drift of X_t, (ii) and (iii), we have

$$-\frac{2CX_t^2}{N} + \delta \le \mathbb{E}[X_{t+1}^2 - X_t^2 \mid \mathcal{F}_t] \le \frac{2CX_t^2}{N} + O(1). \tag{11}$$

Let t_0 be the (random) time when the process X_t (started at $X_0 = x'$) for the first time leaves the interval $I = [x..\sqrt{N}]$ on either side. Note that, trivially, $t_0 \le T$. Let p_ℓ and p_r be the probabilities that the process leaves the interval on the left (that is, at $x-1$ or lower) and on the right (that is, at $\lfloor\sqrt{N}\rfloor + 1$ or higher). By (ii), it is clear that if the process leaves I on the right side, then the expected point it lands on is at most $2\sqrt{N}$; recall that we assumed N to be large.

By (11) there is a constant $D > 0$ such that the process $Y_t = X_t^2 - Dt$ has a negative drift in the interval I. Hence, and using that t_0 is a stopping time, we obtain

$$X_0^2 = \mathbb{E}[Y_0] \ge \mathbb{E}[Y_{t_0}] \ge p_r N - D\mathbb{E}[t_0]. \tag{12}$$

Similarly, regarding the process $Z_t = X^t + Ct/\sqrt{N}$ and noting that this has positive drift, we obtain

$$X_0 = \mathbb{E}[Z_0] \le E[Z_{t_0}] \le p_\ell \cdot (x-1) + 2p_r\sqrt{N} + \frac{C\mathbb{E}[t_0]}{\sqrt{N}}. \tag{13}$$

This gives a lower bound of $p_r \ge (X_0 - C\mathbb{E}[T_0]/\sqrt{N} - p_\ell(x-1))/(2\sqrt{N})$ for p_r. This lower bound together with (12) yields our desired bound

$$\mathbb{E}[t_0] \ge \frac{\frac{1}{2}X_0\sqrt{N} - \frac{1}{2}p_\ell(x-1) - X_0^2}{D + \frac{1}{2}C} \ge \frac{\frac{1}{2}\sqrt{N} - X_0^2}{D + \frac{1}{2}C}, \tag{14}$$

which is of order $\Omega(\sqrt{N})$ if $X_0 \le \frac{1}{2}\sqrt[4]{N}$.

□

7.5.2 Bloat Estimation. The main part of the proof is to study how the size of the tree increases. We show that it increases by little more than a constant factor within roughly $T_{\text{init}} \log T_{\text{init}}$ rounds if $T_{\text{init}} = \omega(n \log^2 n)$.

Theorem 7.3. *There is $\varepsilon > 0$ such that the following holds. Let $f = f(n) = \omega(1)$ be any growing function. Assume $T_{\text{init}} \ge f(n) \cdot n \log^2 n$. Then with probability at least $1 - 1/(2\sqrt{f(n)})$, within the next $r_0 := \varepsilon f(n) T_{\text{init}} \log T_{\text{init}}$ rounds the tree has never more than $T_{\max} := \frac{1}{4}\sqrt{f(n)} T_{\text{init}}$ leaves.*

The proof of Theorem 7.3 is complicated, and the whole remaining section is devoted to it. First we give an outline of the basic ideas. We will couple the size (i.e., the number of leaves) of the GP tree to a different process $S = (S_r)_{r\ge 0}$ on $\mathbb{N}$ which is easier to analyse. 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 $S_r = \sum_{j=1}^r (X'_j + X_j)$, where X'_j are independent random variables without drift, and X_j are only non-zero in critical rounds. The most difficult part is then 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 as described in Lemma 7.2, with parameter $T \approx T_{\text{init}}/n \ge \log^2 n$. For this outline, consider only rounds in which an index $i \in [n]$ 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 $1 + \text{Pois}(1)$-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 $\Omega(\sqrt{T})$ i-rounds to return to zero. Intuitively, this means that in a random i-round, the probability to encounter $V(t,i) = 0$ is $O(1/\sqrt{T}) = O(1/\log n)$. Since each round touches only $O(1)$ indices, and each of them has only probability $O(1/\log n)$ to be critical, there are only $O(T_{\text{init}})$ critical rounds within $T_{\text{init}} \log T_{\text{init}} \approx T_{\text{init}} \log n$ rounds. Thus the size of the GP-tree grows only roughly by a constant factor in $T_{\text{init}} \log T_{\text{init}}$ rounds.

of Theorem 7.3. 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 mutation. Then we set $X'_{j+1} := |t'| - |t|$, and

$$X_{j+1} = \begin{cases} k, & \text{there is a positive critical index in round } j; \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Note that 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 the first case of (15). In particular, in the second case of (15) we have $f(t') \ge f(t)$, and hence the GP-tree t' is accepted. Thus, in this case we have $S_{j+1} - S_j = X'_{j+1} + X_{j+1} = |t'| - |t|$, so the change is identical for the both processes. For the first case, we have $S_{j+1} - S_j = k + |t'| - |t| \ge \max\{0, |t'| - |t|\}$. Since the size of the GP-tree changes either by $|t'| - |t|$ (if t' is accepted) or by 0 (if t' is rejected), the increase of S is at least the increase of the size of the GP-tree. Since this is true for all cases, S dominates the size of the trees, and it suffices to give upper bounds on S. To be precise, in the following we give bounds on S under the assumption that the size of the GP-tree never falls below T_{init}. 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_{init} in some round $r \le r_0$. In the former case we are done, in the latter case we apply the same argument again starting in round r.[2] Moreover, when we compute the change in round r, we may assume that the size of t_r before this round is always at most $T_{\max}$.

So assume that the size of the GP-tree is always at least T_{init}. To bound $S_r = \sum_{j=1}^r (X_j + X'_j)$, we will prove separately that each of the bounds $\sum_{j=1}^r X'_j \le T_{\max}/3$ and $\sum_{j=1}^r X_j \le T_{\max}/3$ holds with probability at least $1 - 1/(4\sqrt{f(n)})$. By the union bound, it will follow that *both* bounds together hold with probability at least $1 -$

[2] Note that the same argument also shows that for $T_{\text{init}} \le f(n) n \log^2 n =: T'_{\text{init}}$, Theorem 7.3 holds for T'_{init}. I.e., the size of the tree does not grow above $\frac{1}{4}f(n)T'_{\text{init}}$ within $f(n)T'_{\text{init}} \log T'_{\text{init}}$ rounds with probability at least $1 - 1/\sqrt{f(n)}$.

$1/(2\sqrt{f(n)})$. The two bounds will imply that the size of the GP-tree is at most $T_{\text{init}} + 2T_{\max}/3 \le T_{\max}$, thus proving the theorem. Recall that we need to consider the range $1 \le r \le r_0 = f(n)\varepsilon T_{\text{init}} \log T_{\text{init}}$.

Bounding 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 relabelling), where k is either constantly 1 or 1 + 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 $k \sim 1 + \text{Pois}(1)$ we have $K_r \sim r + \text{Pois}(r)$ since the sum of independent Poisson distributed random variables is again Poisson distributed. Since Pois(r) is dominated by Pois(r_0), we have

$$\Pr[K_r \ge 3r_0] \le \Pr[\text{Pois}(r_0) \ge 2r_0] \le \frac{e^{-r_0}(er_0)^{2r_0}}{(2r_0)^{2r_0}} = (e/4)^{r_0}$$

for each $r \le r_0$. Note that this estimate holds trivially also for the case that all k are one, because then the probability on the left is zero. Taking a union bound over all $1 \le r \le r_0$ we see that with exponentially high probability[3] $K_r \le 3r_0$ also holds uniformly for all $1 \le r \le r_0$. For each mutation, the probability of insertion, deletion, and relabelling 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.[4] Thus we may use the Chernoff bounds to infer that with exponentially high probability $\sum_{j=1}^{r} X'_j \le r_0^{2/3} < T_{\max}/3$ holds uniformly for all $1 \le r \le r_0$. In particular, this probability is at least $1 - 1/(4\sqrt{f(n)})$.

Bounding X_j: Setup. It remains to bound $\sum_{j=1}^{r} X_j$. Recall that X_j is either zero, or it is the number of mutations applied in the j-th round. So the sum is non-decreasing in r. Hence, it suffices to bound the sum for $r = r_0$, and the same bound will follow for all $r \le r_0$. We will bound its expectation, and then apply Markov's inequality.

Fix some $i \in [n]$, and consider the random walk of the variable $V(t_r, i)$. Assume that the GP-tree t_r as size at least $T_{\text{init}}/2$. (This assumption will be justified later). We call a round an *i-round* if i is touched in this round. 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 Lemma 7.2 with $T := T_{\text{init}}/n$ to this random walk. To this end, in the next paragraphs we prove that the random walk that $V(t_r, i)$ performs in i-rounds satisfies the conditions of Lemma 7.2.

Bounding X_j: Computing the Drift. Let us first consider $v \ge 1$, and compute the drift

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

Mind 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 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

$$\Pr[\mathcal{E}_r] = \Omega(1/n), \tag{16}$$

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

Consider any round r conditioned on $\mathcal{E}_r$. Let M be any mutation in round r. If M does not touch i, then M does not change $V(t_r, i)$, so 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, so the case that M is an insertion contributes zero to the drift. By the same argument, the cases that M relabels a non-i-literal into x_i or into $\overline{x}_i$ 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 $\overline{x}_i$. Assume that the number of x_i is $x + v$, while the number of $\overline{x}_i$ is 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 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 $v/|t_r| \le 2v/T_{\text{init}}$, where $|t_r| \ge T_{\text{init}}$ is the current size of the GP-tree. Thus the conditional probability (on $\mathcal{E}_r$) to pick one of them is at most $O(vn/T_{\text{init}})$ by (16). Since the conditional expected number of deletions is $\mathbb{E}[\#\text{ deletions} \mid \mathcal{E}_r] = O(1)$, the deletions contribute $-O(vn/T_{\text{init}})$ to the drift $\Delta_{v,i}$. By the same argument we also get a contribution of $-O(vn/T_{\text{init}})$ for relabelings of x_i-literals or $\overline{x}_i$-literals.

Summarizing, the only cases contributing to $\Delta'_{v,i}$ are deletions and relabelings of i-literals, which contribute not less than $-O(vn/T_{\text{init}})$. 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 Lemma 7.2 with $T = T_{\text{init}}/n$.

Bounding X_j: Step Sizes and Initial Increase. The second condition (small steps) of Lemma 7.2 follows from Lemma 7.1. Finally, for the third condition (initial increase) we show that for every $v \le \sqrt{T}$, $T := \sqrt{T_{\text{init}}/n}$ 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 7.1, an i-round has probability $\Omega(1)$ to have exactly one mutation. Now we distinguish two cases: if there are less than $|t_r|/n$ 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 $|t_r|/n \ge T_{\text{init}}/n \ge \frac{1}{2}\log^2 T_{\text{init}}$ occurrences of i-literals, and assume that $v \le \sqrt{T_{\text{init}}/n} < \frac{1}{4}|t_r|/n$, where the last inequality holds for n large enough since $T_{\text{init}}/n \ge \log^2 n$ is then large enough. Then $\overline{x}_i$ occurs at least half as often as x_i, and thus the probability of deleting or relabelling a $\overline{x}_i$-literal is at least half as big as the probability to delete or relabel an x_i-literal. Therefore, a mutation that touches i is with

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

[4] This would not be true if the GP-tree was empty because then a deletion would fail. So here we use the assumption that the size of the GP-tree is at least T_{init}.

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 Lemma 7.2 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 Lemma 7.2. Hence, for $0 < v < \sqrt{T} = \omega(\log T_{\text{init}})$ the expected hitting time of $\{[0, 1, \dots, v]\}$ when starting at any value larger than v is $\Omega(\sqrt{T_{\text{init}}/n})$.

Now we have all ingredients to bound the expected number of positive critical rounds. Fix an index i and some $v \geq 0$. We want to bound the number of rounds in which $V(t_r, i) = v$ and i is a critical index. If $v \geq \sqrt{T} = \omega(\log T_{\text{init}})$ then with probability $1 - T_{\text{init}}^{-\omega(1)}$ this never happens for any index and any of r_0 rounds, by Lemma 7.1. In particular, the expected number of such round is negligible. So we may assume $0 \leq v < \sqrt{T}$. We count how many i-rounds occur with $V(t_r, i) = v$ before for the first time $V(t_r, i) > v$. 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 7.1. So in expectation there are only $O(1)$ such rounds before for the first time $V(t_r, i) > v$. On the other hand, once $V(t_r, i) > v$ it takes in expectation at least $\Omega(\sqrt{T_{\text{init}}/n})$ i-rounds before the interval $\{[0, 1, \dots, v]\}$ is hit again. In particular, of the subsequent i-rounds at most an expected $O(\sqrt{n/T_{\text{init}}})$-fraction will have the property $V(t_r, i) = v$. Hence, if E_i denotes the expected number of i-rounds,

$$\mathbb{E}[\#\{i\text{-rounds with } V(t_r, i) = v\}] \leq O\left(1 + \sqrt{\frac{n}{T_{\text{init}}}} E_i\right).$$

An i-round with $V(t_r, i) = v$ has probability $e^{-\Omega(v)}$ for i to be critical by Lemma 7.1. Therefore, the expected number of critical rounds within the first r_0 rounds is at most

$$\begin{aligned}
\mathbb{E}[\#\{\text{crit. rounds}\}] &\leq \sum_{\substack{i \in [n] \\ 0 \leq v \leq \sqrt{T}}} e^{-\Omega(v)} \cdot O\left(1 + \sqrt{\frac{n}{T_{\text{init}}}} E_i\right) \\
&\leq \sum_{i \in [n]} O\left(1 + \sqrt{\frac{n}{T_{\text{init}}}} E_i\right) \\
&\leq O(n) + O\left(\sqrt{\frac{n}{T_{\text{init}}}}\right) \sum_{i \in [n]} E_i
\end{aligned}$$

We bound the sum further by observing that in each round only $O(1)$ literals are touched in expectation. Therefore, $\sum_{i \in [n]} E_i = O(r_0)$. Moreover, by assumption $T_{\text{init}} \geq f(n) n \log^2 n$, which implies $T_{\text{init}} \geq \frac{1}{2} f(n) n \log^2 T_{\text{init}}$ for sufficiently large n. Hence,

$$\begin{aligned}
\mathbb{E}[\#\{\text{critical rounds}\}] &\leq O\left(n + r_0 \sqrt{\frac{n}{T_{\text{init}}}}\right) \\
&\leq O\left(n + \frac{r_0}{\sqrt{f(n)} \log T_{\text{init}}}\right) \\
&\leq \frac{1}{36} \sqrt{f(n)} T_{\text{init}},
\end{aligned}$$

where the last step follows from $r_0 = f(n) \varepsilon T_{\text{init}} \log T_{\text{init}}$ if $\varepsilon > 0$ is sufficiently small. This shows that $\mathbb{E}[\sum_{j=1}^{r_0} X_j] \leq \frac{1}{12} \sqrt{f(n)} T_{\text{init}}$. By Markov's inequality, $\sum_{j=1}^{r_0} X_j \leq T_{\max}/3 = \frac{1}{12} \sqrt{f(n)} T_{\text{init}}$ with probability at least $1 - 1/(4\sqrt{f(n)})$. This proves the desired bound on S_r and thus concludes the proof of Theorem 7.3. □

7.5.3 Runtime Bound. For technical reasons, we prove first a more technical statement that holds with high probability. We will only treat the (harder) case $T_{\text{init}} \geq f(n) n \log^2 n$.

Lemma 7.4. *Let $\varepsilon > 0$ be the constant from Theorem 7.3. There is a constant $C > 0$ such that the following holds for any growing function $f(n) = \omega(1)$. For any starting tree with $T_{\text{init}} \geq f(n) n \log^2 n$ leaves, with probability at least $1 - 1/\sqrt{f(n)}$ the (1+1) GP without bloat control on MAJORITY finds a global optimum within $r_0 := \varepsilon f(n) T_{\text{init}} \log T_{\text{init}}$ rounds, and the size of the GP-tree never exceeds $T_{\max} := \frac{1}{4} \sqrt{f(n)} T_{\text{init}}$.*

Proof. We already know by Theorem 7.3 that with probability $1 - 1/(2\sqrt{f(n)})$, the size of the GP-tree does not exceed $T_{\max}$ within r_0 rounds. Fix an index i which is not expressed at the beginning, and consider $V'(t_r, i) := \max\{-V(t_r, i), 0\}$. Note that $V'(t_r, i) > 0$ if and only if x_i is expressed. We claim that $V'(t_r, i)$ has a multiplicative drift,

$$\mathbb{E}[V'(t_{r+1}, i) - V'(t_r, i) \mid V(t_r, i) = v] \leq -\frac{v}{3eT_{\max}}. \tag{17}$$

To prove (17), 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 most zero. Analogously, relabelling a non-i-literal into an i-literal contributes at most 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 $v/|t_r| \geq v/T_{\max}$ larger than the probability to delete a x_i-literal. Since we always accept deleting a single $\overline{x}_i$-literal, this case contributes no less than $-\frac{v}{3eT_{\max}}$ to the drift. For all the other cases (several deletions, relabelling of one or several i-literals), it is always more likely to pick a $\overline{x}_i$-literal for deletion/relabelling than a x_i-literal, and it is more likely to accept the offspring if a $\overline{x}_i$-literal is deleted/relabelled. Therefore, these remaining cases contribute at most zero to the drift. This proves (17).

By the Multiplicative Drift Theorem [DG13], $V'(t_r$ reaches 0 after at most $3eT_{\max}(r + \log T_{\max})$ steps with probability $1 - e^{-r}$. By a union bound over all i, with probability $1 - ne^{-r}$ all $V(t_r, i)$ reach 0 after at most $3eT_{\max}(r + \log T_{\max})$. Choosing $r = r_0/(3eT_{\max}) - \log T_{\max} = \Omega(\sqrt{f(n)} \log T_{\text{init}})$ gives the desired bound, with room to spare. □

Finally we are ready to prove the runtime bound for MAJORITY without bloat control.

Theorem 7.5. *For any starting tree with T_{init} leaves, the expected optimization time of the (1+1) GP without bloat control on MAJORITY is $O(T_{\text{init}} \log T_{\text{init}} + n \log^3 n)$.*

Proof. We only give the proof in the harder case $T_{\text{init}} \leq Cn \log^2 n$, for a suitable constant C that we choose later.

The theorem essentially follows from Lemma 7.4 by using restarts. Let $f(n) = \omega(1)$ be a growing function such that $f(n) \leq n$. We first assume $T_{\text{init}} \geq f(n) n \log^2 n$. We define a sequence $(T_i)_{i \geq 0}$ recursively by $T_0 := T_{\text{init}}$ and $T_{i+1} := \frac{1}{4} \sqrt{f(n)} T_i$. Moreover, we define

$r_i := \varepsilon f(n) T_i \log T_i$, where $\varepsilon > 0$ is the constant from Lemma 7.4. 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 $1-1/\sqrt{f(n)}$ a global optimum is found within the next r_{i+1} rounds without exceeding size T_{i+1}.

By Lemma 7.4 there is a high chance to find an optimum in r_0 rounds without blowing up 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 $p := 1/\sqrt{f(n)}$. 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 7.4 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 $p = 1/\sqrt{f(n)}$), 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

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

From the recursive definition we easily see that $T_i = (4p)^{-i} T_{\text{init}}$. Plugging this in, we obtain

$$\begin{aligned}\mathbb{E}[T_{\text{opt}}] &\leq \varepsilon f(n) \sum_{i=0}^{\infty} 4^{-i} T_{\text{init}} \log\left((4p)^{-i} T_{\text{init}}\right) \\ &= \varepsilon f(n) T_{\text{init}} \left(\log(T_{\text{init}}) \sum_{i=0}^{\infty} 4^{-i} + \log\left(\frac{1}{4p}\right) \sum_{i=0}^{\infty} 4^{-i} i\right) \\ &\overset{1/4p<n<T_{\text{init}}}{\leq} 3\varepsilon f(n) T_{\text{init}} \log T_{\text{init}}.\end{aligned}$$

This shows that for every arbitrarily slowly growing function $f(n)$ we have $\mathbb{E}[T_{\text{opt}}] \leq 3\varepsilon f(n) T_{\text{init}} \log T_{\text{init}}$ whenever $T_{\text{init}} \geq f(n) n \log^2 n$. We claim that this already implies for a suitable constant $C > 0$ that $\mathbb{E}[T_{\text{opt}}] \leq C T_{\text{init}} \log T_{\text{init}}$ for all $T_{\text{init}} \geq Cn \log^2 n$, i.e., we may replace the function $f(n)$ by a constant. 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 $T_C \geq Cn_C \log^2 n_C$ such that $\mathbb{E}[T_{\text{opt}} \mid t_{\text{init}} = t_C] > 3\varepsilon C T_C \log T_C$. Then choose a growing sequence C_i, for concreteness $C_i = i$. Since for each C_i there are arbitrarily large counterexample 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 $f(n) := \min\{i \mid n_{C_i} > n\} = \omega(1)$ and obtain a contradiction, since we have an infinite sequence of counterexamples for which $\mathbb{E}[T_{\text{opt}}] > 3\varepsilon f(n) T_{\text{init}} \log T_{\text{init}}$ and $T_{\text{init}} \geq f(n) n \log^2 n$. Hence we have shown for a suitable constant $C > 0$ that $\mathbb{E}[T_{\text{opt}}] \leq C T_{\text{init}} \log T_{\text{init}}$ for all $T_{\text{init}} \geq Cn \log^2 n$. This proves the theorem for the case □

We conclude the section with a remark on how to treat the case $T_{\text{init}} \leq Cn \log^2 n =: T'_{\text{init}}$. Note that $T'_{\text{init}} \log T'_{\text{init}} = \mathrm{O}(n \log^3 n)$. Hence, if at any point in time the GP-tree grows in size to T'_{init}, then we may apply Theorem 7.5 for this initial tree and conclude that the optimum is found after an additional $\mathrm{O}(n \log^3 n)$ steps. On the other hand, if the size of the GP-tree never exceeds T'_{init}, then we can apply the same argument as in Lemma 7.4, where we use $T_{\max} := T'_{\text{init}}$ in (17). We omit the details.