General Lower Bounds for the Running Time of Evolutionary Algorithms

Dirk Sudholt

International Computer Science Institute, Berkeley, CA 94704, USA

Abstract. We present a new method for proving lower bounds in evolutionary computation based on fitness-level arguments and an additional condition on transition probabilities between fitness levels. The method yields exact or near-exact lower bounds for LO, OneMax, and all functions with a unique optimum. All lower bounds hold for every evolutionary algorithm that only uses standard mutation as variation operator.

1 Introduction

Rigorous running time analysis has emerged as an important and active area in evolutionary computation. Results obtained by mathematical arguments help to judge the performance of evolutionary algorithms (EAs) on interesting problems and they can be used to compare different algorithms in a rigorous way. Running time analyses have been performed for many pseudo-Boolean functions [1] as well as for many problems from combinatorial optimization [2].

We contribute to this development with a new method for proving lower bounds for the running time of stochastic search algorithms. This method is applied to a very broad class of evolutionary algorithms for pseudo-Boolean optimization. The resulting bounds are not only tight in an asymptotic sense. They contain best possible leading constants when compared to upper bounds for the well-known (1+1) EA (see page 3 for a definition). We also make an effort towards stating bounds with precise constants for all involved terms, without resorting to asymptotic notation.

2 Previous Work

There is a long history of results on pseudo-Boolean optimization, including lower bounds. Already Droste, Jansen, and Wegener [3] presented a lower bound of $\Omega(n \log n)$ for the (1+1) EA on every n-bit pseudo-Boolean function with unique global optimum. The constant factor preceding the $n \log n$-term is $1/2 \cdot (1 - e^{-1/2}) \approx 0.196$. Wegener [1] mentions a lower bound $(1 - \varepsilon) \cdot n \ln n - cn$ where $\varepsilon > 0$ is an arbitrarily small constant and the constant $c > 0$ depends on ε. Doerr, Fouz, and Witt [4] presented a lower bound $(1 - o(1))en \ln n$ for the (1+1) EA on the function $\text{OneMax}(x) := \sum_{i=1}^{n} x_i$. This result was extended by Doerr, Johannsen, and Winzen [5]. They proved that the same bound holds for the (1+1) EA on every function with a unique global optimum.

The function LeadingOnes, shortly $\mathrm{LO}(x) := \sum_{i=1}^{n} \prod_{j=1}^{i} x_i$, is another popular test function that counts the number of leading ones in the bit string. Droste, Jansen, and Wegener [3] showed that the running time of the (1+1) EA on LO is at least $c_1 n^2$ with probability $1 - 2^{-\Omega(n)}$, for some constant $c_1 > 0$.

Black-box complexity of search algorithms as introduced by Droste, Jansen, and Wegener [6] is another method for proving lower bounds. These bounds hold for all algorithms in a black-box setting where only the class of functions to be optimized is known, but the precise instance is hidden from the algorithm. Their results imply that *every* black-box algorithm needs at least $\Omega(n/\log n)$ function evaluations to optimize OneMax and LO (or, to be more precise, straightforward generalizations to function classes). Very recently Lehre and Witt [7] presented a more restricted black-box model. If only unary operators are used that are unbiased w. r. t. bit values and bit positions, every black-box algorithm needs $\Omega(n \log n)$ function evaluations for every function with a unique global optimum.

Recently, drift analysis has received a lot of attention [8–10]. Assume a non-negative potential function such that the optimum is reached only if the potential is 0. If the expected decrease ("drift") of the potential in one generation is bounded from below, an upper bound on the expected optimization time follows. Conversely, an upper bound on the drift implies lower bounds on the expected optimization time. If there is a drift pointing away from the optimum on a part of the potential's domain then exponential lower bounds can be shown [8].

In this work we show that also a more direct approach is sufficient for proving good lower bounds. We introduce the new lower-bound method in Section 4, followed by applications for LO in Section 5 and OneMax in Section 6. Section 7 transfers the last result to all functions with a unique optimum.

3 Preliminaries

The presentation in this work is for maximization problems, but it can be easily adapted for minimization. The technique for proving lower bounds will be applied to a very general class of evolutionary algorithms. It contains all EAs that generate $\mu \in \mathbb{N}$ individuals uniformly at random and afterwards only use standard mutations to generate offspring (see Algorithm 1). The optimization time is given by the time index t that counts the number of function evaluations. The parent selection mechanism is very general as any mechanism based

Algorithm 1 Scheme of a mutation-based EA

create μ individuals $x_1, \dots, x_\mu \in \{0,1\}^n$ uniformly at random.
let $t := \mu$.
loop
  select a parent $x \in \{x_1, \dots, x_t\}$ according to t and $f(x_1), \dots, f(x_t)$.
  create x_{t+1} by copying x and flipping each bit independently with prob. $1/n$.
  let $t := t + 1$.
end loop

on the time index t and fitness values of previous search points may be used. Any mechanism for managing a population fits in this framework. This includes parent populations and offspring populations with arbitrary selection strategies and even parallel evolutionary algorithms with spatial structures and migration.

The (1+1) EA is a well-known special case with population size $\mu = 1$. It maintains a single individual x and in every iteration it creates x' by standard mutation of x and replaces x by x' if $f(x') \geq f(x)$ (for maximization problems). We denote by (1+1) EA_μ a generalization of the (1+1) EA that is initialized with a best individual out of μ individuals generated uniformly at random.

We review the *fitness-level method*, also known as the *method of f-based partitions* [1]. It yields upper bounds for EAs whose best fitness value in the population never decreases. We call these algorithms *elitist EAs*. The *optimization time* is the number of function evaluations until a global optimum is found.

Theorem 1 (Fitness-level method for proving upper bounds). *For two sets $A, B \subseteq \{0,1\}^n$ and fitness function f let $A <_f B$ if $f(a) < f(b)$ for all $a \in A$ and all $b \in B$. Consider a partition of the search space into non-empty sets $A_1, \dots, A_m$ such that $A_1 <_f A_2 <_f \cdots <_f A_m$ and A_m only contains global optima. For a mutation-based EA $\mathcal{A}$ we say that $\mathcal{A}$ is in A_i or on level i if the best individual created so far is in A_i. Consider some elitist EA $\mathcal{A}$ and let s_i be a lower bound on the probability of creating a new offspring in $A_{i+1} \cup \dots \cup A_m$, provided $\mathcal{A}$ is in A_i. Then the expected optimization time of $\mathcal{A}$ on f (without the cost of initialization) is bounded by*

$$\sum_{i=1}^{m-1} P(\mathcal{A} \text{ starts in } A_i) \sum_{j=i}^{m-1} \frac{1}{s_i} \;\leq\; \sum_{i=1}^{m-1} \frac{1}{s_i}.$$

The *canonical partition* is the one in which A_i contains exactly all search points with fitness i. It is known that for LO the method applied to the canonical partition yields an upper bound of $\sum_{i=0}^{n-1} en = en^2$ for the (1+1) EA since the probability of finding an improvement is lower bounded by the probability of flipping the first bit with value 0. This probability is at least $1/n \cdot (1-1/n)^{n-1} \geq 1/(en)$. For OneMax we get an upper bound of $\sum_{i=0}^{n-1} en/(n-i) = en \sum_{i=1}^{n} 1/i \leq en \ln n + O(n)$ for the (1+1) EA since on level i there are $n - i$ 1-bit mutations that flip a 0-bit to 1 and hence improve the fitness.

4 Lower Bounds with Fitness-Levels

The best lower bounds with fitness-level arguments known so far were presented by Wegener in [1], assuming fitness levels $A_1, \dots, A_m$ for some fitness function f:

Lemma 1. *Let u_i be an upper bound on the probability of an EA $\mathcal{A}$ creating a new offspring in $A_{i+1} \cup \dots \cup A_m$, provided $\mathcal{A}$ is in A_i (where "$\mathcal{A}$ is in A_i" is defined as in Theorem 1). Then the expected optimization time of $\mathcal{A}$ on f is at least*

$$\sum_{i=1}^{m-1} P(\mathcal{A} \text{ starts in } A_i) \frac{1}{u_i}.$$

The resulting lower bounds are very weak since we only look at the time it takes to leave the initial fitness level and then pessimistically assume that the optimum is found.

Making an additional assumption about the transition probabilities between fitness levels allows for much better lower bounds. In the following theorem $\gamma_{i,j}$ reflects the conditional probability of jumping from level i to level j, given that the algorithm leaves level i.

Theorem 2. *Consider a partition of the search space into non-empty sets $A_1, \dots, A_m$ such that only A_m contains global optima. For a mutation-based EA $\mathcal{A}$ we again say that $\mathcal{A}$ is in A_i or on level i if the best individual created so far is in A_i. Let the probability of traversing from level i to level j in one step be at most $u_i \cdot \gamma_{i,j}$ and $\sum_{j=i+1}^{m} \gamma_{i,j} = 1$. Assume that for all $j > i$ and some $0 < \chi \leq 1$ it holds*

$$\gamma_{i,j} \geq \chi \sum_{k=j}^{m} \gamma_{i,k}. \tag{1}$$

Then the expected optimization time of $\mathcal{A}$ on f is at least

$$\sum_{i=1}^{m-1} P(\mathcal{A} \text{ starts in } A_i) \cdot \left(\frac{1}{u_i} + \chi \sum_{j=i+1}^{m-1} \frac{1}{u_j} \right) \tag{2}$$

$$\geq \sum_{i=1}^{m-1} P(\mathcal{A} \text{ starts in } A_i) \cdot \chi \sum_{j=i}^{m-1} \frac{1}{u_j}. \tag{3}$$

If the same fitness levels are used and $s_i = u_i$ for all levels then (3) matches the upper bound from Theorem 1 up to a factor of χ.

Proof of Theorem 2. The second bound immediately follows from the first one since $0 \leq \chi \leq 1$. Let E_i be the minimum expected remaining optimization time, where the minimum is taken for all possible histories $x_1, \dots, x_t$ of previous search points with $x_1, \dots, x_t \in A_1 \cup \dots \cup A_i$. By definition $E_1 \geq E_2 \geq \dots \geq E_m = 0$ as the conditions on the histories are subsequently relaxed. By the law of total expectation the unconditional expected optimization time is at least $\sum_{i=1}^{m-1} \mathrm{P}(\mathcal{A} \text{ starts in } A_i) \cdot E_i$, hence we only need to bound E_i. The probability of leaving level i is at most u_i and the waiting time for this event is at least $1/u_i$. In case level i is left, the remaining time depends on the new level. We have

$$E_i \geq \frac{1}{u_i} + \sum_{j=i+1}^{m-1} \gamma_{i,j} \cdot E_j.$$

Assume for an induction that for all $j > i$ it holds $E_j \geq \frac{1}{u_j} + \chi \sum_{k=j+1}^{m-1} \frac{1}{u_k}$. Then E_i is at least

$$\frac{1}{u_i} + \sum_{j=i+1}^{m-1} \gamma_{i,j} \cdot \left(\frac{1}{u_j} + \chi \sum_{k=j+1}^{m-1} \frac{1}{u_k} \right) = \frac{1}{u_i} + \sum_{j=i+1}^{m-1} \frac{1}{u_j} \left(\gamma_{i,j} + \chi \sum_{k=i+1}^{j-1} \gamma_{i,k} \right)$$

where the equality holds since on the left-hand side every term $1/u_k$ in the inner sum appears for all summands $i+1, \dots, j-1$ in the outer sum, weighted by $\gamma_{i,k}\chi$. Using the preconditions on the γ-values, the last term equals

$$\frac{1}{u_i} + \chi \sum_{j=i+1}^{m-1} \frac{1}{u_j} \left(\frac{\gamma_{i,j}}{\chi} + 1 - \sum_{k=j}^{m} \gamma_{i,k} \right) \geq \frac{1}{u_i} + \chi \sum_{j=i+1}^{m-1} \frac{1}{u_j}. \qquad \square$$

Note that in order to apply the theorem, we only have to find an upper bound $u_i \cdot \gamma_{i,j}$ on the probability of jumping from level i to level j. In particular, we can allow ourselves some slack in the definition of u_i, which can make it much easier to prove the desired condition. Also note that the theorem does not require the sets A_i to form fitness levels. The only requirement is that all global optima are contained in A_m. Furthermore, "global optima" can be replaced by any other optimization goal such as finding high-fitness individuals.

5 A Lower Bound for LeadingOnes

Our first application is for LO as here the γ-values can be estimated in a very natural way.

Theorem 3. *Let X_μ be a random variable that describes the maximum* LO*-value among μ individuals created independently and uniformly at random. For every $n \geq 2$ the expected optimization time of every mutation-based EA on* LO *is at least*

$$\sum_{i=0}^{n-1} P(X_\mu = i) \cdot n \left(\left(1 - \frac{1}{n}\right)^{-i} + \frac{1}{2} \sum_{j=i+1}^{n-1} \left(1 - \frac{1}{n}\right)^{-j} \right) \tag{4}$$

$$\geq \frac{e-1}{2} \cdot n^2 - 4n \log n. \tag{5}$$

Proof. Consider the canonical partition and assume that the algorithm is on level $i < n$. This implies that in the best individual created so far the first $i+1$ bits are predetermined. In addition, in all individuals created so far the bits at positions $i+2, \dots, n$ have not contributed to the fitness yet. These bits have been initialized uniformly at random and they have been subjected to random mutations. It is easy to see that this again results in uniform random bits. More precisely, the probability that a specific bit j with $j \geq i+2$ in a specific individual has a specific bit value 0 or 1 is exactly 1/2 (see the proof of Theorem 17 in Droste, Jansen, and Wegener [3]).

Consider an individual x that has been selected as parent among the created individuals. Let $\mathrm{LO}(x) = j \leq i$. We bound the probability of creating an offspring with k leading ones for some $i+1 \leq k \leq n$. One necessary condition is that the first j leading ones do not flip, which happens with probability $(1-1/n)^j$. The bit at position $j+1$ is 0, hence it must be flipped. All bits at positions $j+2, \dots, i+1$ must obtain the value 1 in the offspring. This probability is determined by the number of ones among these bits. But clearly $(1-1/n)^{i-j}$ is a lower bound on this probability since this reflects the best-case scenario that all these bits are 1 in the parent. (Since $n \geq 2$ the probability of flipping a bit is not larger than the probability of not flipping it.) The last necessary condition is to create exactly $k-1-i$ ones among at positions $i+2, \ldots, n$. By the preceding arguments on the "randomness" of these bits, the probability of creating exactly $k-1-i$ ones is $2^{-k+i} := \gamma_{i,k}$ if $k < n$ and $2^{-k+i+1} := \gamma_{i,k}$ if $k = n$. Putting everything together, we have that $\left(1-\frac{1}{n}\right)^i \cdot \frac{1}{n} \cdot \gamma_{i,k}$ is an upper bound on the probability of jumping to level k.

Checking the condition on the γ-values, $\sum_{k=i+1}^{n} \gamma_{i,k} = \sum_{k=i+1}^{n-1} 2^{-k+i} + 2^{-n+i+1} = 1$ and for all $i < j \leq n$ condition (1) holds since

$$\sum_{k=j}^{n} \gamma_{i,k} = \sum_{k=j}^{n-1} 2^{-k-i-1} + 2^{-n-i} = 2^{-j+i+1} = 2\gamma_{i,j}.$$

Setting $\chi = 1/2$, the preconditions for Theorem 2 are fulfilled. Using $u_i := (1-1/n)^i \cdot 1/n$, this proves the bound

$$\sum_{i=0}^{n-1} \mathrm{P}\left(X_\mu = i\right) \cdot \left(n \cdot \left(1-\frac{1}{n}\right)^{-i} + \frac{1}{2} \sum_{j=i+1}^{n-1} n \cdot \left(1-\frac{1}{n}\right)^{-j} \right)$$

and hence (4).

The second bound (5) follows by simple calculations and the following case distinctions. If $n \leq 20$ the claimed bound is negative, hence we can assume $n \geq 21$. Observe that the bound (4) is never larger than en^2, even for the special case $X_\mu = 0$. If $\mu \geq en^2$ then the probability that the optimum is not found during the first en^2 individuals created during initialization is at most $en^2 \cdot 2^{-n} \leq 1/n$ for $n \geq 13$. This proves the claimed lower bound. If $\mu \leq en^2$ then $\mathrm{P}\left(X_\mu > 4\log n\right) \leq en^2 \cdot 2^{-4\log n} \leq n^3 \cdot n^{-4} = 1/n$ for $n \geq 3$. Pessimistically assuming that $X_\mu = 4\log n$ in case $X_\mu \leq 4\log n$ and estimating the conditional expected optimization time by 0 in case $X_\mu > 4\log n$ results in the bound

$$\begin{aligned}
&\left(1-\frac{1}{n}\right) \cdot n \left(\left(1-\frac{1}{n}\right)^{-4\log n} + \frac{1}{2} \sum_{j=4(\log n)+1}^{n-1} \left(1-\frac{1}{n}\right)^{-j} \right) \\
\geq &\left(1-\frac{1}{n}\right) \cdot n \left(\frac{1}{2} \sum_{j=0}^{n-1} \left(1-\frac{1}{n}\right)^{-j} - 2\log n \right) \\
= &\left(1-\frac{1}{n}\right) \cdot n \left(\frac{n-1}{2} \cdot \left(\left(\frac{n}{n-1}\right)^{n} - 1 \right) - 2\log n \right) \\
\geq &\left(1-\frac{1}{n}\right) \cdot n \left(\frac{n-1}{2} \cdot (e-1) - 2\log n \right) \\
\geq &\frac{e-1}{2} \cdot n^2 - (e-1)n - 2n\log n
\end{aligned}$$

$$\geq \frac{e-1}{2} \cdot n^2 - 3n \log n$$

for $n \geq 4$. □

Note that a term $-\Theta(n \log n)$ is, in general, necessary since with, say, $\mu = n$ an EA will start with an average of $\Theta(\log n)$ leading ones in the best search point.

For the (1+1) EA$_\mu$ $u_i \gamma_{i,j}$ is the exact probability of jumping from fitness level i to level $j > i$. Also recall that all conditions (1) on the $\gamma_{i,j}$-values hold with equality. Going through the proof of Theorem 2 again, we find that in this special setting all estimations are, in fact, equalities. This shows that the first lower bound on LO is exact for the (1+1) EA$_\mu$. This proves that among all mutation-based EAs the (1+1) EA$_\mu$ is an optimal algorithm for the function LO. For $\mu = 1$ we get the following.

Corollary 1. *The expected optimization time of the (1+1) EA on* LO *is exactly*

$$\sum_{i=0}^{n-1} 2^{-i-1} \cdot n \left(\left(1 - \frac{1}{n}\right)^{-i} + \frac{1}{2} \sum_{j=i+1}^{n-1} \left(1 - \frac{1}{n}\right)^{-j} \right) = \frac{\left(\frac{n}{n-1}\right)^{n-1} + \frac{1}{n} - 1}{2} \cdot n^2.$$

The factor preceding n^2 converges to $(e-1)/2$ from below. To the author's knowledge this is the first time the leading constant for the (1+1) EA on LO has been stated explicitly. This result also shows that additional information about transition probabilities between fitness levels is also useful for proving better *upper* bounds.

6 A Lower Bound for OneMax

Theorem 4. *The expected optimization time of every mutation-based EA on* OneMax *is at least* $en \ln n - 2n \log \log n - 16n$.

Proof. Assume that $n \geq 34$ as otherwise the claim is trivial. If $\mu \geq 2en \ln n$ then the probability that the first $2en \ln n$ search points generated during initialization find the optimum is at most $2en \ln n \cdot 2^{-n} \leq 1/2$, which establishes the lower bound $en \ln n$. In the following we assume $\mu \leq 2en \ln n$. Let $\ell = \lceil n - n/\log n \rceil$. Consider the following partition $A_\ell, \dots, A_n$. Let $A_i = \{x \mid |x|_1 = i\}$ for $i > \ell$ and A_ℓ contain all remaining search points. With probability at least $1 - 2en \log n \cdot \sum_{i=0}^{\log n} \binom{n}{i} 2^{-n} \geq 1 - 1/(\log n)$ for $n \geq 34$ the initial population only contains individuals on the first fitness level. Consider a situation where the algorithm is on fitness level i, i. e., the best-so-far search point has had up to i ones. The probability of reaching any specific higher fitness level $j > i$ is maximal if an individual with i ones is selected as parent. (See Lemma 11 in [5] for a formal proof.)

For $j > i$ let $p_{i,j}$ be the probability of the event that mutating an individual with i ones results in an offspring that contains j ones. A necessary condition for this event is that in this mutation either $j-i$ 0-bits flip to 1 and no 1-bit flips to 0 or that at least $j-i+1$ 0-bits flip to 1. This yields

$$\begin{aligned} p_{i,j} &\leq \binom{n-i}{j-i} \cdot \frac{1}{n^{j-i}} \cdot \left(1-\frac{1}{n}\right)^{n-j+i} + \binom{n-i}{j-i+1} \cdot \frac{1}{n^{j-i+1}} \\ &\leq \left(\frac{n-i}{n}\right)^{j-i} \cdot \left(\left(1-\frac{1}{n}\right)^{n-j+i} + \frac{n-i}{n}\right). \end{aligned}$$

For $i \geq \ell$ define

$$u'_i := \frac{n-i}{n} \cdot \left(\left(1-\frac{1}{n}\right)^{n-1} + \frac{n-i}{n}\right) \quad \text{and} \quad \gamma'_{i,j} := \left(\frac{n-i}{n-1}\right)^{j-i-1}$$

where the prime indicates that these will not be the final variables used in the application of Theorem 2. Observe that

$$\begin{aligned} u'_i\gamma'_{i,j} &= \frac{n-i}{n} \cdot \left(\left(1-\frac{1}{n}\right)^{n-1} + \frac{n-i}{n}\right) \cdot \left(\frac{n-i}{n} \cdot \frac{n}{n-1}\right)^{j-i-1} \\ &= \left(\frac{n-i}{n}\right)^{j-i} \cdot \left(\left(1-\frac{1}{n}\right)^{n-j+i} + \frac{n-i}{n} \cdot \left(\frac{n}{n-1}\right)^{j-i-1}\right) \\ &\geq \left(\frac{n-i}{n}\right)^{j-i} \cdot \left(\left(1-\frac{1}{n}\right)^{n-j+i} + \frac{n-i}{n}\right) \geq p_{i,j}. \end{aligned}$$

Since Theorem 4 requires the $\gamma_{i,j}$-variables to sum up to 1, we consider the following normalized variables: $u_i := u'_i \cdot \sum_{j=i+1}^{n} \gamma'_{i,j}$ and $\gamma_{i,j} := \frac{\gamma'_{i,j}}{\sum_{j=i+1}^{n} \gamma'_{i,j}}$. As $u_i\gamma_{i,j} = u'_i\gamma'_{i,j} \geq p_{i,j}$, the conditions on the transition probabilities are fulfilled. The condition $\gamma_{i,j} \geq \chi \sum_{k=j}^{n} \gamma_{i,j}$ is equivalent to $\gamma'_{i,j} \geq \chi \sum_{k=j}^{n} \gamma'_{i,j}$ and

$$\sum_{k=j}^{n} \gamma'_{i,k} = \sum_{k=j}^{n} \left(\frac{n-i}{n-1}\right)^{k-i-1} \leq \left(\frac{n-i}{n-1}\right)^{j-i-1} \sum_{k=0}^{\infty} \left(\frac{n-i}{n-1}\right)^{k} = \gamma'_{i,j} \cdot \frac{n-1}{i-1}.$$

Using $i \geq n - n/\log n$, we have

$$\frac{i-1}{n-1} \geq \frac{n-n/\log n - 1}{n-1} = 1 - \frac{n/\log n - 1}{n-1} \geq 1 - \frac{2}{\log n}.$$

Hence, choosing $\chi := 1 - 2/\log n$ we obtain $\sum_{k=j}^{n} \gamma'_{i,k} \leq \gamma'_{i,j} \cdot (n-1)/(i-1) \leq \gamma'_{i,j}/\chi$ as required. Now that all conditions are verified, we proceed by estimating the variables u_i. Bounding the sum of the $\gamma'_{i,j}$-values as before,

$$\sum_{j=i+1}^{n} \gamma'_{i,j} \leq \sum_{j=0}^{\infty} \left(\frac{n-i}{n-1}\right)^{j} \leq \frac{n-1}{i-1} \leq \frac{1}{1-\frac{2}{\log n}}.$$

Hence

$$\begin{aligned}
u_i &\le \frac{n-i}{n} \cdot \left(\left(1-\frac{1}{n}\right)^{n-1} + \frac{n-i}{n}\right) \cdot \frac{1}{1-\frac{2}{\log n}} \\
&\le \frac{n-i}{en} \cdot \left(1+\frac{1}{n-1} + \frac{e(n-i)}{n}\right) \cdot \frac{1}{1-\frac{2}{\log n}} \\
&\le \frac{n-i}{en} \cdot \left(1+\frac{3}{\log n}\right) \cdot \frac{1}{1-\frac{2}{\log n}} \\
&\le \frac{n-i}{en} \cdot \frac{1}{1-\frac{3}{\log n}} \cdot \frac{1}{1-\frac{2}{\log n}} \;\le\; \frac{n-i}{en} \cdot \frac{1}{1-\frac{5}{\log n}}.
\end{aligned}$$

Applying Theorem 2 and recalling that the algorithm is initialized on the first fitness level with probability at least $1-\frac{1}{\log n}$ yields the upper bound

$$\left(1-\frac{1}{\log n}\right)\left(1-\frac{2}{\log n}\right)\sum_{i=\ell}^{n-1}\frac{en}{n-i}\left(1-\frac{5}{\log n}\right) \;\ge\; \left(1-\frac{8}{\log n}\right) en \sum_{i=1}^{\lfloor n-\ell \rfloor}\frac{1}{i}.$$

Since $\sum_{i=1}^{\lfloor r \rfloor} 1/i \ge \ln r$ for any $r \in \mathbb{R}^+$, the bound is at least

$$\begin{aligned}
&\left(1-\frac{8}{\log n}\right) en \ln(n/\log n) \;=\; \left(1-\frac{8}{\log n}\right) en(\ln(n) - \ln(\log n)) \\
&\ge en \ln n - en \ln(\log n) - 8en \cdot \frac{\ln(n)}{\log n} \;>\; en \ln n - 2n \log\log n - 16n. \qquad \square
\end{aligned}$$

We remark that the lower bound does not hold for all mutation operators. The biased mutation operator in [11] leads to a bound $\Theta(n)$ for the (1+1) EA on OneMax.

7 Generalization to all Functions with Unique Optimum

Using arguments by Doerr, Johannsen, and Winzen [5], we now transfer the lower bound for OneMax to all functions with a unique global optimum. This yields a more precise result than the $\Omega(n \log n)$ bound by Lehre and Witt [7].

In [5] the authors proved that the expected optimization time of the (1+1) EA on OneMax is not larger than the expected optimization time of the (1+1) EA on any other function with unique global optimum. Their proof extends to arbitrary mutation-based EAs in a straightforward way.

Theorem 5. *The expected number of function evaluations for every mutation-based EA $\mathcal{A}$ on every function f with a unique global optimum is at least $en \ln n - 2n \log\log n - 16n$.*

Proof. For some $a \in \{0,1\}^n$ denote by f_a the function $f(x \oplus a)$ where $\oplus$ denote the bit-wise exclusive or. Observe that this transformation does not change the behavior of a mutation-based EA in any way, i. e., all mutation-based EAs have the same runtime distribution on f_a as on f. Hence, we do not lose generality if we transform the function f in such a way that 1^n is the global optimum.

Let $E^f_{\mathcal{A}}$ denote the expected optimization time of $\mathcal{A}$ on f and assume that the algorithm has already created search points $x_1, \dots, x_t$. Let $E^f_{\mathcal{A}}(i)$ be the minimum expected remaining optimization time for $\mathcal{A}$ given that $\mathcal{A}$ has only created individuals on the first i fitness levels so far, formally $x_1, \dots, x_t \in A_0 \cup \dots \cup A_i$ with $A_0, \dots, A_n$ the canonical partition for OneMax.

Observe that by definition, since the conditions on $x_1, \dots, x_t$ are subsequently restricted,

$$E^f_{\mathcal{A}}(n) \le E^f_{\mathcal{A}}(n-1) \le \dots \le E^f_{\mathcal{A}}(0).$$

Further define a more specific and slightly modified quantity for the (1+1) EA_μ: let $\widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(i)$ be defined like $E^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(i)$, but with the additional condition that the history $x_1, \dots, x_t$ contains at least one search point in A_i. Since we have only added a constraint, $\widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(i) \ge E^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(i)$.

Following Doerr, Johannsen, and Winzen [5], we now prove inductively that for all i it holds $E^f_{\mathcal{A}}(i) \ge \widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(i)$. Clearly $E^f_{\mathcal{A}}(n) \ge \widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(n) = 0$. Assume $E^f_{\mathcal{A}}(j) \ge \widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(j)$ for all $j > i$. Let x' be the next offspring constructed by $\mathcal{A}$. If the best OneMax-value seen so far is at most i and $|x'|_1 = k > i$ then the expected remaining optimization time is at best $E^f_{\mathcal{A}}(k)$ (or larger). If the new offspring has a smaller number of ones, the remaining expected optimization time is still bounded below by $E^f_{\mathcal{A}}(i)$. Thus, using the assumption of our induction,

$$\begin{aligned} E^f_{\mathcal{A}}(i) &\ge 1 + \sum_{k=i+1}^{n} \mathrm{P}\left(|x'|_1 = k\right) \cdot E^f_{\mathcal{A}}(k) + \mathrm{P}\left(|x'|_1 \le i\right) \cdot E^f_{\mathcal{A}}(i) \\ &\ge 1 + \sum_{k=i+1}^{n} \mathrm{P}\left(|x'|_1 = k\right) \cdot \widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(k) + \mathrm{P}\left(|x'|_1 \le i\right) \cdot E^f_{\mathcal{A}}(i). \end{aligned}$$

The best distribution for $|x'|_1$ is obtained when a parent z with exactly i ones is selected. (Note that the probability of selecting such a z might be 0, in which cases the real bound is even larger.) Let Z be the random number of ones when mutating z, then

$$E^f_{\mathcal{A}}(i) \ge 1 + \sum_{k=i+1}^{n} \mathrm{P}\left(Z = k\right) \cdot \widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(k) + \mathrm{P}\left(Z \le i\right) \cdot E^f_{\mathcal{A}}(i).$$

On one hand this is equivalent to

$$E^f_{\mathcal{A}}(i) \ge \frac{1 + \sum_{k=i+1}^{n} \mathrm{P}\left(Z = k\right) \cdot \widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(k)}{1 - \mathrm{P}\left(Z \le i\right)}. \tag{6}$$

On the other hand for the (1+1) EA_μ on OneMax we have

$$\widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(i) = 1 + \sum_{k=i+1}^{n} \mathrm{P}\left(Z = k\right) \cdot \widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(k) + \mathrm{P}\left(Z \le i\right) \cdot \widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(i),$$

which is equivalent to

$$\widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(i) = \frac{1 + \sum_{k=i+1}^{n} \mathrm{P}\left(Z = k\right) \cdot \widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(k)}{1 - \mathrm{P}\left(Z \le i\right)}. \tag{7}$$

Taking (6) and (7) together yields $E^f_{\mathcal{A}}(i) \ge \widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(i)$. Moreover, $\widetilde{E}^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(i) \ge E^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}(i)$. As $\mathcal{A}$ and (1+1) EA_μ are initialized in the same way, they share the same distribution for the initial fitness level. We conclude $E^f_{\mathcal{A}} \ge E^{\text{OneMax}}_{(1+1)\ \text{EA}_\mu}$ and the bound follows from Theorem 4 applied to (1+1) EA_μ. □

As a side result, the proof has also shown that the (1+1) EA_μ is an optimal algorithm for OneMax.

8 Conclusions

Using an adaptation of the fitness-level method, we have presented general lower bounds for the running time of mutation-based evolutionary algorithms in pseudo-Boolean optimization. The bounds for LO and OneMax are exact or exact up to lower-order terms when compared to upper bounds for the (1+1) EA. This is a rare occasion of results that are both very general and very precise at the same time. In addition, we have proven that among all mutation-based EAs the (1+1) EA_μ (for proper μ) is an optimal algorithm for LO and OneMax, with respect to the number of function evaluations.

The method itself is not restricted to the investigated setting. It is ready to be applied to other search spaces and further stochastic search algorithms.

Acknowledgment. The author was supported by a postdoctoral fellowship from the German Academic Exchange Service.

References

1. Wegener, I.: Methods for the analysis of evolutionary algorithms on pseudo-Boolean functions. In Sarker, R., Yao, X., Mohammadian, M., eds.: Evolutionary Optimization. Kluwer (2002) 349–369
2. Oliveto, P.S., He, J., Yao, X.: Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results. International Journal of Automation and Computing **4**(3) (2007) 281–293
3. Droste, S., Jansen, T., Wegener, I.: On the analysis of the (1+1) evolutionary algorithm. Theoretical Computer Science **276** (2002) 51–81
4. Doerr, B., Fouz, M., Witt, C.: Quasirandom evolutionary algorithms. In: Genetic and Evolutionary Computation Conference (GECCO '10). (2010) To appear.
5. Doerr, B., Johannsen, D., Winzen, C.: Drift analysis and linear functions revisited. In: IEEE Congress on Evolutionary Computation (CEC '10). (2010) To appear.

6. Droste, S., Jansen, T., Wegener, I.: Upper and lower bounds for randomized search heuristics in black-box optimization. Theory of Computing Systems **39**(4) (2006) 525–544
7. Lehre, P.K., Witt, C.: Black box search by unbiased variation. In: Genetic and Evolutionary Computation Conference (GECCO '10). (2010) To appear.
8. Oliveto, P.S., Witt, C.: Simplified drift analysis for proving lower bounds in evolutionary computation. In: Parallel Problem Solving from Nature (PPSN X). Volume 5199 of LNCS., Springer (2008) 82–91
9. He, J., Yao, X.: A study of drift analysis for estimating computation time of evolutionary algorithms. Natural Computing **3**(1) (2004) 21–35
10. Doerr, B., Johannsen, D., Winzen, C.: Multiplicative drift analysis. In: Genetic and Evolutionary Computation Conference (GECCO '10). (2010) To appear.
11. Jansen, T., Sudholt, D.: Analysis of an asymmetric mutation operator. Evolutionary Computation **18**(1) (2010) 1–26